FileMaker + Agile

Quickly solving problems

Audience

This paper is intended for FileMaker Developers and FBA partners who are considering how to use Agile development with FileMaker. It is also intended for Citizen Developers who are engaged in a project with their department or IT Team members within an agile-based company who are considering FileMaker as a development platform.

Introduction

FileMaker by it's nature is a perfect platform for adaptive (Agile) development methodologies, which grew out of the Rapid Application Development theory. Agile is centered around repetitive cycles of Speculate, Collaborate, and Learn, which provides an alternative to the traditional predictive (Waterfall) cycles. The FileMaker platform allows teams to identify and solve a business problem and develop a cohesive solution quickly. It's characteristic blank canvas that anyone can build in quickly and adapt to the team's input through repetitive cycles embodies the Agile methodology. The purpose of this paper is to provide an overview of Agile development methodology and how the FileMaker platform embraces Agile. For any tool, it is important to understand when the tool can be used with a platform, so this paper will also cover what factors may become constraints when pairing Agile with FileMaker.

What is Agile?

Agile (adaptive) is a methodology that encourages teams to develop software by collaboration; producing and deploying software applications within a relatively short period of time with the goal of solving the problem not simply building a solution. It focuses on building quickly with early collaboration with the client/internal department and continually collaboration throughout the building and testing phases[1]. Because Agile handles change easily without derailing a project, Agile excels at solving a problem because iterations guarantee the product is the right solution in the end. The methodology is best defined by the original Agile Software Development Manifesto.

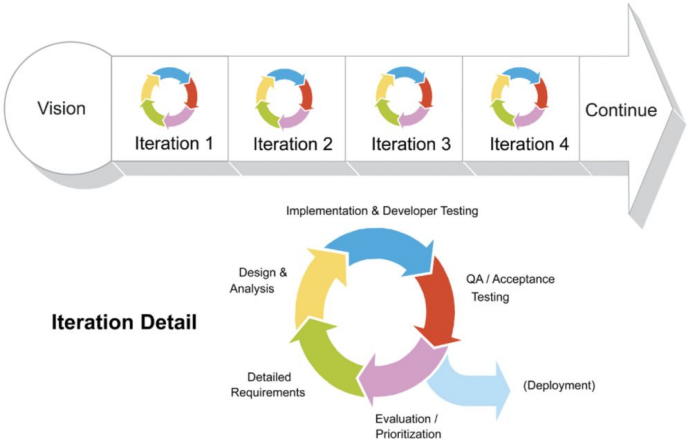
> "We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> - **Individuals and interactions** over processes and tools
> - **Working software** over comprehensive documentation
> - **Customer collaboration** over contract negotiation
> - **Responding to change** over following a plan

[1] http://agilemethodology.org/

That is, while there is value in the items on the right, we value the items on the left more."

Adaptive software development is a rapid application development style that utilizes sprints and iterative reviews to build a solution quickly. It organizes the work into functional groups, creating a product mentality rather than project mentality. It requires a core team working closely together to determine product goals, testing, and criterion of success of individual sprints. The core team typically includes a team lead (Agile product manager), a subject matter expert, multiple stakeholder, and multiple developers (including citizen developers). The core team works in an iterative review process, focusing on one function, and product, at a time.

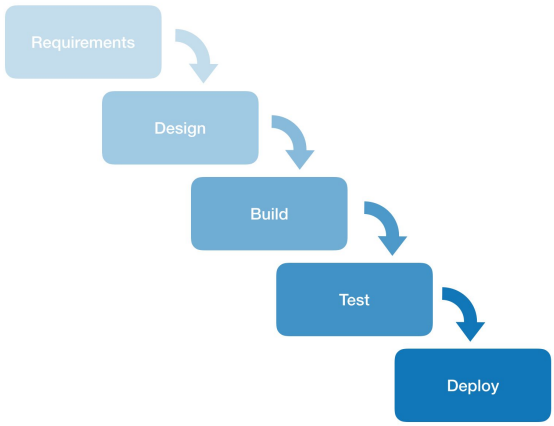
Iterative review is a development process where there are repetitive cycles of building and reviewing. At the beginning of an iterative review, the entire product is reviewed and a function or set of functions are chosen to focus on. The core team develops criterion of success for that iteration. A sprint is launched for quick, highly focus building time to complete the function(s). Each sprint is typically 1-2 weeks and often has a quick (5-15 minutes) stand-up meeting each day to go over what has been accomplish, what roadblocks are impeding work, and what is left on the agenda for the day. In Agile, product reviews and testing are done at the end of each sprint to determine if the current build meets the success criteria or if another iteration is needed. If successful, the team moves to the next function. Adaptive design will have a countless number of iterations[2] until the final product is accepted.



