

1

Agile Projects

7

Introduction	8
About the Book	9
The Problems	10
The Agile Manifesto	12
Agile Approach	14
The Benefits	16
Project Components	18
Summary	20

2

Agile Project Management

21

Traditional Approach	22
The Agile Approach	24
Agile Framework	26
Business Involvement	28
Team Motivation	30
Progress Monitoring	32
Communication	34
Reporting	36
Summary	38

3

DSDM

39

DSDM Atern	40
Principles	42
Project Lifecycle	44
Roles & Responsibilities	46
Facilitated Workshops	48
Prioritization	50
Iterative Development	52
Modelling & Prototyping	54
Timeboxing	56
Estimating	58
Quality	60
Risk Management	62
Summary	64

4

Scrum

65

Scrum Overview	66
Scrum Theory	68
Scrum Team	70
Scrum Master	72
The Sprint	74
Sprint Planning	76
Daily Scrum	78
Sprint Review	79
Sprint Retrospective	80
Scrum Artifacts	81
Product Backlog	82
Sprint Backlog	83
Progress Monitoring	84
Summary	86

5

Extreme Programming

87

Background	88
Concept	90
Values	92
Rules	94
Principles	96
Scalability	97
Practices	98
Issues	100
Summary	102

6

Lean Development

103

Concepts	104
Lean Principles	106
Eliminate Waste	107
Amplify Learning	108
Decide Late	109
Deliver Fast	110
Team Empowerment	111
Inbuilt Integrity	112
See the Whole	113
Summary	114

7**Getting Started****115**

Methods & Deliverables	116
Pre-Project	118
Terms of Reference	120
Feasibility Phase	121
Feasibility Assessment	122
Business Case	124
Outline Solution	125
Feasibility Prototype	126
Outline Plan	128
Summary	130

8**Foundations****131**

Objectives	132
Business Foundations	134
Requirements List	136
Solution Foundations	138
Management Foundations	140
Delivery Plan	142
Delivery Control	143
Risk Log	144
Change Control	145
Summary	146

9**Development****147**

Development Process	148
Exploration Phase	150
Engineering Phase	151
Timebox Plan	152
Testing	153
Assurance Pack	154
Evolving Solution	156
Deployment Plan	158
Benefits Realization Plan	160
Timebox Review	162
Summary	164

10**Deployment****165**

Implementation	166
Deployment Phase	168
Project Review	170
Benefits Enablement	172
End Project Assessment	173
Project Closure	174
Summary	176

11**Post Project****177**

Project Closure	178
Post Project	179
Benefits Assessment	180
Project Success	182
Key Deliverables	184
Summary	186

Index**187**

1

Agile Projects

This chapter provides the background to the need for an agile approach to projects. It covers the benefits of using an agile approach and the typical components of an agile project.

- 8** Introduction
- 9** About the Book
- 10** The Problems
- 12** The Agile Manifesto
- 14** Agile Approach
- 16** The Benefits
- 18** Project Components
- 20** Summary



Beware

This book is not a basic introduction to project management. That is covered in [Effective Project Management in easy steps](#).

Introduction

This book is primarily intended for project managers who are moving into the project management of agile projects. It will also be of interest to agile developers who wish to know more about project management. And finally it will also be of interest to anyone else who wishes to know more about the management of agile projects.

Traditional Projects

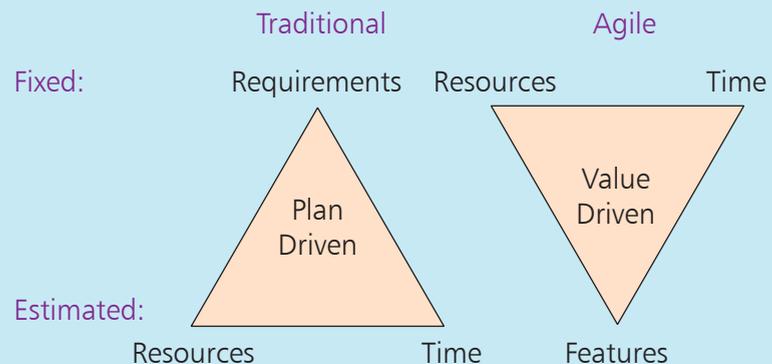
The traditional approach to projects and project management started by defining exactly what the project was expected to produce. This was termed the requirements or specification and was agreed and signed off between the project team and the business or customer.

