

Manage a Challenging Project by using agile methods effectively

Ian Stokes, January 2016

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

Another Perspective on Agile Project Management

1) Introduction

Within the context of any economic activity with enterprise environmental factors and organisational process assets, aspects such as the competitive climate and the technological opportunities are constantly changing. An agile approach to project management enables and facilitates an enhanced interaction between the needs of customers and the potential of the technology, in order to constantly assess value and to ensure that the effort is worthwhile.

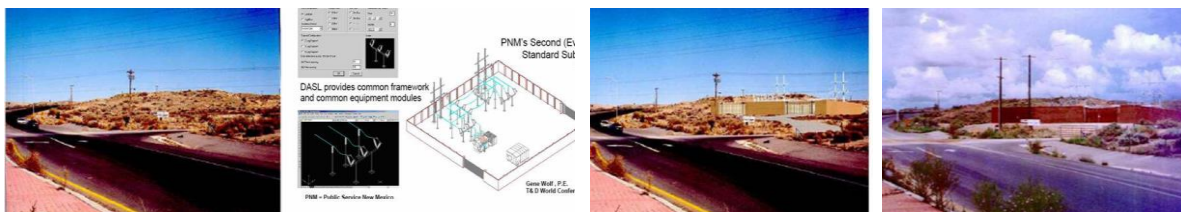
In essence, an agile project is a learning project and this reality imposes constraints on the project such as the necessity to experiment and communicate using models and prototypes, to explore options from different viewpoints, to include beneficial adjustments without harming the overall functionality, and to demonstrate control over the evolving scope. Decisions are justifiable if there is likely to be sufficient value for money compared to other alternatives. *

An important consequence of this is that the business understanding becomes the baseline for cooperation, creating a shared reference that is used to continually assess fitness for purpose, in contrast to the conventional approach that involves delegating a clearly defined set of requirements used as the basis for acceptance testing. A well-developed business case is even more important for an agile project than otherwise, because decisions on scope are being made throughout the project life cycle and are driven by an evolving interpretation of priorities and effort, rather than a contractual agreement at the beginning of the project.

This kind of business evaluation requires a trusting and informed commitment from different categories of customers, users, experts, managers, designers, builders, operators and investors. Inevitably with such a broad range of contributions, there will be at least some divergence and conflict of interests. Relational understanding, negotiation, creative problem-solving and frequent moments of 'storming' are tackled with purpose and by using a facilitative style of management that relies upon open transparent decision making and joint team workshops.

Agile methods on projects are not limited to software development, though they benefit from the ability to deliver parts of the solution in increments which can be tested; either function by function (vertical slicing), like a product you upgrade, extending your use at every stage; or like a progressively developing assembly (horizontal slicing), that you can see evolving at every step from the first blueprint, through the layout, mock-up, simulations, models and samples, prototypes, and even versions, until stabilised in the definitive article; or more likely, a mix of both.

This is an illustration of the progressive elaboration of a construction project. Each step provides an opportunity to verify, to validate, to test, to accept and to provide feedback



1. Proposed Site 2 Architect's design 3 Computer image 4 Completed project

* *On an agile project, benefits may creep upwards, providing that deadlines are respected, whilst scope should actually creep downwards, in the sense that unessential requests are removed to the benefit of necessary outcomes.*

Another Perspective on Agile Project Management

2) The Business Case in an Agile Project

Managing a project using agile project management methods relies upon a clear Business Case to describe why the project is necessary and desirable. The success of the project will be evaluated in terms of measurable objectives that are explained 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 needs should include adequate explanations of why they are required and how they relate to the functional requirements.

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 basis for evaluating priorities in terms of value added for effort and resources invested.

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, it is just as possible to apply a very simple set of roles, such as 'business owner', project leader and project team, or a more elaborate and localised collection of roles.

Another Perspective on Agile Project Management

Roles such as the user ambassador (mandated by the users to represent their point of view), the visionary (focused on ideas and the convergence between the technology and the business), the technical coordinator (who manages the content, a fundamental agile activity), the workshop facilitator, as well as the project manager and the project sponsor, can serve 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 as much as possible in these tests and use their real data in an authentic context, they are able to approve developed product and deliverables as the projects progress. This gives a feeling of ownership to the users, which is noticeable and has positive effects when the product is formally accepted. In fact, approval can be organised in a systematic fashion as intermediate versions are delivered.

This approach to integration, constant synchronisation and acceptance is reassuring as much for the users as for the business, because tangible and visible results emerge. However, it must be managed, because the resulting communication may lead to requirements being removed or added.

The ability to absorb changes reinforces the benefits without harming the stability of the project, because the approach is facilitated by rapid iterations. As long as the deliveries are frequent, there will always be a recent and operational version available. Thus, problems and misunderstandings are detected as early as possible, and any failure will be quickly evident.

The philosophy of “fail fast” is about revealing defects quickly before they become too costly. Agile projects are learning projects. It is preferable to know early if a solution is unsatisfactory. Furthermore, when the client is involved, the risks of misunderstanding are reduced.

An effective and agile project will often use a technology that is built upon a flexible and modular object-based architecture. There will be common protocols and design models that create resilience to changes and problems of reversibility.



Business Case



User Story

Another Perspective on Agile Project Management

3) 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.

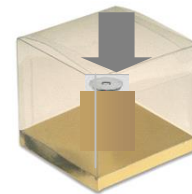
The ‘C’ requirements that could be satisfied are those that are sensitive to compromise. If they are not delivered, it will be in order to obtain some form of compensation such as respecting the constraints of the deadline, or other gain, such as an alternative functionality.

