Agile Advantages for Software Development and Your Business

Posted on February 26, 2020

WHAT IS AGILE?

Agile methodology is an iterative, time-boxed, people-oriented and result-focused approach to software delivery that builds software incrementally from the start of the project, instead of delivering it all at once near the end.

Agile methodology is aligned with the values and principles described in the Agile Manifesto for Software Development. According to this principle, requirements, plans, and results are evaluated continuously; teams have a natural mechanism for responding to change quickly.
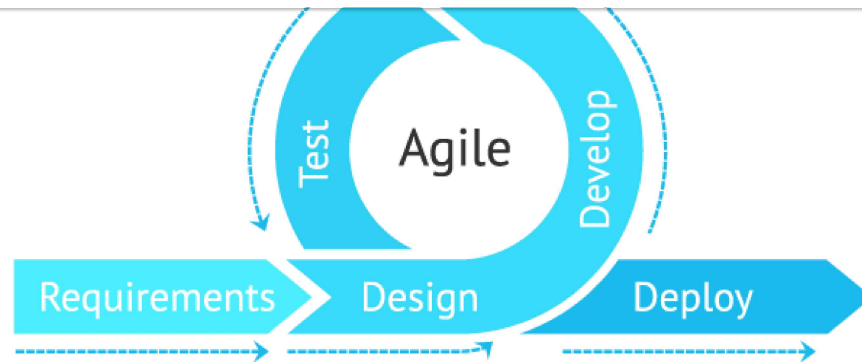
AGILE ADVANTAGES IN SOFTWARE DEVELOPMENT

Agile methods began appearing in the early 1990s as the software industry exploded. Traditional software development approaches appeared to many software developers to be too rigid and resulting in over-budgeting and delays. Then in February 2001, a group of 17 independent practitioners, leading software engineers met in Snowbird, Utah to discuss an alternative and lightweight approach to software development and to develop the basic principles of Agile philosophy.

The leading developers issued a symbolic [Manifesto for Agile software development](#) that consisted of three main principles:

⇒ **Individual and team interactions over processes and tools.**

Software development is a human activity and the quality of interaction between people is vital. Tools are an important part of software development, but making great software depends much more on teamwork, regardless of the tools team may use.

⇒ **Working software over comprehensive documentation.**

Documentation can be a great resource for users and coworkers. However, the key goal of software development is to develop software that offers business benefits rather than extensive documentation.

⇒ **Customer collaboration over contract negotiation.**

Development teams work closely, communicate with the customers frequently. Only by listening to and getting feedback, teams will understand what stakeholders want.

⇒ **Responding to change over following a plan.**

Changes are a reality in software development, a reality that your software process should reflect. A project plan shall be flexible enough to change, as the different cases demand.

Following these Agile software development principles, developers use short, regular and controllable iterations of work called Sprints. In one short Sprint (2-4 weeks), a full development cycle is performed. It includes analysis, design, coding, testing and product release. Moreover, developers release a working software solution at the end of each Sprint. Finally, every next Sprint is an extension to the already working software solution that brings developers closer to a final version.

Since then, many agile frameworks have emerged such as Scrum, Kanban, Lean, and Extreme Programming (XP). Many agile teams combine practices from a few different frameworks, spiced up with practices unique to the team. Agile methods are used in industries besides software.

AGILE ADVANTAGES

Agile methodology suits perfectly for these companies that are looking to transform the way in which projects are managed and the way they operate as a whole. Benefits of agile software development include:

1. **Focuses on Users.** Agile uses user stories with business-focused acceptance criteria to determine product features, each of them delivers value. Also, this provides the opportunity to beta test software after each Sprint and the ability to make changes as needed.

2. **Focuses on Business Value.** The team can deliver features, which provide the most business value based on understanding what is most important to the customer's business.

3. **Improves Quality.** The team focuses on high-quality development, testing, and collaboration by dividing the project into manageable unit producing frequent builds and conducting testing and reviews during each iteration, quality is improved by finding and fixing bugs quick identifying expectation mismatches in the early stages of development.