Contrary to Agile, Waterfall design is a sequential process, where one block is completed before moving on to the next, and it is not revisited. In software development, it means that development of the entire project is finished before reviewing and testing, rather than by individual function group. Like Agile, Waterfall will also have a core team, however, the core team does not meet regularly. The team meets a few times to develop criteria for success and agree on a design. Then the developers will work for several weeks on their own to develop the product, typically with little communication. Testing is the only time that errors and bugs might

[2] http://scrumreferencecard.com/scrum-reference-card/

be found. This is a huge disadvantage compared to Agile, where errors or scope change could have been brought up earlier in an iteration.



Agile lends itself to be more flexible, due to the iterative style, and therefore incorporates a collaborative style of everyone working together to continue to refine requirements - which are loose to begin with- build, and test. Waterfall is very structured, without any repetition, and therefore does not encourage collaboration. It typically starts with a few meetings to kick off the project, agree on scope - which usually comes with well defined requirements- and documentation, but then building is done without much additional input.

How FileMaker Embraces Agile

According to the Agile Manifesto, there are 12 Principles of Agile. FileMaker in general fits well into these principles and is recognized as an Agile platform.

The first few principles highlight that Agile development should be quick and flexible. An Agile platform needs to be able to meet the needs of the client quickly no matter how often the needs change. The FileMaker WYSIWYG interface allows for flexible/rapid UI development allowing for a rapid initial build, and allowing for constant change with minimal impact to the budget or the project timeline. This guarantees that FileMaker can deliver on the highest priority- “to satisfy the customer through early and continuous delivery of valuable software.”

Often the Product Owner will begin a custom software development project with a goal and a vision but as the project progresses the original problems are no longer high priority and other solutions are needed. The team will need to be able to work together to define and refine those goals. The cohesive layers of the FileMaker platform (data storage/logic/user interface) have the capability to “welcome changing requirements, even late in development” without requiring a complete rebuild.

Additionally, Agile platforms need to be able to meet the constantly changing business needs after the product is deployed. “The sponsors, developers, and users should be able to maintain

a constant pace indefinitely." FileMaker shines in sustainable development after building is complete. Teams using FileMaker will find that once the initial application is deployed to the end users, efficiencies might be found that can improve the workflow. These after deployment changes can typically be done rapidly and often can be done by "Citizen Developers". FileMaker has a strong community support system to help Citizen Developers without pulling a large team together again, "[giving] them the environment and support they need" to constantly solve business problems.

The FileMaker platform adopts many of the principles of Agile. The platform allows for an easy initial build and can adapt to constant change throughout the product build as well as after. FileMaker focuses on customizing a solution to meet the client's needs and because it can adapt quickly to change, FileMaker teams welcome constant feedback from the client to guarantee satisfaction.

12 Principles of Agile

12 Principles of Agile[3]	Advantages of FM	Challenges of FM
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	Ability to build fast	Multiple environment deployments can be complicated and lengthy
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	The WYSIWYG interface allows for flexible/rapid UI changes; No Compiling of code across the team	
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	FileMaker is built specifically to this principle; the all-in-one platform and UI tools allows for fast development	Multiple environment deployments can be complicated and lengthy
Business people and developers must work together daily throughout the project.	The WYSIWYG user interface makes it easier for non-developers understand The rapid deployment of FM provides the ability to show daily progress in the check-ins	Need to be circumspect in allowing the non-developer access to the layout editor