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

Another Perspective on 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>

Another Perspective on Agile Project Management

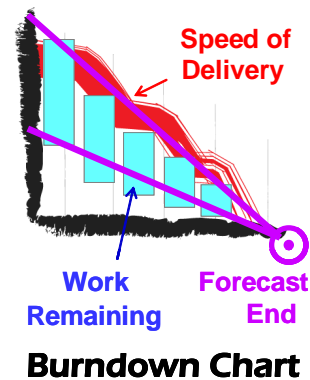
4) Progress Management

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 that 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.

Another Perspective on Agile Project Management

5) An example of how to test, to manage risks and to close an agile project

Upon arrival each morning in a highly agile project the lead user tested, using real data, the working software delivered the previous night, red-lined the mission-critical problems and requests, pink-lined the annoying difficulties for which there was a work-around and left the cosmetic issues blank.

Later in the morning upon their arrival, and after coffee, the team reviewed the daily status during a 15 minute telephone meeting, identifying obstacles and focusing the work to be done during the day.

The lead user developed the artwork for the project brochure and acted as the ambassador for the wider user community. His office was decorated with a Zen garden, equipped with sand, stones, bonsai trees and a rake, handy for those mornings when the work backlog seemed to be growing faster than the remedies. User prototyping was carried out every two or three weeks and organised around a buffet lunch.

Measures were based on the number of outstanding problems and requests per module, and the modules were considered workable for the lead user once the red-lined blockers were eliminated; and usable by other selected members of the user community once the pink-lined difficulties had been resolved. An electronic project work book centralised the latest version of the project content. The users, who could consult the status of the remaining requests and problems, obtained a summary report on the maturity of the whole system or for each module.

The contract was fixed price with an agreed deadline for delivery to a selected user community, based on a prioritised set of requirements. A roll-on contract was awarded in order to deliver to a wider user community, including some extra requests in a second version.

After two years of intensive use, users requested an upgrade and a transfer to a new platform. In this second development, due to the unavailability of a lead user, a more conventional approach was adopted, testing was delegated to an external consultant and the development took almost three times as long.

This experience reveals some of the threats and opportunities in the set-up of an agile project environment. Without an adequate level of involvement, cooperation and commitment, both from users and developers, the project and process can be subverted.

These are some of the symptoms:

- Key participants are not available for decisions or not constructive in workshops
- When requirements are defined or evolve, they are not adequately prioritised
- Developers are unable to interpret business requirements in terms of deliverables
- The working environment does not enable iterative and incremental development
- Prototyping and workshops are unwarranted, because needs are already well understood

Nevertheless, the point of using an agile approach is to avoid some of the significant traditional risks that occur due to misunderstanding between the project protagonists:

- High costs of problems, bad news and failures because they are discovered too late
- Over-specification of expensive features, included just in case they may be needed
- Expensive changes due to an environment and architecture not designed for agility
- Reporting that may lack visual, tangible, constantly checked and accepted outputs
- Missing commitment, cooperation or communication leading to a lack of ownership

Another Perspective on 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

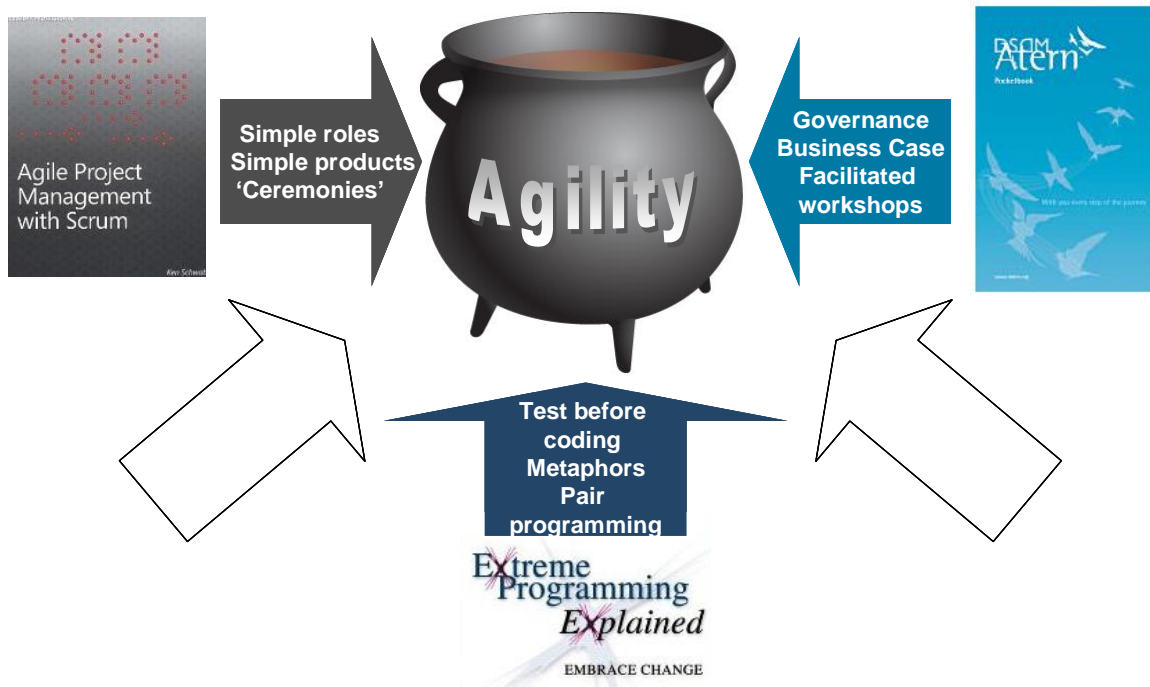


An architecture and the means exist for prototyping and modelling



Another Perspective on Agile Project Management

Annex B) Summary of Methods



Annex C) A Subjective Comparative Matrix of Agile Methods

