

Manage a Challenging Project by using agile methods effectively

Ian Stokes, January 2011

The “Agile Manifesto”
a declaration of agility

**Individuals and
interactions**
over processes
and tools



**Customer
collaboration**
over contract
negotiation



**Responding
to change**
over following
a plan



**Working
software**
over comprehensive
documentation

A presentation of Agile Project Management

1) The Business Case in an Agile Project

An agile development project should have a clear business case. The business case describes all the reasons why the project is necessary and desirable and the success of the project is evaluated based upon measurable goals that are expressed in the business case.

As well as the functional and technical requirements, in an agile project the business requirements must be well-defined. These are not limited to purely financial measures, but include non-financial measures that are nevertheless critical to success.

In an agile project, whilst the functional requirements are expressed from the point of view of the user, the business requirements should include adequate explanation of why they are required and why they have been requested.

For example:

<i>Functional requirement</i>	<i>Technical requirement</i>	<i>Business need</i>
The operator should be able to switch the machine on or off	The system shall contain a start and stop switch or button	The operator shall be able to switch the system on or off to make it available for calculations, printing, data update and to improve security
The sales person shall be able to add a new contact	The system shall contain the capacity to accept new contacts	The sales person shall be able to add new contacts in order to track and manage prospective clients
The sales director shall be able to search client data using keys words	The system shall be able to search in client data tables on key words	The sales director shall be able to access and use information on clients and market segments in order to track and supervise the sales cycle

The business needs are important, because on an agile project they become the baseline for measuring the achievement of the project goals. The business needs, as well as the functional requirements, are defined from the perspective of a user and are the basis for measuring success on an agile project.

Clarification of the business goals is not obvious. It demonstrates a clear link between the strategic purpose of the project and the overall strategy of the organisation, taking into account the business mission (what the organisation is) and the vision (what it intends to be.)

Building a vision requires a strong sense of aspiration, inspiration and knowledge of the skills and effort to be employed. The vision creates a narrative, in the form of a shared story, upon which a response to the user stories can be developed.

Thanks to this strategic alignment, requirements can add value and take account of the importance and urgency of the project. Meanwhile, costs are driven by the degree of complexity and risk. Thus, there is a continuous opportunity to review priorities in terms of value added compared to the costs.

The roles in an agile project are balanced between those that represent the business and the users on one hand, and those that represent the development team on the other. In this sense, agile project management methods suggest a variety of solutions from the very simple – business owner, project manager and project team – to the more elaborate.

A presentation of Agile Project Management

Roles such as the user ambassador (mandated by the users to represent their point of view), the visionary (centred on ideas and the convergence between the technology and the business), the technical coordinator (who manages the scope, a fundamental agile activity), the workshop facilitator, as well as the project manager and the project sponsor, can help to clarify the organisational structure of an agile project.

In practice, this evolving dialogue between the technology and the market implies frequent interaction between developers and users via inspections, tests, modelling, prototypes and simulations. These events are often organised as workshops. They are the essence of the agile approach.

Because the users participate in the tests, as much as possible, and use their real data in an authentic situation, they are in a position to approve the deliverables as they go along. This provides a sense of ownership that is priceless when the product must be approved. In fact, approval can take place systematically as intermediate versions are released.

This process of continuous integration, synchronisation and approval is reassuring both to developers and to the business because tangible and visible results are being produced. However, it must be managed, because the feedback can remove or add to requirements.

The ability to absorb changes reinforces the benefits without harming the stability of the project, because the approach is facilitated by rapid iterations. When deliveries are frequent, there is always a recent and operational version, whilst errors and misunderstanding are detected as soon as possible.

The philosophy of 'fail fast' aims to detect failures before they become too costly. Agile projects are learning projects. It is better to find out quickly if a solution is not going to work. Because the customer is involved, the risks of misunderstanding are reduced.

An effective agile project often employs technology built upon a flexible, modular and object-based architecture, using shared protocols and patterns of design that protect changes from consequences elsewhere and from problems of reversibility.



Business Case



User Story

A presentation of Agile Project Management

2) Planning an Agile Project

The agile planning approach begins with a deadline and works using the principle of the timebox. A timebox has a fixed duration and a limited content. The objective is to deliver the most important functionality within the timebox and to leave less important elements for a future timebox.

The word ‘timebox’ comes from the world of journalism. Each day or week in the life of a newspaper, the most probing results and the most important added value, must be produced within a limited time period that cannot be expanded. Anything else must wait until the next issue. Another word to designate ‘timebox’ is ‘sprint’, which is like a spurt of activity to produce measurable results.

At business level, there is an overall schedule and a calendar of business events. Meanwhile, most of the planning of an agile project takes place at the start of each ‘timebox.’