4. **Transparency.** A unique opportunity for customers to be involved throughout the project process and monitor that the evolution is adopted at any phase of the process: from prioritizing features to iteration planning; from review sessions to frequent software builds within new features. On the other side, this requires customers to understand that they are seeing a work in progress in exchange for this added benefit of transparency.

6. **Predictable Costs and Schedule.** The cost is limited regarding the amount of work team perform in fixed-schedule time because each Sprint is a fixed duration. Plus, the customer can more readily understand the approximate cost of each feature, in turn, it improves decision making about the priority of features and the importance of additional iterations.

7. **Allows for Change.** An opportunity to continuously improve and reprioritize the overall product backlog. Teams make changes in order to improve the effectiveness and efficiency of work. New or changed items of the backlog can be planned for the next iteration within the opportunity to introduce changes.

8. **Stakeholder Engagement.** Stakeholders and developers work closely every day. All stakeholders and team members must remain motivated in order to achieve optimal project results. As a matter of fact, for stakeholder and team engagement Agile provides multiple opportunities before, during and after each Sprint. That is why there exists a high degree of collaboration between the customer and the whole project team, which provides more opportunities for the team to fully understand the customer's vision.

AGILE DISADVANTAGES

Despite the benefits Agile can potentially offer, it's important to be aware of the disadvantages of Agile methodology. With that in mind, here we have five basic disadvantages of Agile:

1. **Limited Documentation.** Lack of attention to documentation can make it difficult for new team members to access needed information. In Agile, documentation happens throughout a project, and often "just in time" for building the output, not at the beginning. As a result, it becomes less detailed and often falls to the back burner.

2. **Poor Resource Planning.** Agile may not work as expected, for example, if a client is not clear about the goals, if the project manager or the team has no experience or if they do not "work well" under pressure. It is therefore recommended to do an adequate analysis in order to identify the best methodology to apply in every situation. The team needs to have a solid foundation and a comparable skill level. Because Agile is based on the idea that teams won't know what their end result (or even a few cycles of delivery down the line) will look like from day one, it's challenging to predict efforts like cost, time and resources required at the beginning of a project (and this challenge becomes more pronounced as projects get bigger and more complex).

3. **Fragmented Output.** There is a danger that the lack of project boundaries will lead to uncontrolled expansion, which can cause the project to never reach completion. Incremental delivery may help bring products to market faster, but it's also a big disadvantage of Agile methodology. That's because when teams work on each component in different cycles, the complete output often becomes very fragmented rather than one cohesive unit.

4. **Difficult Measurement.** Because the Agile methodology has less formal and more flexible processes, it may not always be easily included in larger and more traditional organizations. Since Agile delivers in increments, tracking progress requires you to look across cycles. That long-game makes measuring progress difficult.

5. **No Finite End.** Given that this methodology focuses mainly on the short term, the risk that the long-term vision will be lost does exist. It is difficult to accurately determine the amount of time and money that will be needed to complete the project due to constantly changing requirements. A high level of interaction between the client and the developers is required, which can take time and make the process difficult. The fact that Agile requires minimal planning at the beginning makes it easy to get sidetracked delivering new, unexpected functionality. Additionally, it means that projects have no finite end, as there is never a clear vision of what the "final product" looks like.

AGILE METHODOLOGY VS WATERFALL: WHICH IS RIGHT DEVELOPMENT METHODOLOGY FOR YOUR PROJECT?

Agile and Waterfall are two basic and most popular distinct methods of the modern software development industry. The waterfall is the first of them and it can also be called the traditional method of software development. Waterfall means a linear approach to development, which is based on strict planning and performing the plan step by step. Agile is the second one, this specific type of Rapid Application Development is newer than Waterfall (it originated in the 2000s) and it is typically implemented with Scrum or Kanban. The agile approach to software development has no strict structure.

You might be wondering which method is right for you and your project. Consequently, it is important to do some research and understand the advantages and limitations of each approach.

