



Agile Software Development A Smart Choice for Outsourced Projects

The traditional waterfall approach to software development is giving way to a more flexible, transparent methodology called Agile, which delivers faster ROI while maintaining high levels of software quality. Unlike predictive “build to blueprint” waterfall methods, Agile is adaptive, using time-boxed iterations and collaborative problem solving to quickly create release-ready software that end-users need. Agile makes it easy for product owners to align development with changing business priorities and manage risk more effectively.

Introduction

With every software project that fails, business leaders grow increasingly frustrated with traditional approaches to development.

Failures can include projects that are cancelled because of ballooning costs, scheduling overruns, or changing business priorities. Likewise, projects that are completed but produce software with too many bugs or the wrong features can also be considered failures.

Traditional software development employs a waterfall model, a one-way, sequential process with a strong emphasis on planning and little capacity for change. In waterfall development, the project team defines user requirements, develops a blueprint, builds a complete software application according to the blueprint, and then tests the software, fixing as many bugs as possible before the release date.

When a working version of the software finally sees the light of day, it might not resemble what was initially asked for or, even worse, what end-users currently need. In essence, after feeding user requirements into one side of the development black box, a product owner must hope and pray that useful software eventually emerges from the other side.

Clearly, a product owner could manage risk more effectively if the development process was transparent and responsive to real-time business priorities—no black box, no going over the Falls in a barrel.

This white paper describes Agile software development, a flexible, lower risk alternative to traditional waterfall methods that rapidly produces high quality, release-ready software.

The Trend toward Agility

In the modern global economy, companies are continually driven to work faster, smarter, and cheaper by offshore competitors. Because of these competitive pressures, business leaders are expected to make an airtight business case for each prospective software project, including a risk management strategy that inspires confidence. To meet and exceed these expectations, they must find ways to achieve:

- **Faster ROI**
Getting software to market faster means making money sooner.
- **Higher quality products**
Fewer bugs lead to happier customers, better reputation, and lower costs.
- **Alignment between business and IT**
With business and IT on the same page, companies produce great software that end-users need.

To reach these objectives, business leaders are moving their companies away from traditional waterfall software development toward iterative Agile methods. In a 2009 article, Sue McKinney, vice president of development, transformation, and integration at IBM, stated “At IBM alone, in the last two years, the number of Agile projects has increased from 5 to over 200.”⁴

We firmly believe, and our executives firmly believe, that the most successful organizations of tomorrow will be the ones [able] to adopt agile principles.

- Per Kroll
Chief Architect
Rational Expertise
Development & Innovation
IBM¹

...my definition [of agile] is that you accept input from reality, and you respond to it.

- Kent Beck
Author²

To say that companies or CIOs are reluctant to embrace agile is like saying they wouldn't take aspirin for a headache. And they're not only not taking the aspirin, they're banging their heads against the wall and wondering why it hurts.

- Jim Johnson
Chairman
Standish Group³



IDC research indicates that 70-80% of software development failures result from poor requirements gathering, analysis and management. Proactive, well-managed Agile approaches can help to solve those problems by enabling quick and frequent feedback across the cultural divide between business and IT...

- The Agile Impact Report Rally Software⁵

If you count the cost of bugs that make it too far downstream in the process, there is simply no way to financially justify any approach except some kind of agile approach.

- Kelly Anderson
Technical Team Lead
Sonic Innovations⁶

Waterfall has demonstrably failed. It has failed to deliver on its promise. Requirements always change on an ongoing basis. If you don't give users input early on in the process, you end up giving them something they didn't ask for.

- James Governor
Analyst, Redmonk⁷

About Waterfall Development

Waterfall development is a predictive engineering model that was borrowed from the manufacturing and construction industries, where it has traditionally been used to build things like cars, refrigerators, bridges, and office towers.

A waterfall development process is typically divided into successive stages, like this:

- **Requirements**
Analyze user requirements and formulate a comprehensive list.
- **Design**
Create a software blueprint that includes all of the required features.
- **Construction**
Write code for the required features, as per the blueprint.
- **Integration**
Assemble individual pieces of code into the complete software application.
- **Validation**
Test the software and fix bugs.
- **Maintenance**
Improve functionality in subsequent releases.

Using a waterfall approach, the development team manages risk through detailed analysis, exhaustive planning, formal processes, and conservative scheduling. The product owner is excluded from critical decisions that directly impact the final product and thus remains in the dark when it comes to risk.

