

10.2_C

Project management approaches

The history of large-scale computing projects has many examples of projects which have gone wrong in different ways. Coordinating many different hardware and software elements to provide the necessary results can be very difficult. It is generally agreed that large-scale projects are particularly difficult to manage successfully, as Brooks [1] makes clear in his book, *The Mythical Man-Month*. Even though the percentage of project failures has declined over the years, evidence from the literature indicates that the number of failures remains high, with around 12% of projects seen as completely failed, 20% failing to meet their original goals and over 30% going over budget (Garbajosa, Wang & Aguiar [2]). Various project methodologies have been developed to increase control over projects.

Most practitioners seem to agree that waterfall and agile are the two most common approaches in project management. Waterfall project management methodologies generally assume that all the requirements are defined at the beginning of the project and that the product will only be delivered to the client at the end of the project. PM4A is one example of a relatively simple waterfall method, designed for smaller projects. PRINCE2 is another example, which is generally considered to be more suitable for large-scale projects. There are many different agile approaches to project management, all of which developed from the principles outlined in the Agile Manifesto in 2002. A key feature of agile is that it is iterative and it aims to deliver the product to the user at different times throughout the life cycle of the project. This has the advantage of allowing the user to adjust their requirements as they see different versions of the product. Scrum and Kanban are among the more popular agile approaches to project management.

Some writers have suggested that waterfall methodologies do little to actively manage projects. One reason, as Bentley [3] indicates, is that waterfall projects 'manage by exception'. This means the project manager only highlights problems when changes from the plan exceed their agreed tolerances. In other words, they only intervene when something is extremely wrong with the project. In addition, while waterfall methods discuss planning, risk and controls, they do not provide good practice on managing costs associated with delivery. Cost overruns are a huge failing in many computing projects, so it is important that some methodology is used by the organization to manage costs. Finally, another criticism of waterfall methods is that they are not well suited to working on smaller projects because of the enormous amount of documentation required. However, as a recent publication suggests, the level of documentation on waterfall methods can be adapted to meet the requirements of the project (Madsen [4]).

As a recent article indicates, agile approaches to project management also present a number of challenges for project managers (Santos & deCarvalho [5]). Firstly, they are not very useful for helping managers estimate the total cost of development. Secondly, if work is continued on a product beyond the time allocated to it, then the delivery of other pieces of work will be affected. The iterative nature of the work means that sometimes features may not be completely finished. This can be a major problem for security features. There is also a consensus that agile is not good at managing starting and ending a project. It is much more suited to continuous development. This means that there can be issues when using it on projects with a clear end date and clearly specified deliverables. The advantages of agile are that it is more likely to deliver features that users really want. It can improve the quality of the software and it is much more likely to deliver working software on time.

The literature suggests one possible solution to overcoming these problems is the use of hybrid project management techniques which combine waterfall and agile methodologies. A good example of this is the PRINCE2 Agile methodology. This builds on the existing PRINCE2 methodology which has been used by the UK government and large commercial organizations to control risks when managing big projects. PRINCE2 Agile works by creating an agile-based delivery layer which is separate from the governance layer on top. In this way, organizations should be able to get the benefit of structured risk management and the flexibility of agile delivery. However, the literature suggests that despite the promise of this approach, 'the challenges associated with doing this can be greater than they first appear' (Fance [6]). More work needs to be done to fully explore the range of what is possible using this approach.

10.2_C Answers

1 ■ 2 ■ 3 ■

Project management approaches

The history of large-scale computing projects has many examples of projects which have gone wrong in different ways. Coordinating many different hardware and software elements to provide the necessary results can be very difficult. It is generally agreed that large-scale projects are particularly difficult to manage successfully, as Brooks [1] makes clear in his book, *The Mythical Man-Month*. Even though the percentage of project failures has declined over the years, evidence from the literature indicates that the number of failures remains high, with around 12% of projects seen as completely failed, 20% failing to meet their original goals and over 30% going over budget (Garbajosa, Wang & Aguiar [2]). Various project methodologies have been developed to increase control over projects.

Most practitioners seem to agree that waterfall and agile are the two most common approaches in project management. Waterfall project management methodologies generally assume that all the requirements are defined at the beginning of the project and that the product will only be delivered to the client at the end of the project. PM4A is one example of a relatively simple waterfall method, designed for smaller projects. PRINCE2 is another example, which is generally considered to be more suitable for large-scale projects. There are many different agile approaches to project management, all of which developed from the principles outlined in the Agile Manifesto in 2002. A key feature of agile is that it is iterative and it aims to deliver the product to the user at different times throughout the life cycle of the project. This has the advantage of allowing the user to adjust their requirements as they see different versions of the product. Scrum and Kanban are among the more popular agile approaches to project management.

Some writers have suggested that **waterfall methodologies** do little to actively manage projects. One reason, as Bentley [3] indicates, is that waterfall projects 'manage by exception'. **This means the project manager only highlights problems when changes from the plan exceed their agreed tolerances. In other words, they only intervene when something is extremely wrong with the project.** In addition, while waterfall methods discuss planning, risk and controls, they do not provide good practice on **managing costs associated with delivery. Cost overruns are a huge failing in many computing projects, so it is important that some methodology is used by the organization to manage costs. Finally, another criticism of waterfall methods is that they are not well suited to working on smaller projects because of the enormous amount of documentation required.** However, as a recent publication suggests, the level of documentation on waterfall methods can be adapted to meet the requirements of the project (Madsen [4]).

As a recent article indicates, **agile approaches to project management** also present a number of challenges for project managers (Santos & deCarvalho [5]). Firstly, they are not very useful for helping managers estimate the total cost of development. Secondly, if work is continued on a product beyond the time allocated to it, then the delivery of other pieces of work will be affected. **The iterative nature of the work means that sometimes features may not be completely finished. This can be a major problem for security features.** There is also a consensus that agile is not good at managing starting and ending a project. It is much more suited to continuous development. This means that there can be issues when using it on projects with a clear end date and clearly specified deliverables. The advantages of agile are that it is more likely to deliver features that users really want. It can improve the quality of the software and it is much more likely to deliver working software on time.

The literature suggests one possible solution to overcoming these problems is the use of **hybrid project management techniques** which combine waterfall and agile methodologies. A good example of this is the PRINCE2 Agile methodology. This builds on the existing PRINCE2 methodology which has been used by the UK government and large commercial organizations to control risks when managing big projects. PRINCE2 Agile works by creating an agile-based delivery layer which is separate from the governance layer on top. In this way, organizations should be able to get the benefit of structured risk management and the flexibility of agile delivery. However, the literature suggests that despite the promise of this approach, 'the challenges associated with doing this can be greater than they first appear' (Fance [6]). More work needs to be done to fully explore the range of what is possible using this approach.