

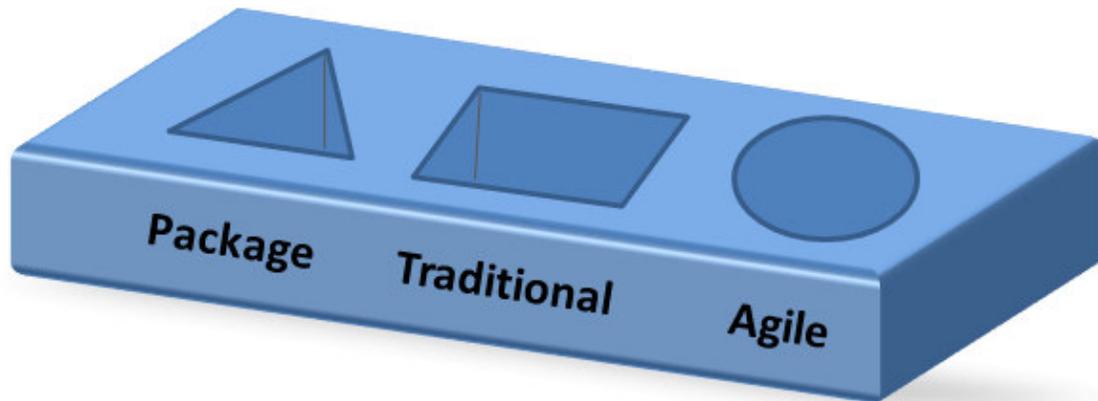
Agile Suitability Filters

Why wouldn't you always use agile methods? Well, shock-horror, maybe they are not always the best approach for the circumstances! So how do we decide?

It is tempting to think that all projects should be handled in an agile way. Indeed, I am convinced that all projects would benefit from the improved collaboration and communications encouraged on agile projects. However, collaboration and communications are just two attributes of agile projects and we must consider wider parameters that influence project success or failure.

I am working with the assumption that the end result we are looking for is a successful project with satisfied stakeholders. It would be better to have a successful traditional project than a failed agile project. Others might disagree, but I'd classify them as purists rather than pragmatists.

Our choices typically include using a Package that may or may not require customization and integration, running the project using a Traditional (single pass waterfall) approach, or running the project as an Agile project.

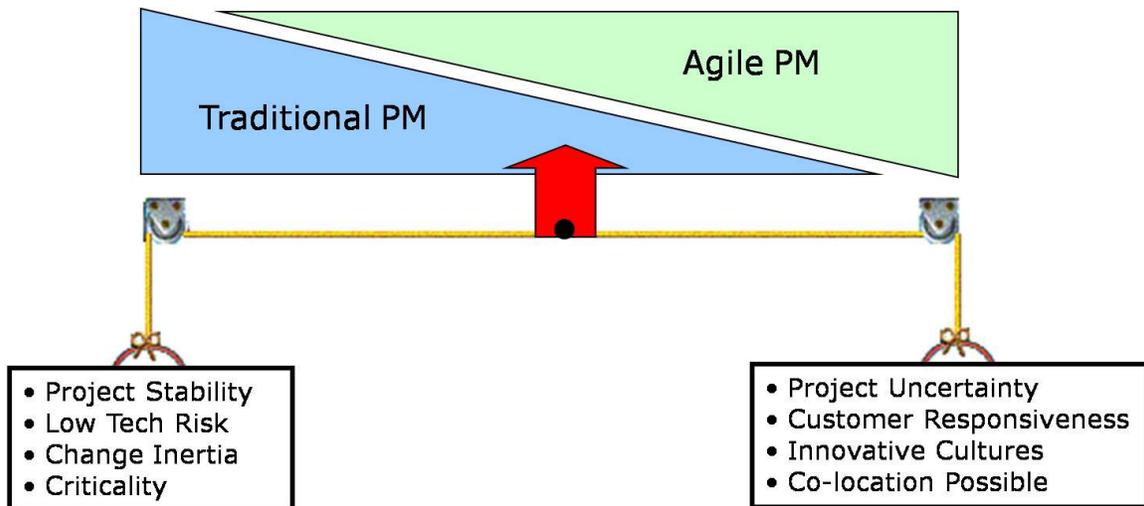


So, how do we identify and assess the project characteristics that determine what “shape” project we have? Fortunately there is a wealth of good research to draw on and build upon. This post outlines some popular and some less well known agile suitability tools:

1. The Slider
2. DSDM Suitability Filter
3. Alistair Cockburn's Project Criticality and Team Size
4. Boehm and Turner – Radar Chart
5. Dave Cohen's Agile Factors
6. The Organizational Suitability Filter
7. The Methodology Bias Hammer

1) The Slider

The Slider is a simple model I created to illustrate that the Agile / Traditional choice is not necessarily binary.



The model is fashioned after an old fashioned car heater control mixer, allowing for No, a Mixture, or a Fully agile approach as indicated by the position of the arrow. Characteristics that would pull the pointer one way or another are depicted as weights.

Some project characteristics such as "", "", and "" are a good fit agile and would pull the pointer towards the agile scale. Other characteristics such as "", "", and "" make agile introduction more problematic and might instead favour a more traditional approach.

Traditional and agile approaches can be successfully combined to effectively address hybrid or hard to classify projects. "You can use all agile some of the time and some agile all of the time."

2) DSDM Suitability Filter

I had the good fortune to be involved in the creation of DSDM back in 1994 and remember the early days of using the Suitability Filter Questionnaire. It was quite primitive then and has remained a simple list of Yes/No questions to answer. The basic idea is to check conformance to project characteristics that favour agile development such as:

1. **Acceptance of the agile philosophy before starting work** - timeboxing, user involvement, iterative development, etc
2. **The decision making powers of the users and developers in the development team** – acceptance of empowered teams
3. **The commitment of senior user management to provide significant end-user involvement** – availability of user resources
4. **Incremental delivery** – acceptance that this is acceptable and desirable
5. **Easy access by developers to end-users** – on site customers
6. **The stability of the team** – keeping at least a stable core base to support tacit knowledge rather than requiring documentation
7. **The development team skills** – having a skilled team with good communication skills.
8. **The size of the development team** – small teams to leverage face to face communications and minimize communication and documentation costs.

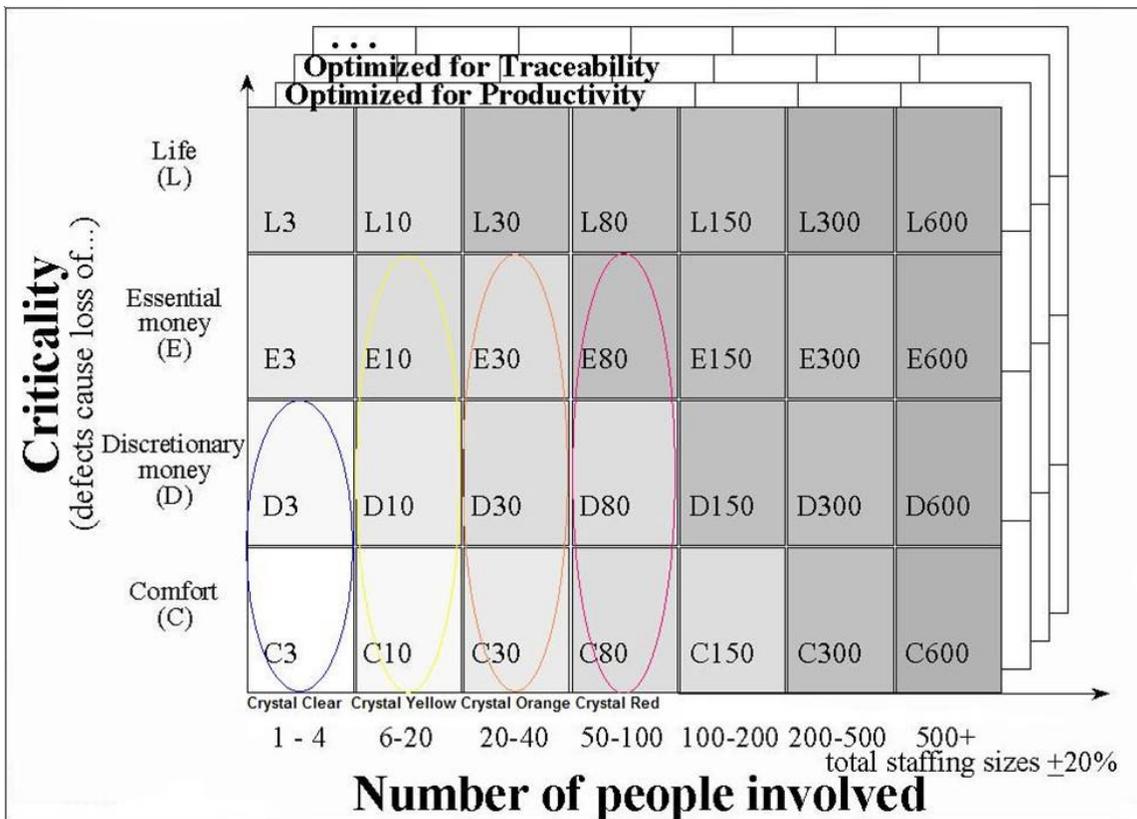
9. **A supportive commercial relationship** – trust and collaboration over contract negotiation.
10. **The development technology** – supports incremental delivery, rapid prototyping and refactoring