The ‘planning game’ consists of a team planning session where the tasks that would deliver requirements are written on cards. Worthwhile effort is evaluated in terms of benefits, cost and risk. As the project progresses through the iterations, the needs are prioritised and can be re-prioritised at the beginning of each timebox.

The priorities are often based upon the MoSCoW criteria (‘must’, ‘should’, ‘could’, ‘wont’) and taking due account of any technical constraints, tasks are ordered and work time assigned.

The ‘M’ requirements are those that must be delivered and are not negotiable. They are critical to the success of the project. Without these requirements, the solution would be inoperable, unusable, unused or useless, and even dangerous. These ‘must haves’ are, as a rule, contractual, in which case they must be rigorously assessed in order to calculate the required effort. An experienced team would typically target two thirds of the requirements as ‘must haves’, as opposed to one third for an inexperienced agile development team.

The ‘S’ requirements are those that should be satisfied. They lay represent between a third and two thirds of the remainder. Should an answer to these requirements be absent, it would be annoying and even problematic, but they are not mandatory for project success, because an alternative would be acceptable.

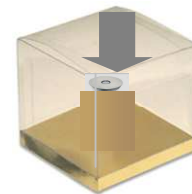
The ‘C’ requirements that could be satisfied are those that are sensitive to compromise. If they are not delivered, it will be in order to obtain some form of compensation such as respecting the constraints of the deadline, or other gain, such as an alternative functionality.

The ‘W’ requirements are those that will not be satisfied, at least in this timebox and are therefore postponed to a later date, or else considered as not useful or worth further effort.

Project updates must be frequent, because the tempo of an agile development project is very high. The reason for doing such a project is to obtain measurable and usable results early,



Prioritized Requirements



Timebox

A presentation of Agile Project Management

At the end of each day, the project development team runs a quick meeting to assess what has been achieved, what remains and whether there are any factors blocking progress.

Since the output of a timebox is often a set of models or prototypes, due care is taken to plan and to manage the modelling and prototyping process effectively. The prototypes may have different purposes. They may have to be tested and verified differently.

A ‘business prototype’ develops the understanding of the business requirements. Typically, a use case, a user scenario, user stories and process models are examples of ‘business prototypes’. A ‘business prototype’ may be in the form of a role play or an opportunity to observe the user, as well as a physical prototype.

A ‘usability prototype’ explores the interface between the solution and the customer. In the case of a product this is a physical interface – the screen, the control panel, or the instruments that enable a user to interact with the product or system. In the case of a service, this interface may be a combination of location, merchandising, point of sale and communication.

A ‘performance prototype’ enables the non-functional requirements; otherwise know as the quality of service requirements or the ‘ilities’, such as reliability, accessibility, portability, maintainability, safety, security, as well as the transaction speed to be measured and tested.

A ‘technical prototype’ experiments with different alternatives, from a technical perspective. As such this is the only prototype that is of interest primarily to the development team, as oppose to the users.

The term ‘prototype’ is used generically. In practice, a sketch may evolve; become a drawing, a design, a model, and then a prototype, going through several released versions before becoming the definitive product.

Thus prototype versions are planned within each timebox and the outputs of each prototype are specified clearly together with the checks and tests to be carried out.

Overall Planning Schedule			
Timebox 1	Timebox 2	Timebox 3	Timebox 4
<i>Prototype A1</i> <i>Prototype B1</i>	<i>Prototype A2</i> <i>Prototype C1</i>	<i>Prototype A3</i> <i>Prototype B2</i> <i>Prototype D</i>	<i>Prototype A4</i> <i>Prototype C2</i>
<i>A1 products</i> <i>B1 products</i>	<i>A2 products</i> <i>C1 products</i>	<i>A3 products</i> <i>B2 products</i> <i>D products</i>	<i>A4 products</i> <i>C2 products</i>
<i>A1 checks</i> <i>B1 checks</i>	<i>A2 tests</i> <i>C1 checks</i>	<i>A3 tests</i> <i>B2 tests</i> <i>D tests</i>	<i>A4 tests</i> <i>C2 tests</i>

A presentation of Agile Project Management

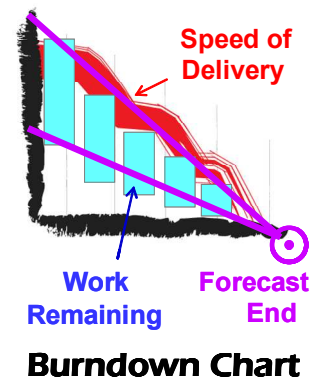
3) Progress Management

In the same way that deadlines are fixed on agile projects, costs can also be limited, so that the progressing of the project becomes a matter of how much scope can be delivered within these constraints. It is important to capture the feedback in terms of results accepted, actions defined, and needs identified, clarified or negated.