The waterfall model assumes that working from a detailed blueprint is better than making design choices on the fly as a product is being built. “Build to blueprint” is most appropriate for well-defined, fixed scope projects—bridges, for example—intended for users with simple requirements and undertaken by organizations with deep pockets and long timelines.

5 Problems with Waterfall Software Development

There are 5 major problems with using a waterfall approach to software development:

1. **Rigidity**
It lacks the flexibility to produce software that is relevant to an ever-changing market. Design changes are effected through a formal change request process that can be expensive and time consuming, especially if a significant change is made late in development. Such a change can force developers to scrap significant amounts of code and start from scratch, which negatively impacts the development schedule.
2. **Lack of user-centric focus**
Performance metrics are too abstract and process-focused, which makes it easy for developers to fall into the “software first” trap.



Agile methods derive much of their agility by relying on the tacit knowledge embodied in the team, rather than writing the knowledge down in plans.

- Dr. Barry Boehm
Software Engineer⁸

Describing a feature in terms of what the customer is trying to achieve enables the development team to deliver a solution that focuses on value.

- Dave West
Senior Analyst
Forrester Research⁹

...as compared to industry averages, the development teams utilizing Agile practices were on average: 37% faster delivering their software to market...16% more productive...Able to maintain normal defect counts despite significant schedule compression...

- The Agile Impact Report
Rally Software¹⁰

3. **Inefficiency**

The sequential process forces some team members to wait for output from others; the final software often includes obsolete features which are ultimately deleted; and the overall process is not aligned with how developers actually work, so it's like trying to fit a square peg into a round hole.

4. **Slowness to market**

It's an all-or-nothing approach, which means there is no product until the last step of a long process, at which time *all* the planned features—some useful, some obsolete—and *all* the bugs are delivered. Near the end of a project, as the release date looms large (or has come and gone), software is often rushed to market, bugs and all.

5. **Higher risk**

Problems 1–4, coupled with the all-or-nothing, sequential nature of waterfall development, make it a gamble for product owners who are competing in fast-moving markets.

Agile Development

Agile is an adaptive approach to software development that assumes requirements will inevitably change, and therefore makes it easy to incorporate changes into the product at any time. Using Agile's incremental, iterative process, developers build high value features first, ensuring that a potentially releasable product is always close at hand.

Since Agile development assumes a changing scope, the product owner understands the risk involved, namely that a project will require an unknown number of iterations, or sprints, to meet its stated objectives. The product owner is responsible for defining scope changes and therefore sits in the driver's seat when it comes to risk management.

To encourage outstanding results, Agile methods foster daily communication between stakeholders and the project team, collaborative problem solving, individual discretion, and adherence to best practices.

Benefits

The key benefits of an Agile approach to software development include:

- **Faster ROI**

The most important features are developed first, resulting in release-ready software.

- **Responsiveness to changing business needs**

Sprint objectives are defined by current business priorities; this flexibility is a competitive advantage.

- **Transparency**

The product owner has full visibility on the development process through daily updates and online tools for monitoring project status.



Agile comes easier, and results in greater success, in organizations where building software is their business...

- David Rubinstein
Editor-in-chief
SD Times¹¹

Business people and developers must work together daily throughout the project.

- Principles behind the Agile Manifesto¹²

- **High quality software**
In each sprint, developers fix bugs from the previous sprint, allowing them to quickly zoom in on code problems and rectify them. Using Agile, developers are not overwhelmed by a “bugalanche” near the end of the development process.
- **User-centric focus**
Success is measured by user-accepted software.
- **Higher efficiency**
Agile minimizes the amount of wasted effort by incrementally adding and refining valuable features instead of adding and subsequently deleting extraneous features. Agile also empowers developers to achieve objectives in their own way (using best practices, of course).

