

The Agile Framework

Author: Alan Miles, 6th November 2011.
<http://www.amba.co.nz>



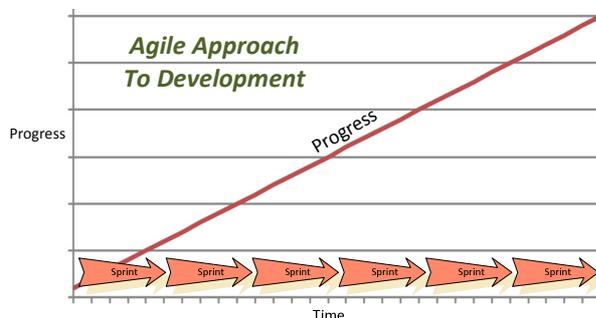
1. What is “The Agile Framework”?

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. This is achieved through incremental and iterative sections of work, known as “Sprints”.

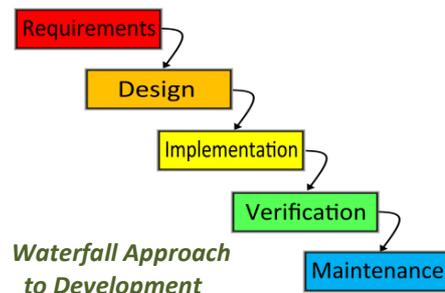
Agile development methods will greatly reduce both development costs and time to market. Organising developers into smaller teams means that the phenomenon known as “analysis paralysis” doesn’t happen and productivity is maintained by breaking the project down into “bite-sized” chunks.

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 workable increment of work, and re-evaluates the direction of the project, allowing adjustments to be made on a regular basis. This will often mean that the same code base will be worked on repeatedly. So the Agile framework can be described as both “iterative” and “incremental.”

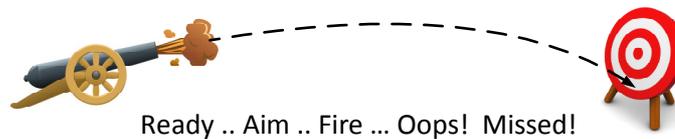


This is markedly different from the “waterfall” approach to development which requires all specifications to be completely known and “signed off” in advance. This provides the development team with only one chance to get each and every aspect of the project right.



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.

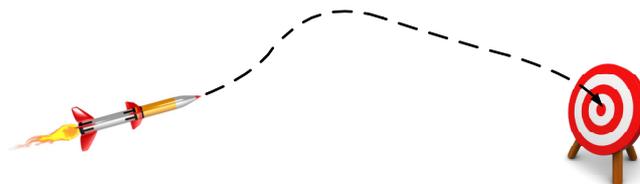


These methods had worked reasonably well with small projects, but as projects became larger and took longer, simply producing even more detailed specifications and applying a greater degree of control failed to achieve the desired results. One of the problems is that when project requirements are expected 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!

2. 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.

3. 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.

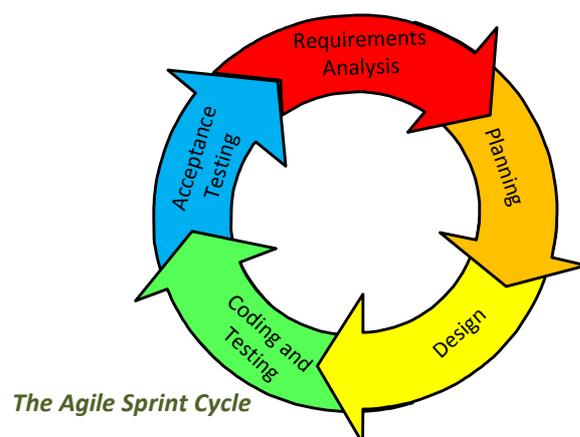
Working software – the business needs working software rather than impressive documentation.