[3] http://agilemanifesto.org/principles.html

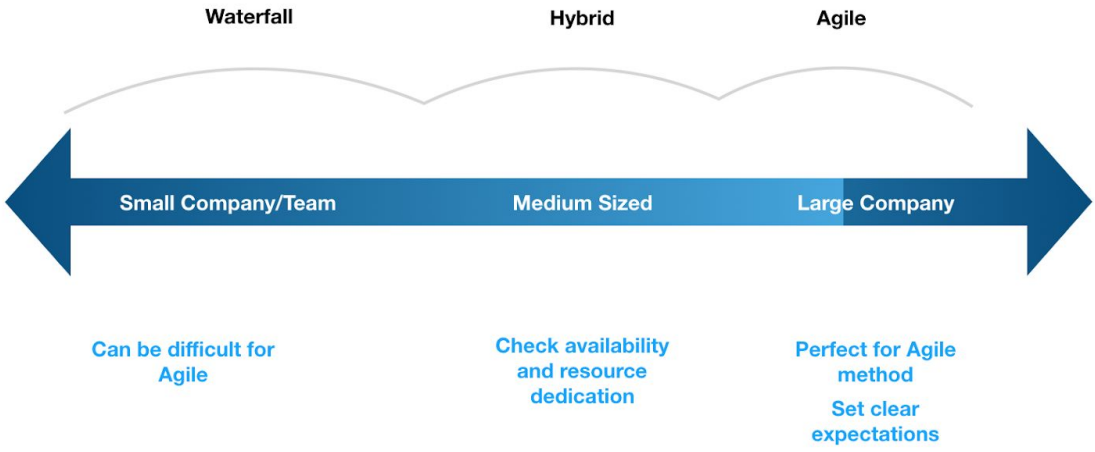
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	FileMaker allows for a “Citizen Developer” / DIY Approach	The result is only as good as the skill or knowledge of the team
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	Pair-programming is relatively easy in FM; the user interface works well with screen-sharing and face to face conversations; it allows teams to share ideas and make live interface refinements	There can be a perception that the coding is as easy as changing the interface
Working software is the primary measure of progress.	FM is a RAD environment	Limited by how much you can do with the all-in-one platform
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	FM provides an environment that encourages users to imagine how the application can be improved to make their workflow more efficient	
Continuous attention to technical excellence and good design enhances agility.	FM is easy to design	Dependant on the skill and expertise of the team
Simplicity--the art of maximizing the amount of work not done--is essential.	These principles are analyzed during the development stage and not specific to a particular platform	
The best architectures, requirements, and designs emerge from self-organizing teams.		
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.		

Considerations for Agile with FileMaker

Throughout this paper, it has been proven that FileMaker is an agile platform and FileMaker projects can be run as agile. However, there are other considerations besides what platform is being used for a development style to be chosen. The main constraints for any development project are resources, scope, time, and budget. As with any methodology, it is important to understand the potential challenges that might come with applying the methodology, in order to

adapt the project and use the methodology to its optimum potential. Agile's ideal environment is a project with well defined resources, a loose timeline, and a flexible budget. However, different levels of resources, timelines, scopes, and budgets can still work with FileMaker and Agile by understanding how to set up the project to consider the potential challenges.

Resources



Agile requires teamwork and close collaboration, but often, a company has a lot of moving variables. Companies might not have enough resources, those resources might be working on too many projects to provide enough time, or the resources can change and new team members have to be brought up to speed. If change is expected, extra documentation can help new team members adjust quickly.

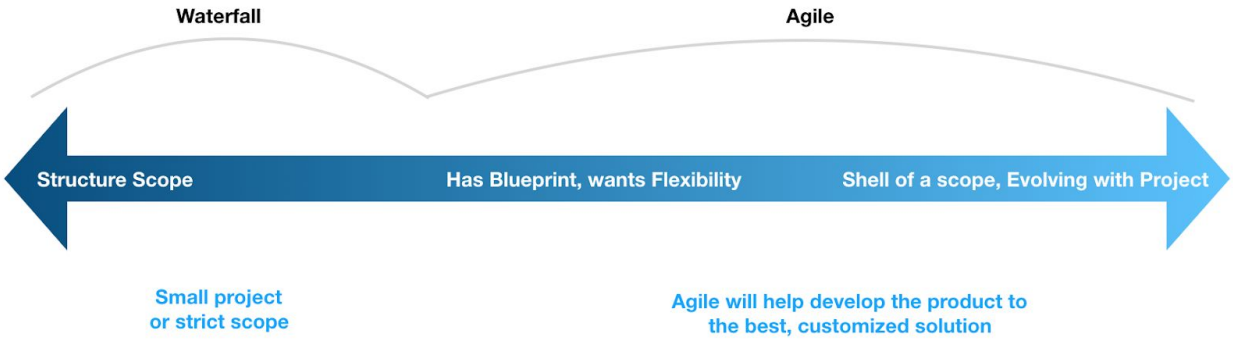
Team size can impact how Agile is applied to a project and is therefore a constraint. Agile works really well with larger teams, around 7-9 team members with several of them being developers. FileMaker cannot compile code together like other software, but there can be multiple developers if it is a big enough product. Therefore, FileMaker teams usually have only a few developers. Agile can still be applied, but sprints look differently because daily stand-ups tend to be shorter and more of a check-in style. Additionally, small teams have to be cautious of groupthink that can cause critical goals to be overlooked.