An agile project welcomes new requirements, provided that they are justified in terms of benefits and taking into account other the risks. However, the project team must demonstrate control over the process and show that the project is building value and converging upon a conclusion.

A 'Burndown Chart' is ideal for showing the requirements that have been revealed against those have been realised, in each timebox. When there are more requirements delivered and completed than new requirements discovered a point of convergence can be reached.

Key business performance indicators as well as project quality measurements are shown on the project scorecard, such as user satisfaction and number of modules that are operational.



Whilst the solution is being developed, agile conventions encourage developers to share their output and to co-own the modules they are developing. This is not mandatory, but it is part of the feedback process. Just as open source stimulates peer feedback and cooperation, peer programming is a recognised source of productivity, because errors are detected more easily and often much earlier.

Furthermore, an agile approach builds upon good design, modular architecture and the ability to reuse components, models and prototypes. Agile developers are conscious that a well-finished component will be much easier to maintain and to redeploy in the future.

The management of the progress of an iterative and agile project relies upon good configuration management, which is about understanding the status of the different components and their integration.

An agile environment is very dynamic. Everything that could change should be managed. The role of the technical coordinator in an agile project is the equivalent of the project coordinator in a traditional project. The role covers the managing of the impact of changes on other components, ensuring that modules can interoperate and synchronise, and ensuring that every component is testable and has been tested.

During the life of the agile project, the process of developing and testing requires frequent contact between developers and users, which this is facilitated by workshops. By the end of the project, the acceptance tests are likely to be much more straightforward because the business and the development team will have built their understanding and the users are more likely to have developed a sense of ownership, which is undoubtedly a critical success factor for any technology project.

A presentation of Agile Project Management

Annex A) When to use Agile Methods

The project is founded on a solid Business Case



Some of the requirements are more important or urgent than others



Requirements will evolve and the project will generate new knowledge



A dialogue can be established between developers and customers in order to provide and obtain feedback

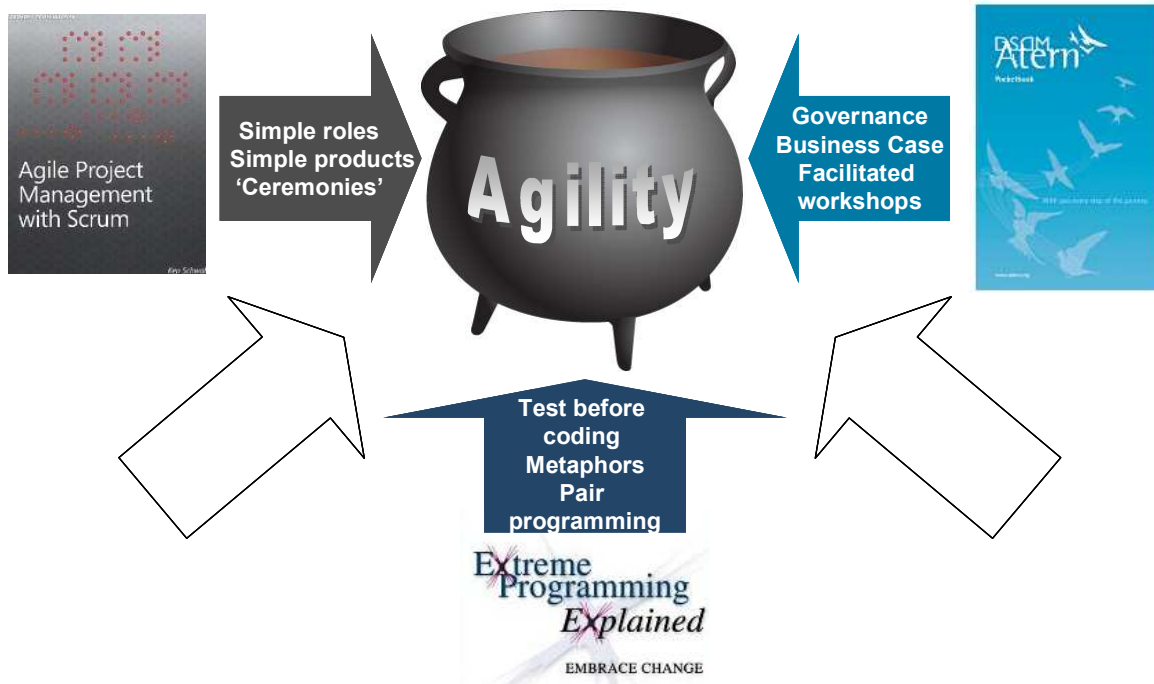


An architecture and the means exist for prototyping and modelling



A presentation of Agile Project Management

Annex B) Summary of Methods



Annex C) A Subjective Comparative Matrix of Agile Methods