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

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

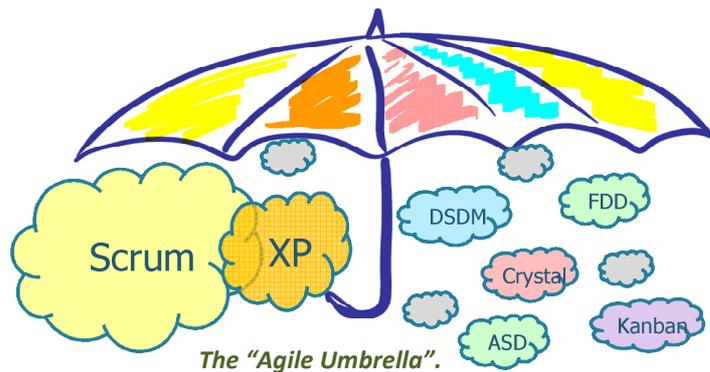
4. How Does it Function?

“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, at which point a working product is demonstrated to the business. This minimises the overall risk, optimises predictability and allows the project to adapt to changes quickly.



A single Sprint might not add enough functionality to justify a release to users, but the objective is to have an available release with no faults at the end of each Sprint. Multiple Sprints may be required to release the product to users with sufficient new features to justify the effort required to do so. This will depend to some extent on management and the operating environment.

There are a number of recognised methodologies that are described as “Agile”. Some of the earlier ones include Scrum (1995), Crystal Clear, Extreme Programming (XP) (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995).

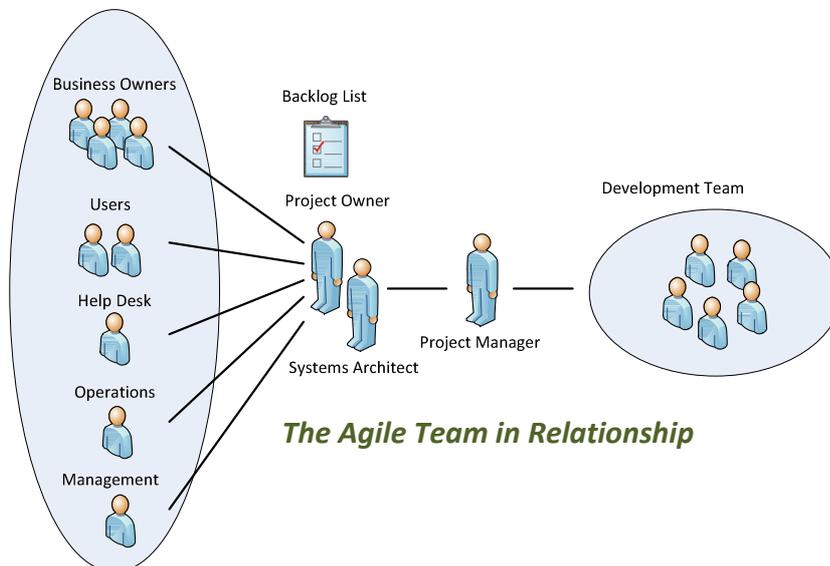


The diagram above is derived from original work by Henrik Kniberg. (www.crisp.se)  Used with permission.

There are variations in approach with each methodology, but the following description describes my personal interpretation of “Agile” methodologies as a whole. The details outlined below provide a basic framework that can be worked and adapted somewhat to gain the best advantage for the project, the technology used and the people involved.

5. The Agile Team

The Agile Team consists of a Project Owner, a Project Leader, a Systems Architect and the Development Team. The way in which these identities work together may vary somewhat, depending on the business and the individuals concerned.



a. *The Project Owner*

This is a single person who represents the interests of and reports to the owners, management, users and general staff of the business. With the assistance of the Systems Architect, any decisions required by the Development Team must be made in a timely manner so that the pace of development is not slowed. Also, the Project Owner sets the priorities of items remaining in the Backlog List (the list of features yet to be developed), so if anyone wants to change a backlog item’s priority, they must first convince the Project Owner.

For the Project Owner to succeed, he or she must have the full authority and confidence of the business' management and the entire organisation must respect his or her decisions, which are "final". No one is permitted to change the order of priorities set by the Project Owner, or to depart from the current set of requirements.

b. The Systems Architect

This is a single person who takes responsibility for the overall systems architecture. This needs to be someone with the high level vision required to ensure that the various pieces of development produced by multiple teams in multiple Sprints all fit together to form a cohesive whole.