If a company or department is too large, there may be too many decision makers who need to be involved in making decisions. In this case, there may be too many stopping points to wait for decisions to be made, making the project run more waterfall than agile.

Agile is designed to be used with a team that sits together, to rapidly get the answers to make changes. Often, FileMaker developers are remote so it could be a challenge with teams. Some teams adapt by using video calls to help with communication.

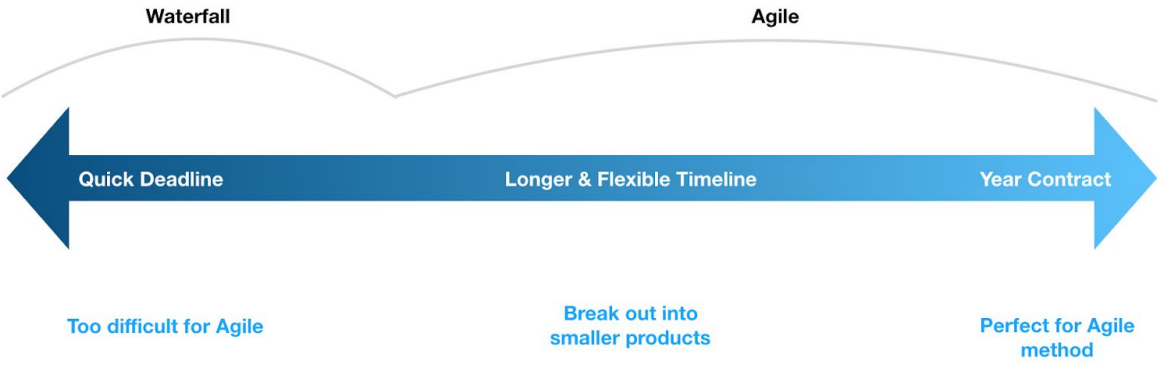
FileMaker and Agile can work together in different types of team, but it is important to get clarity to set up the Agile framework from the start.

Scope



FileMaker with Agile lends itself to continual refinement of requirements. FileMaker is easy to build in and helps set an environment of iterations. Projects work most efficiently when the client and developers can work together to articulate a clear set of requirements, and continue reviewing them during each iterative process. If a strict scope is outlined and detailed requirements have been established without flexibility, it may be a constraint for running an agile environment. There may be opportunity to apply iterative reviews during the build to take advantage of some of the Agile methodology, but it should be carefully considered whether that is the best approach with the team, considering the other constraints.

Time

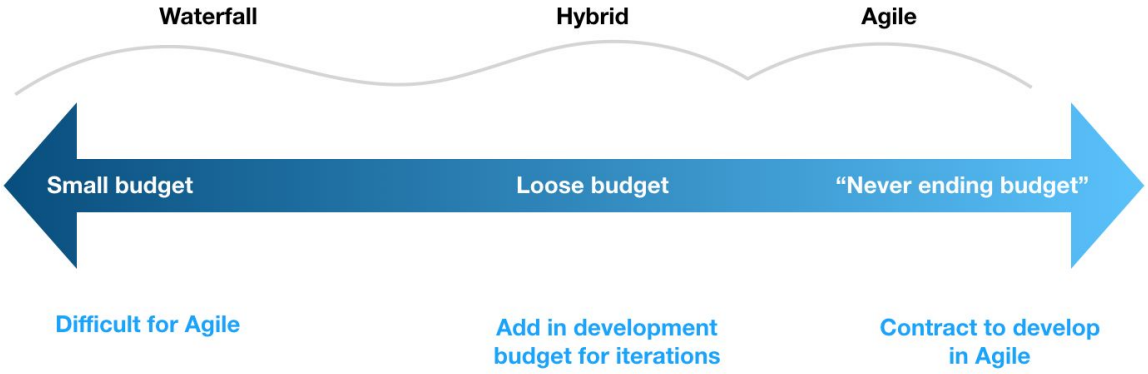


Agile within FileMaker is streamlined when the client has time to dedicate to as many iterations as it takes to get what they want. In an ideal environment, the vendor signs a contract with the client for an annual number of hours to get the work done.