The project team then went away and built a product or system that they thought met those requirements and, some time later, presented the finished product to the customer. The problems with this approach are set out later in this chapter but the end result was all too often that it was not what the customer needed.

Agile Projects

The agile approach to projects starts out with the expectation that the requirements (or features) will evolve and change during the course of the project. What is fixed and agreed between the project team and the customer is the resources that will be used and the time that will be taken by the project team to deliver as much as possible of the prioritized features the customer wants. The difference between the two approaches is illustrated below:

The Paradigm Shift



About the Book

This first chapter provides an introduction to agile projects, beginning with some of the difficulties of using the traditional approach and how the agile approach deals with them. It then examines the typical components of an agile project.

Agile Project Management

Chapter two goes through the details of agile project management starting with the differences between this and the traditional approach to project management. It then sets out a framework of agile project management, which covers all aspects of project managing an agile project.

Other Agile Components

While there are a wide variety of methods and processes that could be defined as agile, the best defined and most typical components are covered in chapters three to six. These are the Dynamic Systems Development Method (DSDM), Scrum, Extreme Programming and Lean Development.

Getting Started

Chapter seven deals with the agile approach to starting up a project including pre-project activities and establishing the feasibility. It covers producing the terms of reference, feasibility assessment and the outline project plan.

Project Phases

The next three chapters cover the main project phases of an agile project. Chapter eight introduces the foundation phase, including establishing the business foundations, developing the requirements list, systems architecture, development approach, solution prototype and delivery plan.

Chapter nine covers the development of the solution including the exploration and engineering phases. It also covers product assurance, testing, deployment planning and review.

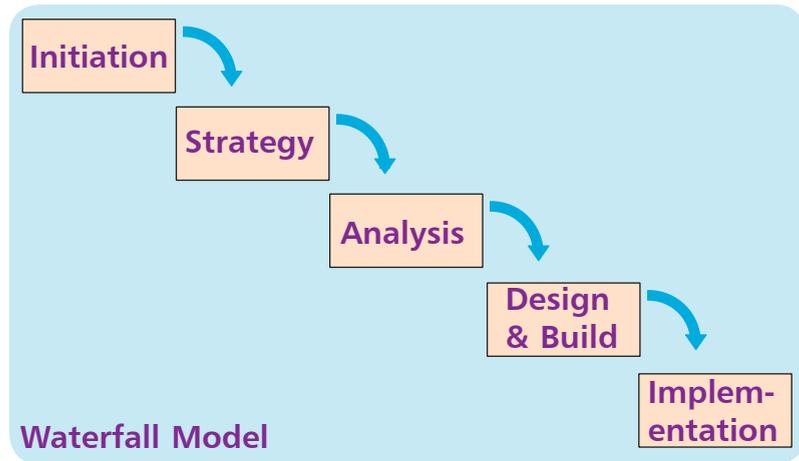
Chapter ten covers the deployment of the solution together with the project and increment reviews. It looks at benefits enablement and ends with the end project assessment.

Project Closure

Chapter eleven deals with closing the project down, planning and conducting a post-project benefits assessments. It concludes with a set of guiding principles for the success of an agile project.

The Problems

Traditional project methodology originally evolved from the construction industry, where the prohibitive cost of making late changes meant requirements were frozen as early as possible. It was based on a sequential design process often referred to as the waterfall model, so called as progress is seen as flowing steadily downwards (like a waterfall) through the stages of the project, each stage building on the work of the preceding stage.