For the following descriptions I have placed the Systems Architect in alignment with the Project Owner, although this role might be combined with the Project Leader. This will depend to a large extent on the business and the project.

c. The Project Leader

This is a single person (sometimes referred to as the "Scrum Master"), who is responsible for ensuring that the Agile methods being used are understood and properly executed.

The Project Leader assists the Project Owner to find effective methods to manage the Backlog List so that items are prioritised according to the needs of the business, as well as in a manner that will produce results from each Sprint, bearing in mind the relationships between separate pieces of work and long-term objectives. He or she ensures that the overall vision of the project is conveyed to the Development Team and that they accurately produce the results required from each Sprint.

When required, he or she will attend the daily stand-up meetings (or "stand-ups", sometimes referred to as "Scrums") with the Development Team so that problems are resolved and each Sprint is kept on-track. He or she needs to be able to coach the Development Team on how to apply Agile methods to stay productive while adapting to change, in a culture that enhances personal satisfaction with the task at hand.

The Project Leader co-ordinates any supporting activity required by the Development Team, and advises those outside the Development Team on how this support can best be delivered. When implementing the work produced by a Sprint, the Project Leader plans the implementation then trains and encourages users in the proper application of the new features and functionality released to them.

The Project Leader needs to have an attitude of "servant-leadership", meaning that amongst other things, they have to be confident, reassuring and value the needs of others before their own.

What is Servant Leadership?

The phrase "Servant Leadership" was coined by Robert K. Greenleaf in "The Servant as Leader", an essay that he first published in 1970. In that essay, he said:

"The servant-leader is servant first... It begins with the natural feeling that one wants to serve, to serve first. Then conscious choice brings one to aspire to lead. That person is sharply different from one who is leader first, perhaps because of the need to assuage an unusual power drive or to acquire material possessions...The leader-first and the servant-first are two extreme types. Between them there are shadings and blends that are part of the infinite variety of human nature." ...

... "A servant leader looks to the needs of the people and asks himself how he can help them to solve problems and promote personal development. He places his main focus on people, because only content and motivated people are able to reach their targets and to fulfil the set expectations."

d. The Development Team

This is a group of professionals who are chosen so that the make-up of the team is designed to include all the competencies required to complete the work, without needing to refer to any other person. This allows the team to choose how to proceed to best achieve the desired results, without needing to get direction from outside the team. Not even the Project Leader should direct the team how to proceed. As Agile Teams are totally self-contained, they possess all they need to be flexible, creative and productive.

Individual Development Team members may have specialised skills but team members do not have titles other than “Developer”, and accountability for success belongs to the Development Team as a whole. Furthermore, Development Teams do not contain sub-teams dedicated to particular domains like testing or business analysis. They are chosen to be capable of working together as a single unit, supported by and supporting each other. Development Teams do not have leaders, but should endeavour to work together as equals in a collaborative community. There is no hierarchy.

Agile methods emphasize face-to-face communication over written documents. Development Teams will generally work in a single open office which allows regular interaction where members can problem-solve and support each other. Sometimes team members may need to be located remotely, in which case phone, Skype, e-mail etc. should be used freely to ensure a community spirit is created through open communication, collaboration and co-ordination.

Independent from any other teams that may be involved with other parts of the project, Development Team members decide as a group how best to meet each Sprint’s requirements. The individual members then take responsibility for the tasks allocated to them to ensure the success of the Sprint.

Each Development Team is typically small (3 to 6 people), which simplifies team decision-making, collaboration and communication. The following diagrams illustrate the extent to which communication gets more complex as more members are added to a team;



Three members – three interactions

Six members – fifteen interactions

Development Team size needs to be small enough to stay flexible and able to develop a supportive sense of community. It also needs to be big enough to be able to make progress on significant blocks of work. Fewer than three members tends to decrease interaction (members tend to work as individuals) and can result in reduced efficiencies. In addition, teams that are very small may lack the capabilities required to complete the work required during a Sprint, causing the team to fail to meet their commitment to release a useable increment of work at the end of the Sprint.