Products on a short timeline can struggle with an Agile approach if not considered upfront. Often, a client or internal department wants to understand exactly how long the project will take, and be held to that commitment. If this constraint is given, the project will fall more in line with waterfall than agile.

Fortunately, the FileMaker platform UI allows for quick changes.This works in favor of any timeline, and makes sure there is efficient use of however much time is offered. If a project can use iterative reviews, it will likely save time in the end by catching any necessary changes earlier rather than later.

Budget



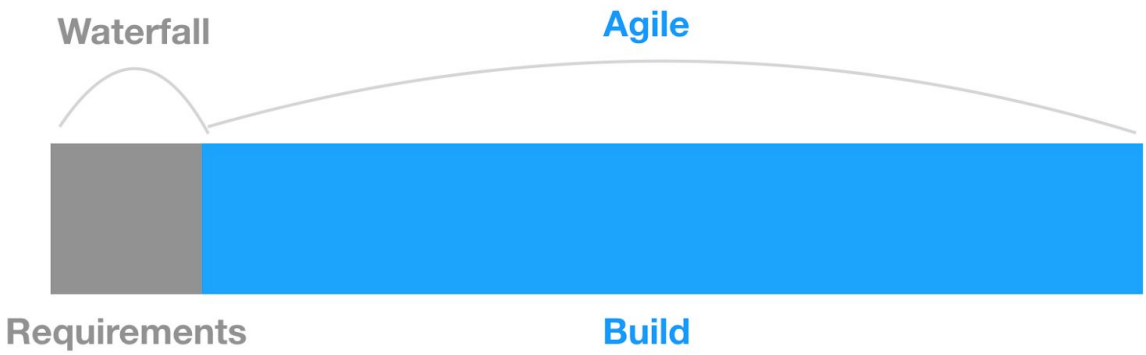
In Agile, it might be possible to estimate a developer's time, but the number of meetings and reviews and hours of testing involved for each sprint cannot be estimated. Estimating a firm budget can become challenging. If the client has a fixed, limited budget then the team will be limited how to apply Agile. FileMaker is designed to be Agile with an easy to edit interface, so a team on a stricter budget can limit the number of iterations in the development process but still run Agile.

Hybrid Approach

Typically, a FileMaker team is not going to be building in the ideal environment; there will be some constraints impeding a perfect application of Agile methodology. Because FileMaker and Agile work well together when the iterative cycles of Agile can be taken advantage of with the rapid application environment in FileMaker, a team might have to alter the project to take

advantage of both of these concepts. Therefore, most FileMaker and Agile projects will run as a hybrid approach.

While there are many hybrid approaches, the most likely scenario will be a waterfall approach to gathering requirements, and an Agile approach to the build.



The requirements gathering process typically runs in a waterfall methodology because a client wants to know exactly how much time and money will be required. Stopping to view the entire project in detail forces a constraint that no longer meets agile principles. In additional, the client will likely want to know a billing cadence with specific tasks associated per billing cycle. Again, this constraint pushes this portion of the project into waterfall.

The build can be run in an Agile environment. Once the requirements have been collected, the project is on a clear path. The optimal way to then move into the Agile environment is to create a minimum viable product (MVP), and divide the rest of the requirements into segments that can be built onto the MVP in iterations. The team should agree on the segment that will get the company up and running with what they need as soon as possible. Fortunately, no matter what the timeline, FileMaker is designed to be iterative and flexibility to constant change so it can work quickly on a short timeline. It can be built upon just as easily, so when more budget, time, and/or resources becomes available, it can be expanded. And, building in small product launches diminishes any scope creep, because additional scope can be added to a later product. Breaking the product into small products allows for an Agile approach to be applied to the smaller product, take advantage of the iterative review, and continue iterations on the bigger product.

Conclusion

It has been shown that FileMaker embraces the Agile methodology well, especially with how easy it is to design in and make changes in, using an iterative approach. An Agile company will

find it easy to pick up the FileMaker platform and build initial products and continually enhance upon those products.

FileMaker and Agile work well together, but due to constraints like number of resources, available time, scope definition, and budget, not every project will include all aspects of Agile. The concepts of close collaboration, quick development, and repetitive reviews can always be applied to a FileMaker project to make it Agile. A recurring challenge is budget and time constraints which typically constrain how many iterations there can be during the development process. When necessary, the product can have a waterfall approach to gathering requirements and creating a direction, and then be segmented into smaller products to reduce constraints and build in a fully Agile environment. Then, FileMaker and Agile can work well to solve any business problem.