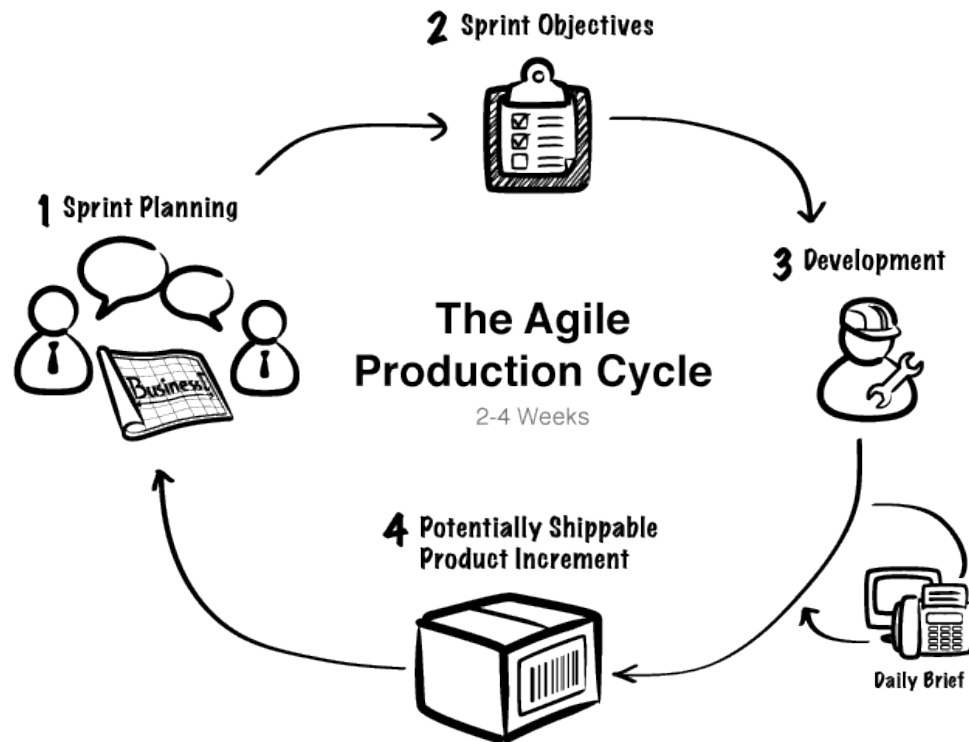


Figure: Agile Development Process

How It Works

Typical steps in an Agile development process are:

1. Using input from end-users, customers, team members, and other stakeholders, the project team and product owner create/revise a prioritized list of “user stories” for the software project. A user story describes a complete, purposeful user interaction with the software. For example, a user story might explain the need to create an account for gaining system access. Several product features may be required to satisfy a single user story.

That's what agility might be said to be about: encountering all the problems so early and so often that the effort to fix them is less than the pain of enduring them.

- Ron Jeffries
Author¹³



We have to produce a product that people want to use. They can describe it 18 months ahead, but what they want could change half way through and that's where problems happen. The way we used to handle software development and project management was unacceptable.

- Dick Faris
CTO
Primavera Systems¹⁴

It is the product owner who ultimately decides which user stories to include in the Product Backlog (the list of user stories) and how they are prioritized. For each user story, the product owner writes a “Definition of Done”, which explicitly states criteria that must be met for a coded story to be considered complete and working. By creating a clear Definition of Done for each user story, the product owner purges ambiguity and false assumptions from the evaluation process. Helpful tools like burn down charts are available for managing user stories and working through the Product Backlog.

2. The team selects high-priority user stories to be developed in a sprint.
3. During the sprint, developers create code for the selected user stories. Each sprint has a fixed duration, typically 2–4 weeks. The team and product owner meet daily to discuss the project and receive feedback from customers and other stakeholders.
4. After a sprint, the product owner evaluates the software produced. How well does it meet end-users' needs? The team and product owner hold a retrospective meeting where they assess the sprint and identify opportunities for improvement in the next sprint. Thus, the team achieves a measured, steady improvement in efficiency over time.
5. The team repeats steps 1–4 until the project's objectives are met.

Naturally, Agile works best in conjunction with proven software engineering principles such as test driven development, automated builds, automated regression testing, and architectural best practices.

Success Stories

Companies that have benefited from switching to Agile software development include:

- **Salesforce.com**, who went from one software release per year using traditional waterfall methods to one potentially releasable product every 30 days using Agile.¹⁵
- **Yahoo.com**, who surveyed employees after the migration to Agile, asking them to rate Agile against other methods they'd been using. A large majority of respondents indicated an increase in 30-day productivity, clarity of goals, collaboration and cooperation, efficiency (less time wasted), and business value of their product. Support for Agile remained consistently high—greater than 80%—in three surveys conducted in 2005, 2006, and 2007.¹⁶
- **British Airways**, who previously realized ROI on software projects in 9–12 months, now sees incremental ROI in as little as 3 months since moving to Agile.¹⁸
- **BMC Software**, who now brings products to market almost three times faster than its competitors, with productivity about 50% higher than the industry norm.¹⁹