Having more than six (or perhaps eight) members increases the complexity of interactions, so that the team becomes unable to properly coordinate activities without a leader. The Project Owner and Project Leader are not included in the count of Development Team members unless they are also doing development work that counts towards the objectives of the Sprint. In this case, all the responsibilities of any other Development Team member will apply.

Keeping Development Team sizes small need not limit the size of project that the Agile framework can be used with. If required, larger projects may be undertaken using small teams by allocating work to multiple teams that simultaneously work on different parts of the whole.

6. Agile Events

The Agile framework consists of various regularly occurring time-limited events that create a regular cycle. This allows all participants to settle in to a regular routine that feels familiar and controlled. The need for meetings other than as defined in the framework are minimised.

The most obvious event is the Sprint, during which the actual work of producing a result is done. All other events represent an opportunity to review and adapt. By this means, the whole process maintains a level of transparency as well as the obvious benefits of being able to fix something before it all goes horribly wrong.

If it can be seen that some aspect of a process fails to meet requirements then an adjustment must be made as soon as possible, so that future work continues in the right way. Adaptations can be made at any time the need is recognised, but there are four specific opportunities that Agile frameworks offer to review and adapt. These are the Sprint planning meeting, the daily Stand-up meeting, the Sprint review meeting, and the Sprint retrospective.

a. The Sprint Planning Meeting

A Sprint begins at the same time as the previous Sprint ends. The Planning phase of each Sprint occurs with a fresh evaluation of the project to date, and requirements to get to the desired outcome. Any adjustments to the overall architecture need to have been completed at this time.

At the Sprint Planning Meeting the entire Agile Team collaborates to understand and define the work that will be achieved in the next Sprint. The Sprint Planning Meeting is limited to eight hours for a one-month Sprint, and proportionally shorter for shorter Sprints.

Priorities are re-evaluated and the objectives of the next Sprint established. To achieve this, the Project Owner presents items from the Backlog List in priority order that aligns desired results with the business' needs and objectives. Items on the Backlog List have a brief description, a priority order, a risk factor and estimate of work required to complete.

Throughout this document, reference is made to "the Development Team". In general, sections of work may be allocated to as many Development Teams as may be required to get whatever progress may be required. In larger projects with more than one Development Team, one advantage is that there is more opportunity to allocate sections of work that are specifically suited to each team's individual talents.

When multiple Development Teams work together on the same project, the one Backlog List is used to provide items for work to be done in the next Sprint. To manage this situation, a further attribute can be added to group items by Development Team.

The Project Owner maintains the Backlog List, which remains under review throughout the project. It is assumed that it will be incomplete at the start and will change throughout the period that the project continues to be worked on. All desired features and functions are placed in the list at the start of the project, and these are added to by changes, additions and fixes. As successive iterations of released product are implemented, users will almost inevitably provide a stream of requests and suggestions that progressively add to the Backlog List.

Items at or near the top of the list will be clearer and more detailed than those lower down the list. They are placed at the top of the list not only because of the value to the business, but also because much more precise analysis has been done of what is required to successfully complete them in the course of a Sprint. This extra analysis work is done for those items for which there has been a consensus that they have the greatest value to the business, so in general both priorities will be aligned. Backlog List items that are selected for a Sprint are those that are considered ready to go.

Once items have been selected, the Development Team works to forecast what can be achieved during the next Sprint, based on previous progress made, the size of items under consideration, and previous experience of what is achievable.

The Development Team is solely responsible to define how many items are selected from the Backlog List and how it will build this functionality into a single release that will result in a workable increment of functionality. The Project Owner can assist the Development Team by explaining items from the Backlog List, and helping them to decide on any compromises required to make them attainable in a single Sprint. There is opportunity in subsequent Sprints to revisit any compromises made so that the “ideal” solution can be implemented.

