

The Benefits of Agile Development

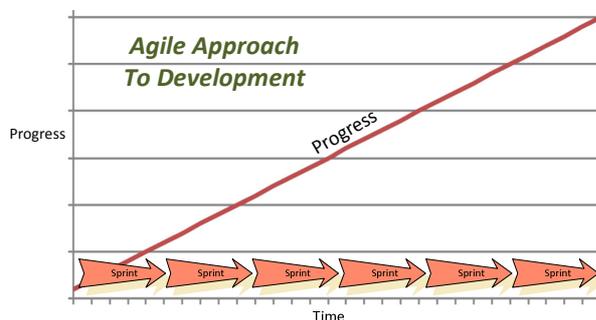
Author: Alan Miles, 7th November 2011.
<http://www.amba.co.nz>

The “Agile” methodology framework is an approach to project management typically used in software development, although it could be applied to other types of projects. It helps the development team to respond to unexpected problems and changes in requirements that can adversely affect the progress of any project.

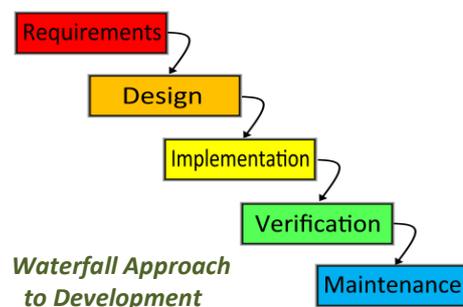
Agile development methods will greatly reduce both development costs and time to market. Organising developers into smaller teams means that the phenomenon known as “analysis paralysis” doesn’t happen and productivity is maintained by breaking the project down into “bite-sized” chunks.

In addition, there can be many opportunities to assess a project throughout the development lifecycle. Problems encountered that require the reassessment of the approach taken, and any changes in the perceived business requirements can be taken account of before they become a significant cause of resource “leakage”.

These opportunities are created by breaking up the work into short bursts (normally 2 weeks at a time) known as “Sprints” or iterations, at the end of which the development team delivers a useable release of the software under development, and re-evaluates the direction of the project, allowing adjustments to be made on a regular basis. This will often mean that the same section of code will be worked on repeatedly. So the Agile framework can be described as both “iterative” and “incremental.”

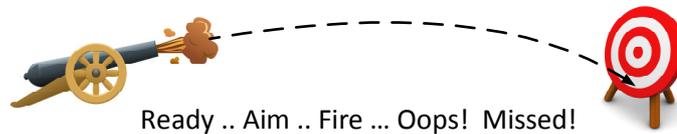


This is markedly different from the “waterfall” approach to development which requires all specifications to be completely known and “signed off” in advance. This provides the development team with only one chance to get each and every aspect of the project right.



In a “real world” complex situation, having this much insight from the start is often not possible, and has been the cause of many failed projects.

As systems became larger and more complex towards the end of the 1980s, development projects increasingly either blew out their budgets, or failed altogether. Agile methods were proposed and evolved during the 1990s as a response to the short-comings of what were increasingly seen as over-specified, tightly controlled, micromanaged, traditional methods of development.

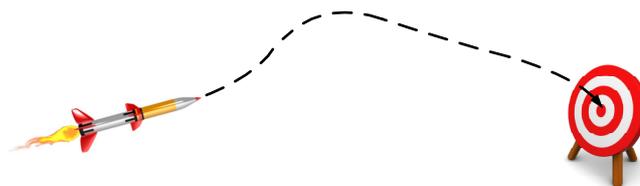


These methods had worked reasonably well with small projects, but as projects became larger and took longer, simply producing even more detailed specifications and applying a greater degree of control failed to achieve the desired results. One of the problems is that when project requirements are expected to be completely known before starting a project, there is a significant risk that the end product will not make full use of the potential that new technologies can deliver, or that the existing applications will simply be “updated” into some new technology that appears to meet the needs.

Even if what was contracted for was delivered on time and on budget (a rare occurrence anyway due to unforeseen problems), as the customer becomes more aware of new functionality available there is a realisation that the project scope needs to be changed, or else by the time the product is released to users it is already obsolete!

Why Agile?

Not only does the Agile framework allow developers to respond to unexpected problems and changes in the specifications during development, but a very significant advantage is that the customer may receive incremental improvements to the programs under development. This benefits the customer because desired features will be received earlier. Also, in spite of all the testing that will have been done, problems may arise when a new program is put into production. So it also benefits the developers, as any such problems can also be resolved incrementally and quickly, keeping the project “on track” and the code base “clean”.



Like a guided missile, the direction is constantly corrected to ensure the target is hit.

This incremental release of improvements enables businesses to build the right product the first time. If the product is being developed to release to market, it means that it can incorporate new methods as they become available so that the resulting product is as current and competitive as possible when released.

Agile methods depend on a relationship of trust between customers and vendors. The level of trust should build as customers and vendors collaborate throughout the project. There is no place for outdated attitudes in which the vendor is considered the enemy because you spend more time talking about contracts than ensuring the delivery of value. Similarly the vendor needs to feel invested in the project, rather than feeling that they have to negotiate an inflated price in case problems are encountered. Everyone benefits from successful projects.

The Agile development team needs the support of the business owners, directors, and their consultants. Ideally there needs to be someone with solid systems architecture skills working with and representing the interests of the business, so that the developers are well supported with clear working concepts and appropriate documentation as required.

The “Agile Manifesto”

In February 2001, 17 software developers met in Utah to discuss “Agile” development methods. They published the “Agile Manifesto”, which reads as follows;

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.

Individuals and Interactions – in Agile development self-organisation and motivation are important.

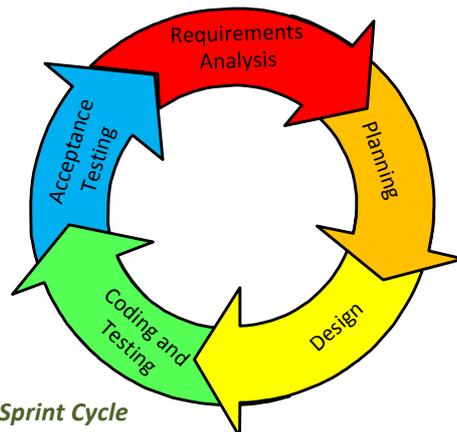
Working software – the business needs working software rather than impressive documentation.

Customer collaboration – customer involvement doesn’t end with a signed contract.

Responding to change – rapid responses to change mean they have minimal impact.

How Does it Function?

“Agile” development methods promote teamwork, collaboration, and adaptability throughout the life-cycle of the project. Tasks are broken into small increments with minimal planning, and do not directly involve long-term planning. These iterations known as “Sprints” are short time frames that typically last from one to four weeks.



Each Sprint involves a team working through a full software development cycle including requirements analysis, planning, design, coding, unit testing, and acceptance testing. The objective is to have an available release with no faults at the end of each Sprint, at which time a working product is demonstrated to the business. This minimises the overall risk, optimises predictability and allows the project to adapt to changes quickly.

For further details, contact Alan Miles, 027 220 9745, <http://www.amba.co.nz>