WATERFALL

AGILE METHODOLOGY		WATERFALL METHODOLOGY
Work Assignment	Self-organizing team	Project Manager
Responsibilities	Shared	Delineated
Status reports	Transparency, shared knowledge	By Project Manager
Project Scope	Changes can be made well in advance with time and within the budget. Agile works well even if the Scope is not defined in advance	Waterfall performs well when the Scope is well known in advance and contact terms limit changes
Team	Agile intends small or midsize dedicated teams with high coordination	Waterfall involves large teams. It decreases coordination among team members
Customers	Agile allows customers to be available throughout the project	Waterfall requires customers to be available only at milestones

DevCom

WATERFALL ADVANTAGES AND DISADVANTAGES

👍	👎
The ease of managing. Each Waterfall's phase has specific deliverables and a review process	Makes changes difficult, it is difficult to measure progress within stages
Faster projects delivery	Excludes the client and/or end user
The method works well for smaller size projects (with easily understandable requirements)	Delays testing until after completion
The well-documented processes and results	Not a good model for complex and object-oriented projects, poor model for long and ongoing projects
Rather beneficial to manage dependencies	Adjusting scope during the life cycle can end a project
Easily adaptable for shifting teams	

DevCom

⇒ When new changes need to be implemented. The freedom agile gives to change is very important: new changes can be implemented at very little cost because of the frequency of new increments that are produced.

⇒ To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.

⇒ Unlike the waterfall model in an agile model very limited planning is required to get started with the project. Agile assumes that the end-users' needs are ever-changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.

⇒ Both system developers and stakeholders alike find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

⇒ The agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time-critical applications. Extreme Programming (XP) is currently one of the most well known agile development life cycle models.

CONCLUSION

Agile methodologies are not best suited for all projects. When communication between the developer and the customer is tough, or when the development team does not have experienced developers, Agile Methodologies will not give the best results. These methodologies exhibit optimal results when there is a strong communication process between the developer and the customer, and the development team compromises skilled team members. When there is a chance for misunderstanding the accurate customer requirements, or when the deadlines and budgets are tight, then Agile methodologies are the most optimal approach for a business solution.

Keep in mind. **DevCom has adopted Agile software development methodologies** *designed to create lean, timely and effective solutions that achieve our client's goals.*

These methodologies represent an iterative development model in which the overall effort is broken into multiple releases in order to achieve the goals outlined for a particular phase. We put a lot of emphasis on three elements of the development lifecycle: Initial Project Assessment and Planning, Quality Project Management, Quality Assurance.

In case you have a project that requires Agile software development, **share it with us**.

Written by: Liza Hazevytch – Marketing Manager at DevCom, and Halyna Vilchynska – Head of Marketing at DevCom.

ALL INDUSTRIES   ALL SERVICES

Don't miss out our similar posts:

+1 (800) 780 8560
welcome@devcom.com

Services   Expertise   Projects   Industries   Company   CONTA

Hybrid vs Native App: Which One to Choose for Your Business?

The modern world has witnessed a significant reliance on portable technology Most people nowadays prefer to choose their mobile devices or tablets for their work processes Mobiles are also becoming more and more capable...

January 20, 2021

How to Design and Develop Successful SaaS Application

Not so long ago, all of a company's IT systems were on-premise Back then, a cloud was just something white and fluffy above us Cloud computing has since become the norm Today, cloud platforms can be utilized for just...

October 15, 2020

What Are NoSQL Databases and Why Enterprises Rely on NoSQL?

The business shift to digitization and customer experience has ushered enterprises into the digital revolution era At the heart of the revolution are cloud technologies, mobile apps, big data applications, social and...

September 7, 2020

Let's discuss your project idea

In case you don't know where to start your project, you can get in touch with our Business Consultant. We'll set up a quick call to discuss how to make your project work.

Contact us

USA: +1 800 780 8560
EU: +380 32 244 4077

welcome@devcom.com

Company

About DevCom
How We Work
Projects
Industries
Contact Us
Partnership

Services

Expertise
Custom Software Development
Mobile Application Development
Web Development
Software Support

Blog

Continuous Integration vs. Continuous Delivery vs. Continuous Deployment

News

DevCom Chess Tournaments

    Cookie Policy   Privacy Policy   Privacy Tools   Sitemap