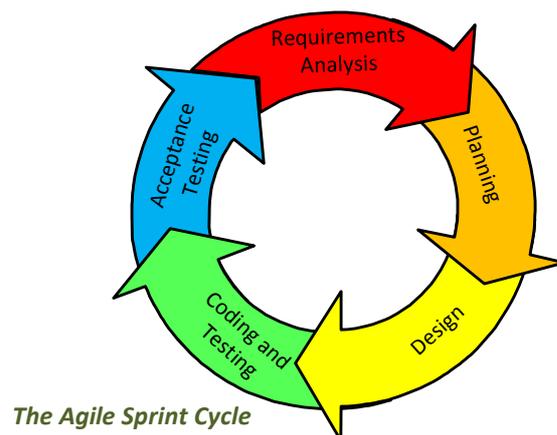
Once all items have been selected, the Sprint Objectives are stated as an expression of the work that will be delivered. The Sprint Objectives may be a significant milestone in the overview of the project. By the end of the Sprint Planning Meeting, the Development Team should be able to explain to the Project Owner and the Project Leader how it intends to achieve the work required to meet the stated Sprint Objectives.

b. The Sprint

The foundational activity of the Agile framework is a Sprint, which could be considered to be a small self-contained project that contributes to the overall vision of the project as a whole. Throughout the project, Sprints have a consistent length of not more than one month, during which an increment of work is completed that is fully functional and potentially able to be released.

If development was to continue for any longer than one month, the definition of what is being built may change significantly during the activity. Also, if a larger increment of work was done, the complexity would increase, which in turn would increase various risks. Anyway, as each Sprint produces a product that is fully functional and potentially able to be released to users, the financial risk is limited because the maximum value that is at risk at any point in the project is the expense associated with the duration of a single Sprint.

During a Sprint, no changes are made that would affect the Sprint Objectives, the composition of the Development Team is unchanged, and although the objectives remain fixed, the scope may be clarified and redefined between the Project Owner and Development Team as more is learned.



With every new Sprint, the Development Team starts all over again by identifying business requirements and the scope of the release to be achieved at the end of the Sprint. The tasks required to achieve the objectives are identified, and these are then allocated to team members.

In Agile methodologies the traditional architecture and design phase reduces somewhat to just design work. The high level systems architecture will have been largely determined by the Systems Architect in association with the Project Owner at the start of the project, and adjusted as necessary by the time of the Sprint planning phase.

The project's architecture is usually described to the Development Team using diagrams, showing technical infrastructure, the major business entities, business processes, and their relationships. Each team does all design work required to achieve the specific objectives for the current Sprint in enough detail to be able to follow through with the coding phase, which produces results in the limited context of the tasks allocated.

Development continues until the allocated Sprint Objectives are achieved. Unit Testing is carried out by each team throughout the coding phase within the relatively limited environment affected by the work being performed. This is crucially important so that the code base is maintained in a "clean" and tidy state. No code should ever be released with known faults still present.

During a Sprint, Development Team members need to frequently refer to the project's business requirement and design documents as a whole, especially with reference to the portions that relate to the current Sprint. Care needs to be taken to ensure that progress is made towards the end objectives of both the Sprint and the project overall. Any undesirable variance needs to be corrected immediately.

This reviewing process should not be so frequent that it gets in the way of the work. The experience gained with each iteration builds an accumulating level of understanding about the project, making future decisions increasingly easy to make with confidence built from past successes.

Each Sprint ends only when the Project Leader and the Development Team are convinced that the newly developed software meets the requirements. Once this occurs, the Project Owner is directly involved with the testing of each new release. This needs to occur whether or not it is intended to actually distribute the release to live users. This is crucial so that accurate feedback is given to the Project Leader and the Development Team as a new Sprint begins.



The difference between these two levels of testing is that Unit Testing is achieved from the developer's perspective and Acceptance Testing is conducted from the business' perspective. As the Project Owner, Project Leader and Development Team are in regular contact, any problems can be ironed out as soon as they are recognised. This ensures that the software produced is exactly matched to the business needs they are intended to meet.

c. Daily Stand-up Meetings

Development Teams meet daily for stand-up meetings (referred to as "Stand-ups"), so-named because they are held standing up to remind participants to keep the meeting short and to the point.