The presence and absence of these attributes provide a good handle on the likely ease of acceptance to using an agile approach. Negative answers do not automatically mean that agile will not work, rather they highlight potential risk areas that need to be managed. For instance, “No” answers around “user availability” may mean that we a risk of poor availability and so we need to do something in the project start-up to try and assure their availability. “No” answers to the majority of the questions however should be a red flag. With so many challenge areas is this really set up for success?

You can download a copy of the DSDM Suitability Questionnaire below. Answering the questions can provide valuable insights for any type of agile methodology.

3) Alistair Cockburn’s Criticality and Team Size Factors

Alistair Cockburn’s Crystal family of methods are based on project fit and suitability. Using the characteristics of System Criticality and Team Size Alistair divided his methods as shown below.



On the X axis we have team size starting with very small teams of 1-4 people on the left and progressing to large projects of 500+ people on the right. On the Y axis we have system criticality expressed as what failure of the system would result in. If a games and word processors crashes then we lose some time. If a billing system goes down the Discretionary or even Essential monies could be lost.

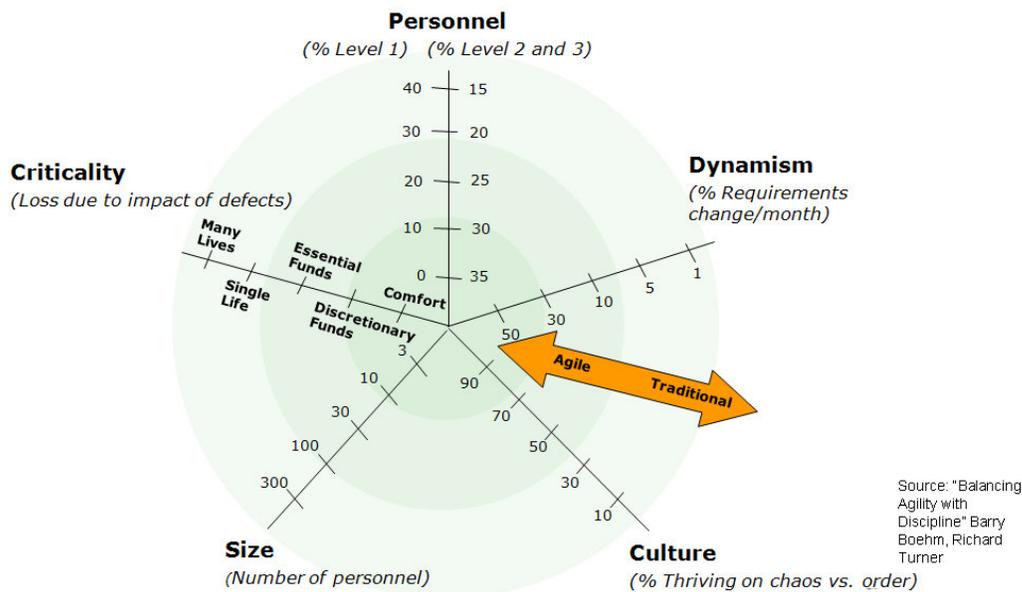
We can see that Crystal Clear is an agile methodology designed for small teams tackling projects up to a criticality of Discretionary funds. Crystal Red on the other hand has more detail, rigour and controls and is targeted at teams of 50-100 people working with systems up to Essential Money in criticality.