Fixed Requirements

This methodology was initially adopted by software developers but it did not always suit the software development process, where requirements could (and very often did) change through the course of a project. Various attempts were made to deal with this, such as 'upstream propagation' in which changes made later in the project were fed back upstream to the earlier stages of the project and the requirements were then changed.

While this had some success a more radical alternative was needed and methodologies such as Rapid Application Development (RAD), Spiral, Iterative and Incremental were developed. In these the requirements are developed throughout the life of the project and delivered through a series of releases, which gradually deliver more functionality to the customer.

In addition to the rigid methodology and project life-cycle being used there were a couple of further problems stemming from the use of the traditional approach.

...cont'd

Business Involvement

In the traditional approach the business was kept well away from the development team. They were consulted in the early stages to define the requirements and they were involved at the end to test the finished product. But if they saw what was being done during the project, they might well decide it was wrong and ask the developers to change it. This would be disastrous for the project as re-working and the consequent need for re-testing would delay the project. Delivering on time was the prime aim of the developers, although it was rarely achieved.

The problem was that delivering something on time that was not what the business needed was an even worse outcome. It gradually dawned on people that it was better to involve the business actively in the project. By having them involved right through the project it would at least ensure that what was delivered was what the business needed, even if it was a little late.

Project Management

Traditionally the project manager operated in what has now been referred to as a Command and Control style. In this the project manager developed the detailed project plan, identifying all the tasks that required completion. They then allocated tasks to each member of the project team telling them exactly what to do. While this might, originally, have been fine for the construction industry it did not get the best or most creative output from software or engineering developers.

The solution was to empower the project team more and actively engage them in developing the detailed plans for the project. That way the plans were not only likely to be better but the developers would also be more committed to them. Secondly, by allowing the developers to decide for themselves what needed to be done to deliver the requirements, they could make best use of the skills and knowledge of each team member and probably get much better commitment from the team members.

These types of changes to the methods being used on software development projects began to be referred to as lightweight methods to differentiate them from the old heavyweight approach. These included DSDM, Scrum, Extreme Programming, Lean Development, Agile Testing and several more.

The Agile Manifesto

In February 2001, representatives and users of most of the more popular lightweight development methods met at the Snowbird ski resort in Utah to discuss the need for an alternative to the existing heavyweight, documentation-dependent, software development processes. At the end of the conference, they published their manifesto to define the approach now known as agile development. The manifesto reads as follows:

The Agile Manifesto

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.

What it Means

The meanings of the manifesto items on the left within the agile software development context are as described below:

- **Individuals and Interactions:** in an agile development, self-organization and motivation are important, as are interactions like co-location and pair programming
- **Working Software:** working software will be much more useful and welcome than just presenting detailed documents to clients in meetings
- **Customer Collaboration:** requirements cannot be fully defined at the beginning of the software development cycle, therefore continuous customer involvement is very important
- **Responding to Change:** agile development is focused on quick responses to change and continuous development, harnessing change for the customer's competitive advantage

...cont'd

Twelve Principles

These twelve principles underlie this agile manifesto:

- 1 Customer Satisfaction: by the early and the continuing delivery of useful software
- 2 Changing Requirements: welcome changing requirements, even late in the development process
- 3 Frequent Delivery: of working software, from every couple of weeks to every couple of months
- 4 Measure of Progress: delivery of working software is the principle measurement of progress
- 5 Sustainable Development: so the sponsors, developers and users can maintain a constant pace indefinitely
- 6 Close Cooperation: business people and developers must work together daily throughout the project
- 7 Motivated Individuals: by giving them the support they need and trusting them to get the job done
- 8 Face-to-face Conversation: the most efficient and effective method of conveying information in a development team
- 9 Technical Excellence: through continuous attention to technical excellence and good design
- 10 Simplicity: by keeping things simple the amount of work that has to be done is minimized
- 11 Self-organizing teams: the best architectures, requirements and designs emerge from self-organizing teams
- 12 Regular Adaptation: the team reflects on how to become more effective and adjusts its behavior accordingly

Agile Approach

As we have already seen, traditionally project management and software development were largely based on a sequential design process referred to as the 'waterfall' model. However this did not suit the software development process, where requirements could and often did change during the course of a project and more agile methods were evolved.