During the meeting each Development Team member describes what they have accomplished since the previous meeting, what they expect to get done before the next meeting, and any problems they can see that might prevent them from achieving their objectives. This provides a means to coordinate efforts and to resolve any problems and / or time-consuming issues before they seriously impact the Sprint.

If the Stand-up determines there are problems to overcome, then the Development Team will enter a planning meeting to work out how to overcome whatever the issue might be, and re-plan the rest of the Sprint's work before any further work continues. Every day, the Development Team should be able to explain to the Project Owner and the Project Leader how it will do the remaining work to achieve the Sprint Objective.

At each Stand-up, the team's progress towards the Sprint Objectives is assessed. In this way, all members are mindful of the approaching deadline and the expectation of being able to deliver working code. Although the Stand-up is run by and for the Development Team, it is desirable that the Project Leader is often present so that he or she maintains an overview of progress and a "feel" for how things are going. It can also be an opportunity for a limited amount of coaching.

Although these meetings are intended to be for the benefit of other team members and not a status update to the business, the Project Owner may also be available at these meetings to answer any questions, or address any mid-Sprint problems, allowing timely decisions to be made to ensure progress is not slowed. Team members may want to ask for some clarifications but the stand-up does not usually include lengthy discussions.

d. The Sprint Review

At the end of each Sprint, the Sprint Review meeting is held to present the new release of the software, review progress and update the Backlog List.

A demonstration of the new software produced is done in an informal way so that all concerned can get satisfaction from seeing progress made and value delivered. The Development Team describes what problems it ran into and how those problems were solved. This is an opportunity for feedback from all members of the Agile Team. On conclusion of the demonstration, the Project Owner confirms which items on the Backlog List have been completed and which still remain outstanding. A projection is then made of a likely project completion date based on progress so far.

The entire Agile Team reviews any changes or additions to the Backlog List, and discusses the options for the next increment of work to be done in the next Sprint.

If any changes or additions require adjustments in the system architecture, ideally this will have been done along with changes to the related documentation by the Systems Architect prior to the meeting so that the Agile Team has the new design to work from.

e. The Sprint Retrospective

The Sprint Retrospective is an opportunity for the entire Agile Team to review how well it has performed to date and make any adjustments required. Quality of product, relationships, execution of methods and use of tools all come up for review. Lists of things that went well and things that didn't go so well make this process transparent to all. The team gives special consideration to ways of improving those things that didn't go so well.

It is the Project Leader's job to encourage the Agile Team to improve and to coach all members on how best to apply the Agile framework in use so that it is more effective and satisfying for the next Sprint. Although improvements may be made at any time, the Sprint Retrospective provides a dedicated opportunity to focus on the Agile Team itself and make changes to continuously improve both performance and quality.

f. Cancelling a Sprint

Under rare circumstances, a Sprint may be cancelled before it is completed. Only the Project Owner has the authority to make the final decision to cancel a Sprint, although it is likely that the decision to do so has involved most or all of the Agile Team. In general, a Sprint would be cancelled if its objectives no longer make any sense. This might be through significantly changed circumstances such as the business changing direction, as can happen in a take-over. Another possibility is that some newly released technology now makes it best to stop work immediately to re-evaluate the project's direction. As a Sprint is of relatively short duration, the need for this sort of radical change should be rare.

When a Sprint is cancelled, the project is reviewed and further action starts with a new Sprint Planning Meeting. If any of the work done before cancellation is ready for implementation, then this could proceed, subject to approval by the Project Owner with advice from the whole Agile Team.

Sprint cancellations are expensive. There is a direct cost in the time taken for everyone to stop what they are doing and go into another Sprint Planning Meeting to review the current status. A less obvious cost is that it can be quite detrimental to team morale. As Agile methodologies rely on developing an optimistic community spirit, this can be quite expensive in terms of productivity.