Considering the projects team size and criticality is a good way to assess the suitability of agile. Agile methods have been proven to work well on large teams and even life critical systems but this takes a lot more skill, effort and augmentation. They are much easier to implement on small, non life critical applications.

4) Boehm and Turner – Radar Chart

Barry Boehm and Richard Turner in their book “Balancing Agility and Discipline: A Guide for the Perplexed” (a great book with a poor title since agile is disciplined) created a nice visual way of assessing the project characteristics that they felt determined suitability to an agile approach.

http://www.amazon.com/Balancing-Agility-Discipline-Guide-Perplexed/dp/0321186125/ref=pd_bbs_sr_1/105-1482500-6010820?ie=UTF8&s=books&qid=1182174011&sr=8-1



The idea is that the project is assessed along five attributes and the scores are plotted on the radar (spider) diagram. Scores towards the centre indicate a good fit for an agile approach, while scores towards the outside indicate a better fit for a more traditional approach. The characteristics measured are:

Personnel – this measures team skills, borrowing from Alistair Cockburn’s Level 1 (beginner), Level 2 (intermediate), and level 3 (expert) developers scale. For agile projects to go smoothly; it is easier with a low proportion of beginner developers and a high proportion of intermediate and expert level users. This is reflected by the mixed axis score towards the centre of the graph (agile) there is a low percentage of beginners (level 1’s) and a high percentage of experienced (level 2’s and 3’s). If you team has a higher percentage of beginners (and therefore a lower percentage of more experienced staff) then perhaps a more traditional approach (coding to specifications) might be easier.

Dynamism - this is a fancy term to assess the likelihood of changes. How dynamic (changing) is the project, what percentage of the requirements are likely to change during the project? If 50% of the requirements are likely to change then we are towards the centre of the graph in the agile zone. If only 1% of the requirements are likely to change then we are near the outside in the traditional zone. This is not to say you can not use an agile approach, but perhaps (for this characteristic at least) planning the work and then working the plan might actually work.

Culture (Thriving on chaos vs. order) – what is the temperament of the organization? Is it one that can accommodate, even feed off of change, or is it one that leans on the familiarity of order and tradition. Getting buy-in for agile in a very ordered environment can be challenging as ideas such as emergent requirements, empowered teams, and servant leadership appear counter to the culture and values of the organization. It can still be done by appealing to their wish to reduce project uncertainty, but the way agile approaches are introduced needs more artful consideration.

The next two characteristics “Team Size” and “Criticality” Boehm and Turner are from Alistair Cockburn’s Crystal family of methods.