In 2007, CNN Money ranked Agile Software Development eighteenth in “The 50 Who Matter Now”.¹⁷



Key Features: Agile versus Waterfall

Agile	Waterfall
Iterative	Sequential
Time-boxed sprints	Open-ended (until release date)
Assumes changing scope	Assumes fixed scope
Incremental	All-or-nothing
Risk managed by product owner	Risk hidden from product owner
Change-friendly	Change-averse
Daily communication	Communication variable
Collaborative problem solving	Building to blueprint
Empowers team members	Constrains team members
Adheres to best practices	Adheres to best practices
Transparent to product owner	Isolated from product owner
Bugs fixed throughout development	Bugs fixed after development
Release-ready software available throughout development	Release-ready software available after development
Software success is defined by user-acceptance	Software success is defined by the quality of various artefacts other than the software itself

References

1. Krill, Paul. "IBM Promotes Agile Development", www.infoworld.com, March 4, 2008.
2. Krill, Paul. "Extreme Programming Inventor Talks About Agile Development", www.computerworld.com, November 12, 2007.
3. Wailgum, Thomas. "How Agile Development Can Lead to Better Results and Technology-Business Alignment", mobile.cio.com, May 21, 2007.
4. McKinney, Sue. "Evaluating Agile Software Development Programs", www.cio.com, January 23, 2009.
5. "The Agile Impact Report", Rally Software Development Corp, Boulder, Colorado, 2009.
6. Schindler, Esther. "Getting Clueful: 7 Things CIOs Should Know About Agile Development", www.cio.com, February 6, 2008.
7. Kanaracus, Chris. "'One Size Fits All' Will No Longer Work", www.computerworld.com, November 20, 2007.
8. Boehm, Barry. "Get Ready for Agile Methods, with Care", *Computer*, vol. 35, no. 1, pp. 64-69, Jan. 2002, doi:10.1109/2.976920.

The change we made was going from a traditional waterfall method to an incremental development model. Before...we were scrambling so hard to get all the committed features in by the feature complete date...that the program was always very buggy at that point. We'd be desperately finding and fixing bugs, with little time to revise features based on tester feedback.

- Russell Williams
Co-architect of CS3
Adobe Systems²⁰



9. West, Dave. "Case Study: Microsoft Speeds Tool Delivery With Agile Development", Forrester Research Inc., March 27, 2009.
10. "The Agile Impact Report", Rally Software Development Corp.
11. Rubinstein, David. "More Agile Success Seen at ISVs than Corporate IT", www.sdtimes.com, June 15, 2009.
12. "Principles behind the Agile Manifesto", <http://agilemanifesto.org/principles.html>.
13. Jeffries, Ron. <http://tech.groups.yahoo.com/group/agile-testing/message/3878>, Message #3878 of 18356, March 25, 2004.
14. Schwaber, Ken, and Robert C. Martin. "Primavera Gets Agile", Object Mentor Inc. (Gurnee, Illinois) and Advanced Development Methods Inc., July, 2004.
15. Fry, Chris, and Steve Greene. "Large Scale Agile Transformation in an On-Demand World", Salesforce.com, 2007.
16. Benefield, Gabrielle. "Rolling out Agile in a Large Enterprise", Proceedings of the 41st Hawaii International Conference on System Sciences, 2008.
17. "The 50 Who Matter Now", [money.cnn.com: http://money.cnn.com/galleries/2007/biz2/0706/gallery.50whomatter.biz2/jump.html](http://money.cnn.com/galleries/2007/biz2/0706/gallery.50whomatter.biz2/jump.html).
18. Guglielmo, Karen. "Agile Development Methodology Not Easy but Worth the Effort, Users Say", SearchCIO.com, October 28, 2009.
19. Desmond, John P. "Agile Development Techniques Produce Record Results at BMC Software", www.softwaremag.com, March, 2009.
20. Branscombe, Mary. "Adobe Edits the Development Cycle: Less Bugs, More Weekends Off", www.theregister.co.uk, March 8, 2007.

Architech Solutions is a Toronto-based technology consulting and software development firm. We design and build powerful, user-centred systems that work. We're agile, disciplined, and passionate about delivering for our clients.



Architech Solutions
3 Church Street
Suite 602
Toronto, Ontario
M5E 1M2

Phone: (416) 607-5618
Fax: (416) 352-1768

To book a free on-site Discovery Workshop led by our team of consultants, e-mail info@architech.ca or visit us at www.architech.ca