Project Management Methodology

Although traditional project management methodology can be applied to all types of projects there are some special constraints that apply to agile projects. The features (requirements) are allowed to change and evolve through the life of the project, while the resources and time are frozen. So the project will deliver as much of the prioritized requirements as can be delivered in the available time and within the cost budget.

Within each phase of an agile project, the developers collaborate closely with representatives of the business or customer so they understand the detail of the next step and can create an evolving solution. Before the product, process or software goes into production the business can decide if they want to continue on the same path or make alterations.

Agile Project Management

Because of the radical nature of these methods, the traditional (waterfall based) approach to project management, with requirements being defined and fixed early in the project, did not fit comfortably with this new approach. So a new form of agile project management began to develop.

In 2010 the DSDM Consortium published a definition of Agile Project Management, based on the DSDM method and interfacing with other agile methods such as Scrum and XP. This Agile Project Management differs from traditional project management in a number of key respects:

Management Style

On a traditional project, the project manager may be actively involved in directing the work of the team and telling them what to do. This is sometimes referred to as Command and Control. In agile project management the project manager is more of a facilitator and their role is to ensure that the collaboration between the business and the developers is effective.

...cont'd

Features

As the required features are expected to develop and change during the project the traditional approach of fixing requirements and allowing time and resources to flex to meet them is reversed. In an agile project time and resources are fixed (through time-boxing) and features are allowed to change at the start of each new iteration of the product.

Planning

In a traditional project the project manager would develop and own the project plan. In an agile project the features are constantly changing, so planning for each phase, release or iteration is carried out as late as possible. Further, although the outline project plan is produced by the project manager, the detailed plans are produced by the development team.

Project Phases

In place of the traditional (waterfall) project stages, agile projects use a number of phases, containing several iterations, leading to a number of product releases and therefore a series of implementations.

Change Control

The traditional project concept of change control is replaced by the features backlog. This is a list of prioritized business requirements, which is controlled by the business.

Risk Management

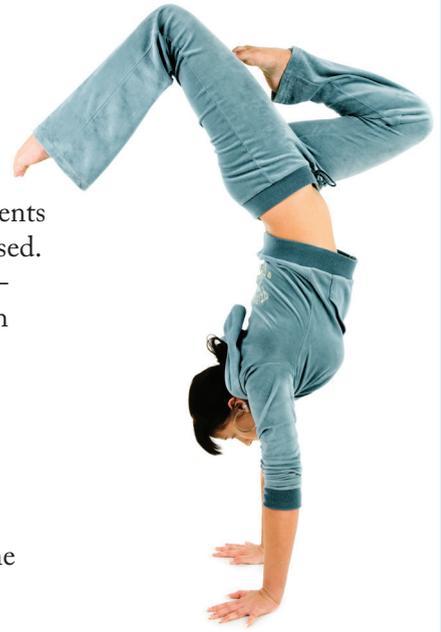
In place of the traditional approach to risk management and concerns about scope creep a broader approach to risk is taken in an agile project. The developers own the development risks and the business takes a more proactive role as the product owner.

Organization

In a traditional project the project manager hands out work packets to the team. In an agile project this is managed by the development team and the project manager takes on more of a supportive role to the team.

Monitoring Progress

In a traditional project the project manager has a detailed Gantt chart against which to monitor progress. In an agile project their role is to record the effort used on a burndown chart.



Hot tip



In an agile project, the project manager's role is to facilitate and support the team, not to tell them what to do.

The Benefits

The traditional (or waterfall) project management methodology, has been criticized for not being able to cope with constant changes in software or other types of development projects. This is probably the single most significant benefit of using agile methodology.