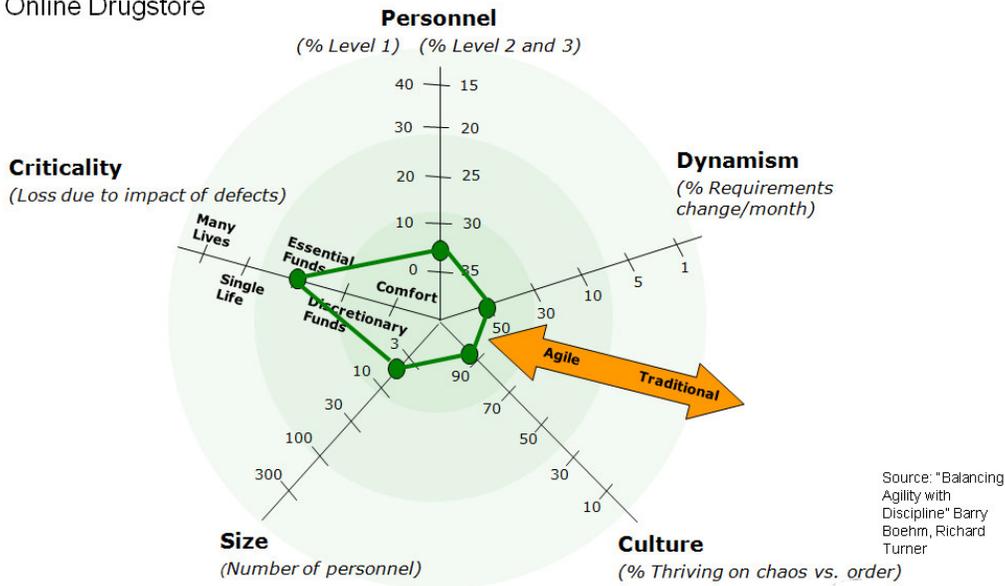
Team Size – agile methods are easier to introduce, execute, and manage with small teams. This is not to say large agile teams can not work, just it is much harder to accomplish. Teams of less than 10 are a great fit for agile as this number is easier to physically co-locate, they can communicate via face-to-face discussions, they can support unwritten (tacit) knowledge by conversations, and facilitate simple visible tracking systems. As team sizes grow, supporting these agile principles require additional techniques to scale effectively. It can be done, but takes more work and skill. In contrast, the hierarchical structures of traditional projects were made to scale upwards with the natural proliferation of committees, documentation and matrices.

Criticality (system failure results in loss of...) – This is more contentious one, it refers to the consequence of a system failure. Boehm and Turner take Alistair Cockburn’s model and assert that agile is more suitable for trivial applications where failure of the system results in a loss of convenience (such as losing time if a game crashes or work if a word processor fails) but less applicable for mission critical or life critical applications.

I can see where they are coming from, having worked on military projects; there is always a strong traceability from tests to requirements and from requirements to specifications that can be automatically checked in both directions. Modern agile testing frameworks can provide similar functions, but out-of-the box, many agile approaches do not mandate this level of detail. However this is not to say it can not be added and I would recommend an iterative agile approach to tackle and test architecturally significant functionality early and often in a project for building life-critical applications. Start with agile and add the layers of additional verification as demanded by the system criticality.

To illustrate how the radar chart is used I have shown an example of an online drug store project I managed while working at Quadrus Development.