7. Some “Glue”

Having broken the project into small pieces that are quite possibly undertaken independently by several separate teams of developers, it is important to attend to having a common language with clear definitions, and adequate documentation. These provide the “glue” to maintain the integrity of the project as a cohesive whole.

a. Language and Definitions

A common language and agreed definitions create a common standard that allow participants and observers alike to share a common understanding of what is being produced. This becomes crucially important when it comes to agreement between the Project Owner and the Development Team that work has proceeded sufficiently through a Sprint to gain final acceptance.

Key aspects of the process that define “success” must be clearly understood by all those engaged to create the desired outcome. In particular, what constitutes “completed” in a Sprint needs to be defined and understood by the whole Agile Team. If the definition of “completed” includes documentation (as I believe it should) then how much development work can be completed in each Sprint will be affected.

b. Documentation

Agile development emphasizes working software as the primary measure of progress. This combined with informal (usually face-to-face) communication can result in less written documentation than other methods. There needs to be some consideration given to this.

Documentation needs to be maintained in a complete and up-to-date state, otherwise if team members leave, a lot of knowledge is lost and it may be difficult for a project to recover from this. If fully working technical design documents are available, new team members or even entirely new teams will be able to familiarise themselves by reading them. So the technical documentation needs to be maintained by the Development Team as the work is done.

Because different developers will have different styles, the Project Leader should review all documentation produced and with a minimum of rework, pull it all together into a group of documents with a consistent presentation and style that is largely representative of the team as a whole. Establishing broad guidelines for documentation standards to guide the entire Agile Team would help in this process.

Operational documentation generally needs to be undertaken by or on behalf of the business, as it will need to include processes other than that involving the system under development.

8. Monitoring Progress

Progress in the project as a whole is monitored as is progress in each Sprint. Progress in the project as a whole is monitored by means of the Backlog List. This is a prioritised list of everything required to complete the product, so estimating the time required to complete each item based on past experience will provide a means to monitor progress towards the ultimate completion of the project. The Project Owner makes this calculation and presents the results at every Sprint Review meeting. Each new estimation can then be compared from Sprint to Sprint to get some idea of the pace of progress towards completion, and will establish a reasonable estimate of the time required.

Similarly, the Development Team maintains a list of items yet to produce before the end of each Sprint, which can then be used to estimate whether the Sprint Objectives will be achieved by the end of the Sprint. This is done for each Stand-up meeting.

There are various technical methods that can be used to evaluate the trends and forecast progress, but no mathematical trend analysis can forecast what problems may occur, or what brilliant new features may originate from users or arrive in the form of new technologies. While it is entirely up to the Project Owner whether to continue accepting new items for the Backlog List, the only thing that is certain is that the unexpected will occur!

9. Conclusion

The Agile framework as presented will make a big difference, when compared to traditional “Waterfall” development methods. The key individuals in the Agile Team are the Project Owner, Project Leader and Systems Architect. It is up to these three to provide supportive leadership and direction to the Development Team. They need to be able to maintain a high level view, but also to be able to see the detail in their dealings with the Development Team.

The Project Leader needs to have a “servant leader” attitude to the role, to enable others to perform to their best capability and to create a community spirit within the entire team. Having broken the whole project into small pieces, it is also up to the Project Leader to bring them all together to produce a single cohesive product. He or she needs to pay attention to enabling everyone on the entire Agile Team to work together as a single unit to achieve their true potential. This means being aware of how the system they are working in affects them and taking every opportunity to fine-tune the system to make the team more content and motivated.

10. Acknowledgements

I acknowledge the work done by Ken Schwaber and Jeff Sutherland, who first co-presented Scrum at the OOPSLA (Object-Oriented Programming, Systems, Languages and Applications) conference in 1995. Many others have also contributed since then, whose documentation helped me to put my own words and diagrams to methods that I have intuitively used for many years without being aware that these methods were recognised as a framework in general usage.

The diagram “The Agile Umbrella” was taken from a slide deck produced by Henrik Kniberg.

(www.crisp.se)  Used with permission under a Creative Commons license.

For licensing details, see <http://creativecommons.org/licenses/by/3.0/>

The original slides are available at <http://www.crisp.se/henrik.kniberg/>

Other images were produced using Microsoft Office Visio, with images purchased from Shutterstock, <http://www.shutterstock.com>