Changing Requirements

Not only does the agile approach have the benefit of dealing with changing requirements, it also overcomes the difficulty that customers have in adequately specifying their requirements in the first place, before they have even seen some sort of prototype. Specifying requirements before starting the actual development places a huge and unnecessary overhead on the project and is likely to cause long delays to the project starting to produce anything. The iterative nature of the agile approach makes it an excellent choice when it comes to managing development projects.

Customer Involvement

Failure to involve the business or external customer fully in the project is likely to lead to their eventual dissatisfaction with the final product. Close involvement of the customer means they can share the decision making, set the priorities and agree on the best solution to any issues. They will be committed to the final product and it will meet their requirements.

Quick Results

The use of agile methods ensures that the project delivers a quality product much earlier than would happen on a traditional project. The product will not yet have all of the required features but, on the Pareto principle, the delivery of the most important 20% of the features is likely to deliver around 80% of the benefits.

Progress Measurement

Using the delivery of a series of working products as the main form of progress measurement is one that everyone involved in the project can understand. The project manager can measure and report on it. The developers can work to delivering it and the customers can touch and feel it.

Team Motivation

By empowering the development team, allowing them to organize themselves and having active customer involvement the team will be much more highly motivated and produce better results. Close

...cont'd

cooperation on a daily basis with the customer will also add to their motivation and the delivery of a better product.

Product Quality

The focus on technical excellence and good design in an agile project coupled with continuous testing will ensure that a product of excellent quality is delivered. The close involvement of the customer will ensure their feedback to the process so that the product is not only excellent but it is what the customer needs.

The Ideal Project

We can now begin to see the type of project that can most benefit from the use of agile methods. Any development type project that has poorly defined requirements will benefit from the agile approach. The project should also have a fairly short timeline, ideally less than one year. The business or customer has to be able to make use of iterative product delivery with gradually enhanced features. The development team should not be too large because of the close cooperation and face-to-face working methods. A large team would begin to place a heavy communications burden on the project.

Disadvantages

Having established the ideal type of project to benefit from the agile approach, we can now see where it would not be beneficial. If this were not the case then all projects would be run using agile methods.

Many project management practitioners believe that agile methods do not scale well. Hence large-scale projects (even software development projects) should probably still be conducted using the traditional waterfall development and project management methods.

The strength and usefulness of agile methods are both clearly demonstrated in projects with poorly-defined and frequently changing requirements. So it would not seem to offer any advantage over traditional methodology when it comes to projects where the requirements are clearly defined and unlikely to change significantly over the course of the project. So large projects with clear requirements (such as major construction projects) are probably best managed using the traditional methods.

Hot tip



If your project fits this profile you should seriously consider using agile methodology.

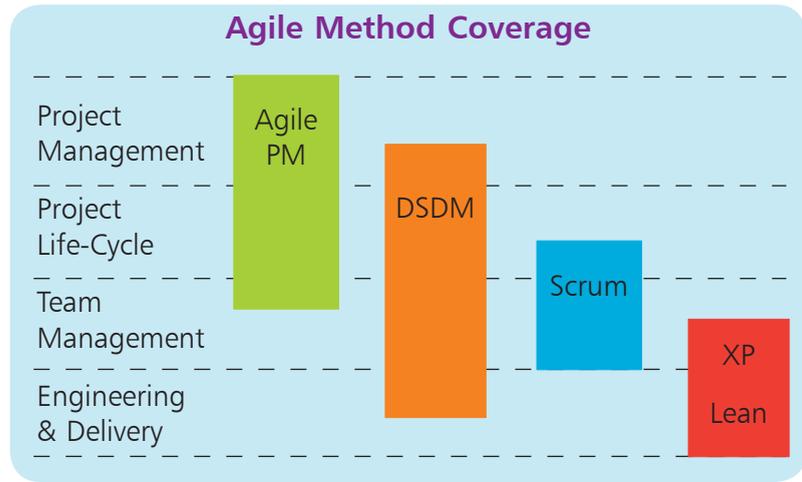
Beware



If your project fits this profile you should not be considering the use of agile methodology.