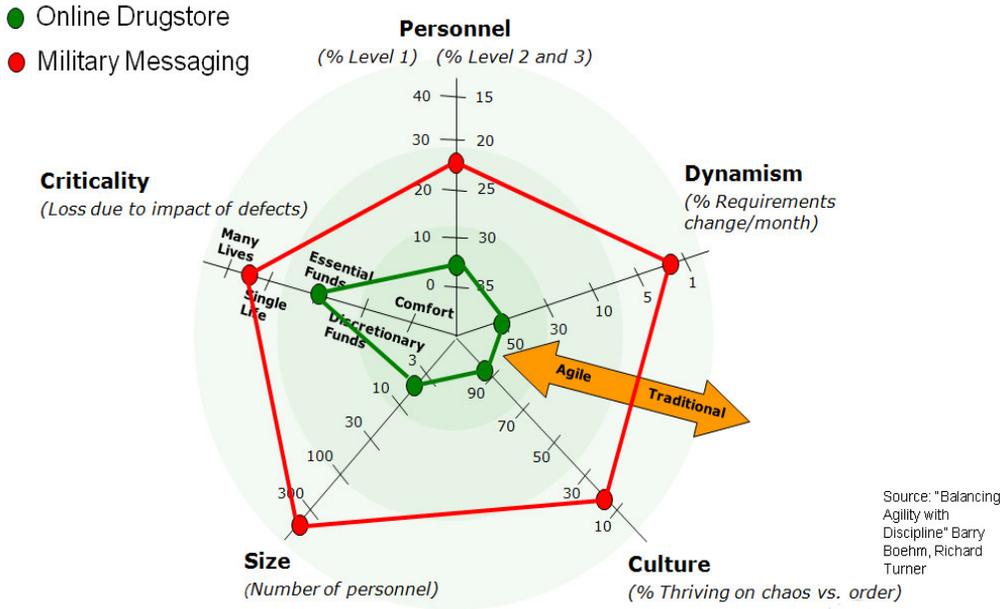
● Online Drugstore



The project was to develop an online drug store to sell cheap Canadian prescription drugs to (primarily) US customers. The sale of these drugs is a contentious subject in Canada as well as the US and as a result the industry is characterized by swift regulation changes and fierce competition. As a result we faced extremely volatile requirements with major changes week on week. We used very short (2-day) iterations and weekly releases to tackle the high rates of change.

As shown on diagram, we had an experienced team of level 2 and 3 developers that worked on very dynamic requirements in a culture of high change approaching chaos. The team size was small (5 people), but the system criticality was fairly high with essential funds for the pharmacy at stake. The approach was very successful and extremely agile.

Contrast this with a project I worked with while at IBM Global Services. It was a military messaging system that had already been running for 5 years when I joined the team.



There was a mixture of skill levels, requirements were locked down because changes impacted so many sub-contractor organizations, and the culture was one based on specification and control. The project was huge with over 300 people from IBM alone and the criticality was high as potentially life-critical information was being passed. Parts of the project could have been carved off and run as agile projects, but at the heart of the initiative was still a monolithic monster.

Reflections on the Boehm and Turner model

I occasionally teach an agile project management course for the University of Calgary and one exercise we do is to think of additional vertices to add to the diagram. In other words what other factors should you consider when determining the fit for agile. A popular answer is "Testing Effort" if testing an increment is expensive in terms of effort or time then you probably can not afford to do it very often. (Or you want to find other ways to simulate testing it. Car makers crash test thousands of designs iterations using computer simulation rather than testing with real vehicles. If testing is costly in time or effort we similarly have to find a way (via stubs and test harnesses) to make it easier.)

5) Dave Cohen's Agile Factors

Cohen, Lindvall, and Costa wrote a good introduction to agile in the publication "Advances in Computers" back in 2004. In it they identified the following factors as precursors to agile acceptance:

1. The culture of the organization must be supportive of negotiation
2. People must be trusted
3. Fewer, but more competent people
4. Organizations must live with the decisions that developers make
5. Organizations must have an environment that facilitates rapid communication between team members

Item 4 can seem an odd one and you may feel that the business should ultimately direct development. However, when you read the paper this comment is made when discussing the need for competent developers and trusting them to make empowered decisions about design rather than business functionality

These are good factors and I like the fact that the organizational factors are being assessed more. This is where I feel a lot of influence or resistance lies. We can look at the project in isolation and think it is a great candidate for agile, but if the organization has a different culture or directive then we are missing an important piece of the puzzle.

6) The Organizational Suitability Filter

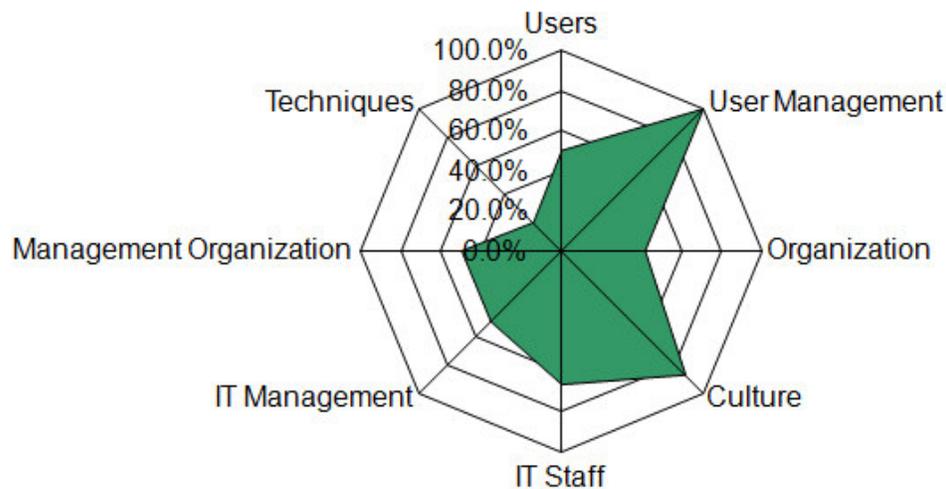
The DSDM Consortium published an Organizational Suitability Filter questionnaire as a white paper to accompany the Project Suitability Filter. Unfortunately the white paper did not get the same publicity as the regular project suitability filter which is a shame because it provides useful insights in to how well suited a company is towards adopting agile working practices.

The questionnaire comprises of 46 questions divide across the following categories. I have listed the categories and para-phrased the gist of the questions below:

1. **Users** – do we have the right users? do they know what they are talking about?, and can they make decisions?
2. **User management** – do they understand iterative development? will they make their people available? and will they trust/empower them to make decisions on behalf of the group?
3. **Organization** – what is the relationship with IT like? And can they accept agile contracts?
4. **Culture** – Is there an open culture? are people prepared to try new approaches? and is an 80% solution first acceptable?
5. **IT staff** – do they know enough about agile? can they speak effectively to the users? and what is their relationship with the user community like?
6. **IT management** – do they understand agile methods? and are they willing to change project management standards?
7. **Management organization** – are they procedure driven? will stakeholders be available to participate? and does the business have the appetite for incremental delivery?
8. **Techniques** – is there a history of BDUF?, and are the build and test tools available to support the technical environment?

I have not really done the questionnaire justice; it provides good insights into the likely agile adoption risk areas and takes only about 45 minutes to complete. You can download a copy here:

Once you have completed the questionnaire you can use the attached assessment spreadsheet to help interpret the results.



In the radar diagram above we can see some sample results from the Organizational Suitability Filter Questionnaire. The graph plots risk, so a high score represents a high risk in that area. In our sample “User Management” and “Culture” are the high risk areas.

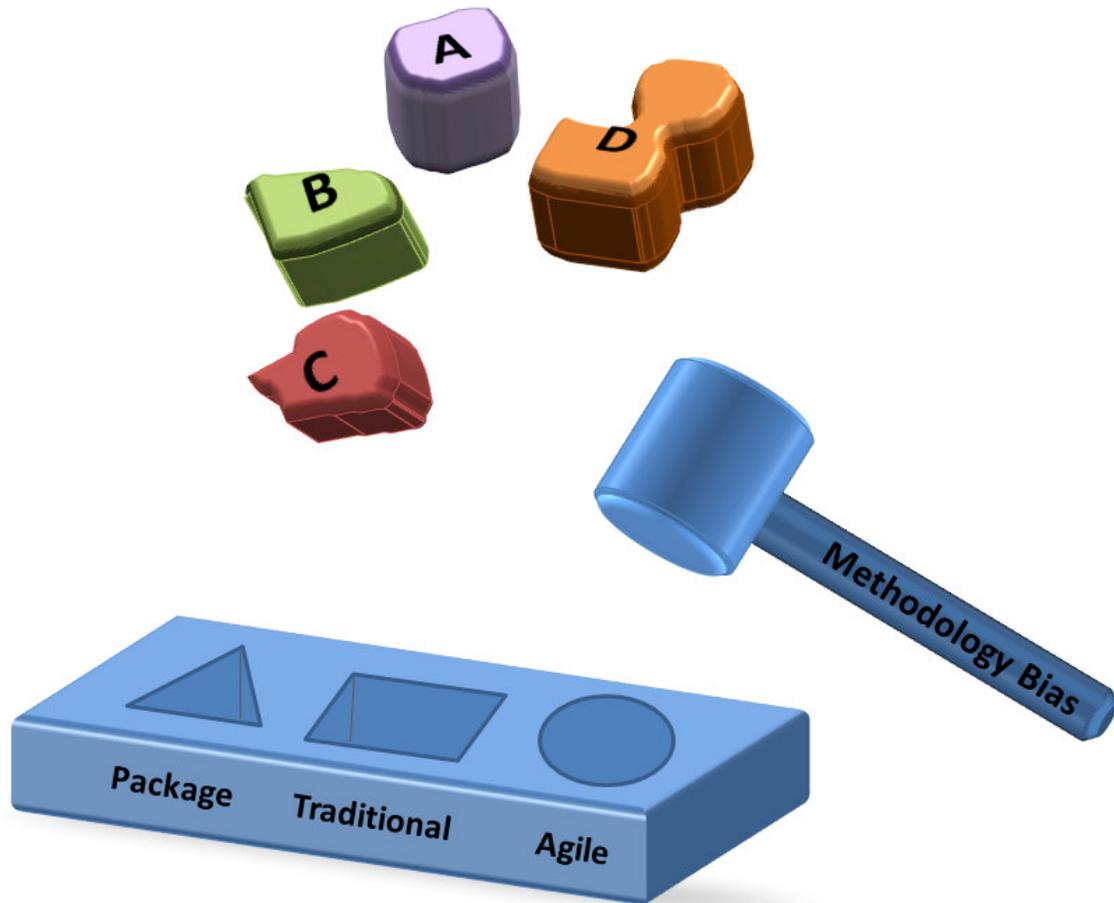
Like with the Project Suitability Filter, negative scores or high risks does not automatically mean that you can not use an agile approach. Instead it identifies risk areas to be understood and reduced. Learning about likely risk areas is a positive thing because we can proactively do something to reduce them (“to be forewarned is to be forearmed”).