Project Components

In addition to Agile Project Management, the other typical agile methods that may be used on an agile project include: DSDM, Scrum, Extreme Programming and Lean Software Development. The relationship between these elements is illustrated in the following method coverage diagram:



As can be seen from the diagram above, there is a significant amount of coverage cross-over between the various methods. All of these agile methods are compatible and to a certain extent complimentary. Where the methods are not fully complimentary, then the most suitable method for the particular project can be selected by the project team.

Project Management

Agile Project Management was a new initiative launched in 2010 by the DSDM Consortium (www.dsdm.org). It took the project management and project life-cycle elements of DSDM and enhanced them by the addition of advice and guidance based on existing good practice. This methodology enables project managers to adopt an agile approach within their organizations and to interface well with agile development teams.

DSDM

The Dynamic Systems Development Model provides a framework for agile project delivery, which provides best practice guidance for the delivery of quality products on time and within budget. It is designed to be tailored and used in conjunction with other

...cont'd

agile methodology and traditional configuration and quality management systems.

Scrum

Scrum was originally formalized for software development projects, but works well for any complex, innovative project. Starting with the prioritized product backlog (requirements), the team pulls a small number from the top of that backlog and decides how to implement them. The team has an agreed amount of time (a sprint) to complete its work (usually two to four weeks) and meets each day to assess its progress (daily scrum).

At the end of the sprint, the work is delivered, the sprint is reviewed and the next sprint begins. This continues until all the product backlog has been completed or the time box ends.

Extreme Programming

Extreme Programming (XP) is another popular agile process, which stresses customer satisfaction. Instead of delivering everything on some date in the future this process aims to deliver the software that is needed as it is needed.

Extreme Programming allows the software developers to respond confidently to changing customer requirements, even late in the life cycle. Extreme Programming emphasizes teamwork with management, customers and developers all being equal partners in a collaborative team. The team is self-organized around the problem to solve it as efficiently as possible.

Lean Development

Lean Software Development was developed from the manufacturing process of the same name and follows agile principles. Lean development can be summarized by seven major principles which are again based on lean manufacturing principles: Eliminate Waste (including unnecessary code and functionality); Amplify Learning; Decide as Late as Possible; Deliver as Fast as Possible; Empower the Team; Build Integrity In (so components work well together) and See the Whole (software systems are the product of their interactions).

Putting together these (or similar) agile methods maximizes the effectiveness of the project team and ensures that the delivered product meets the needs of the business or customer.

Beware



XP and Lean methods only apply to software projects, DSDM and Scrum can be used on all types of development projects.

Summary

- Traditional projects start out by specifying and agreeing the requirements; agile projects start out by defining the resources and time frame and allow the required features to evolve through the life of the project
- This book provides an introduction to agile projects, a definition and framework of agile project management, a set of typical agile methods and a description of each phase of an agile project with their deliverables
- Traditional methodology was based on the waterfall model which did not suit projects with poorly defined requirements
- In addition to the problem of trying to fix the requirements too early in a project, traditional methodology also suffered from a lack of customer involvement and an authoritarian approach to managing the project team
- As a result of these problems software developers started evolving lightweight methods such as DSDM, Scrum, Extreme Programming and Lean Development
- This lightweight approach was defined in the Agile Software Manifesto, which was focused on individuals and interactions; working software; customer collaboration; and responding to change
- Trying to manage an agile project with traditional project management methodology proved problematic so the concept of agile project management was born
- A definition of agile project management was published by the DSDM Consortium and this focused on managing a project using the principles of the agile manifesto
- The ideal agile project is a development type project, with poorly defined requirements, a short time frame, a small team and full involvement of the business or customer
- Conversely, large projects, with well defined requirements are probably best managed using traditional methodology
- The typical components of an agile project are therefore likely to include some or all of: agile project management, DSDM, Scrum, Extreme Programming and Lean Development