Non-Project Factors

These organizational factors are a critical consideration, even if the project is a great fit for agile in theory, if management or other stakeholders are against that approach, then its application carries significant risk. Also, if the project manager just does not believe agile methods will work, then I am convinced they can prove themselves correct and fail using an agile approach. If your heart is not in it then project obstacles can appear as vindication of process weakness, not setbacks to overcome.

I call this “Methodology Bias” and I think it is the most significant factor to be evaluated. You could call it “Methodology Preference” if you like, but that hides the unconscious side of a bias that influences our decisions subconsciously. I have a strong pro-agile Methodology Bias and will employ agile principles on package selection projects and projects with little chance of variation (such as “roll out this update to 500 users”). That’s just my approach; I like collaboration and consensus building followed by a gradual evolution and check-pointing approach.

I visualize Methodology Bias as the hammer we use to make projects fit our methodology of choice.



If a team has a strong desire to succeed with an agile (or any other) approach they will likely find inventive ways to make their method work. We can use tools such as Boehm and Turners radar chart to determine the ideal characteristics for an agile shaped project, but we must never forget the size and persistence of the Methodology Bias hammer that can make even the most unlikely looking projects “fit”.

In the diagram above Project A looks a good fit for our Agile selector, without too much modification it should be a straightforward agile project. Project B looks like it may better suit a traditional approach. Project C is more problematic, is has a package component in it and the rest could be traditional or agile. Welcome to the real world, things are complicated; perhaps a hybrid approach is required here with both a package implementation and development project occurring in parallel.

Project D seems to have a traditional part and an agile part; perhaps we can split this into two sub-projects and handle it that way? This is a valid approach that is often overlooked. (Alternatively we could just keep beating it with our favourite hammer until we convince ourselves and others that we have a fit for our preferred method).

All these different suitability filters, which should I use?

Well, it would not be very adaptive or agile to mandate a single approach or strict progression between models. Instead I'd suggest you get familiar with a variety of them and then select what you think would work best in your environment.

Be aware that theoretical fit and practical fit are often quite different. Acknowledge the presence of the Methodology Bias hammer and learn how it influences choice and success. Finally remember that these are just tools and they are not a replacement for thought and dialogue with the project stakeholders. Rather than using them in isolation, use them to start conversations about agile suitability and build consensus around the method of choice.

Mike Griffiths is an independent consultant specializing in effective project management. Mike was involved in the creation of DSDM in 1994 and has been using agile methods (Scrum, FDD, XP, DSDM) for the last 13 years. He serves on the board of the Agile Alliance and the Agile Project Leadership Network (APLN). He maintains a leadership and agile project management blog at www.LeadingAnswers.com.