



AGILE SOFTWARE DEVELOPMENT AND ATLASSIAN PRODUCTS

@spkandassociates



Introduction

Software development teams increasingly face mounting pressure to deliver high-quality products efficiently while adapting to evolving customer needs. In response to these challenges, agile methodologies like Scrum and Kanban have gained immense popularity. They empower teams to be flexible, collaborative, and responsive throughout the software development lifecycle. [Atlassian](#), a leading provider of software development and collaboration tools, offers a comprehensive suite of products specifically designed to bolster agile software development practices. This white paper delves into the core principles of agile software development and elucidates how Atlassian products, including [Jira](#), [Confluence](#), and [Bitbucket](#), can significantly enhance team productivity, streamline collaboration, and ensure successful project delivery in a software development environment.

We will now explore some fundamental terms and principles of Agile. If you are an experienced software developer or a seasoned scrum master working in a high-performing Agile team, feel free to skip to the rest of this section. However, if you are new to the Agile development journey, this section is tailored to your needs. While reading this section alone may not provide you with all the necessary insights, it will lay a solid foundation for your success in Agile software development. It's important to note that while this white paper emphasizes the Atlassian tools that support Agile development, it's important to recognize that tools alone cannot fully support development teams in accomplishing their goal and a holistic approach to tools, teams and methodologies must be implemented.

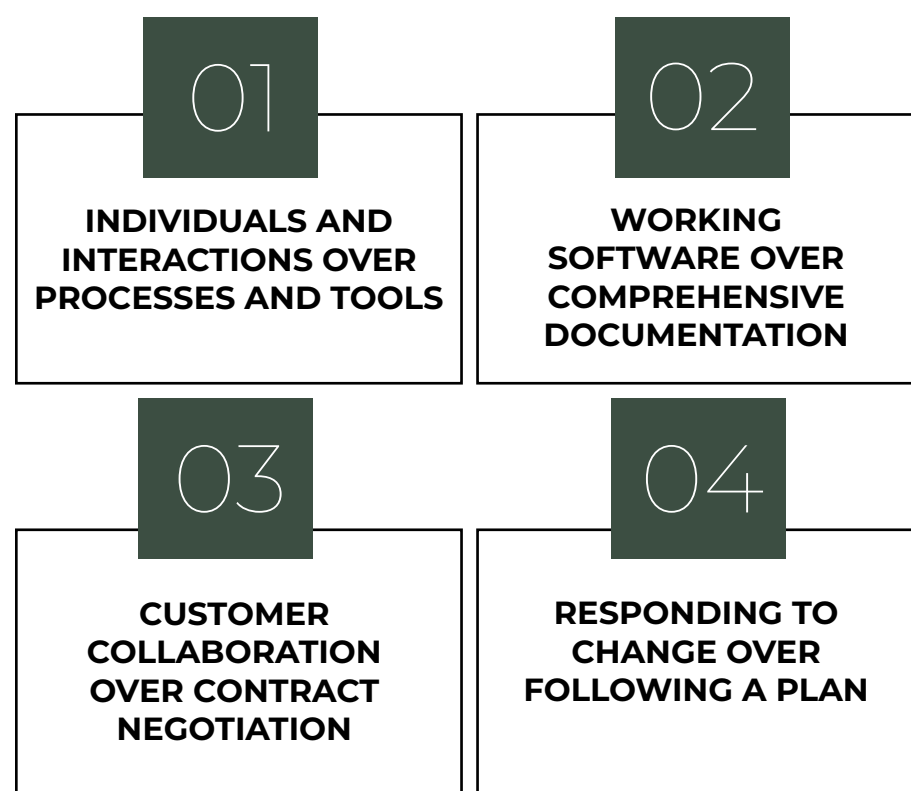
WHAT IS AGILE?

Agile software development represents an iterative and collaborative approach to software building, placing a premium on adaptability rather than rigid planning. The agile methodology centers around the frequent delivery of functional software, breaking projects into smaller, more manageable tasks, and continuously iterating and refining those tasks.

The concept of “agile” was coined in the 2001 Agile Manifesto, which advocates for valuing individuals and interactions, working software, customer collaboration, and responsiveness to change, over extensive documentation, adherence to plans, tools, and processes. Popular agile frameworks encompass Scrum, Extreme Programming (XP), and Kanban.

WHAT IS THE AGILE MANIFESTO?

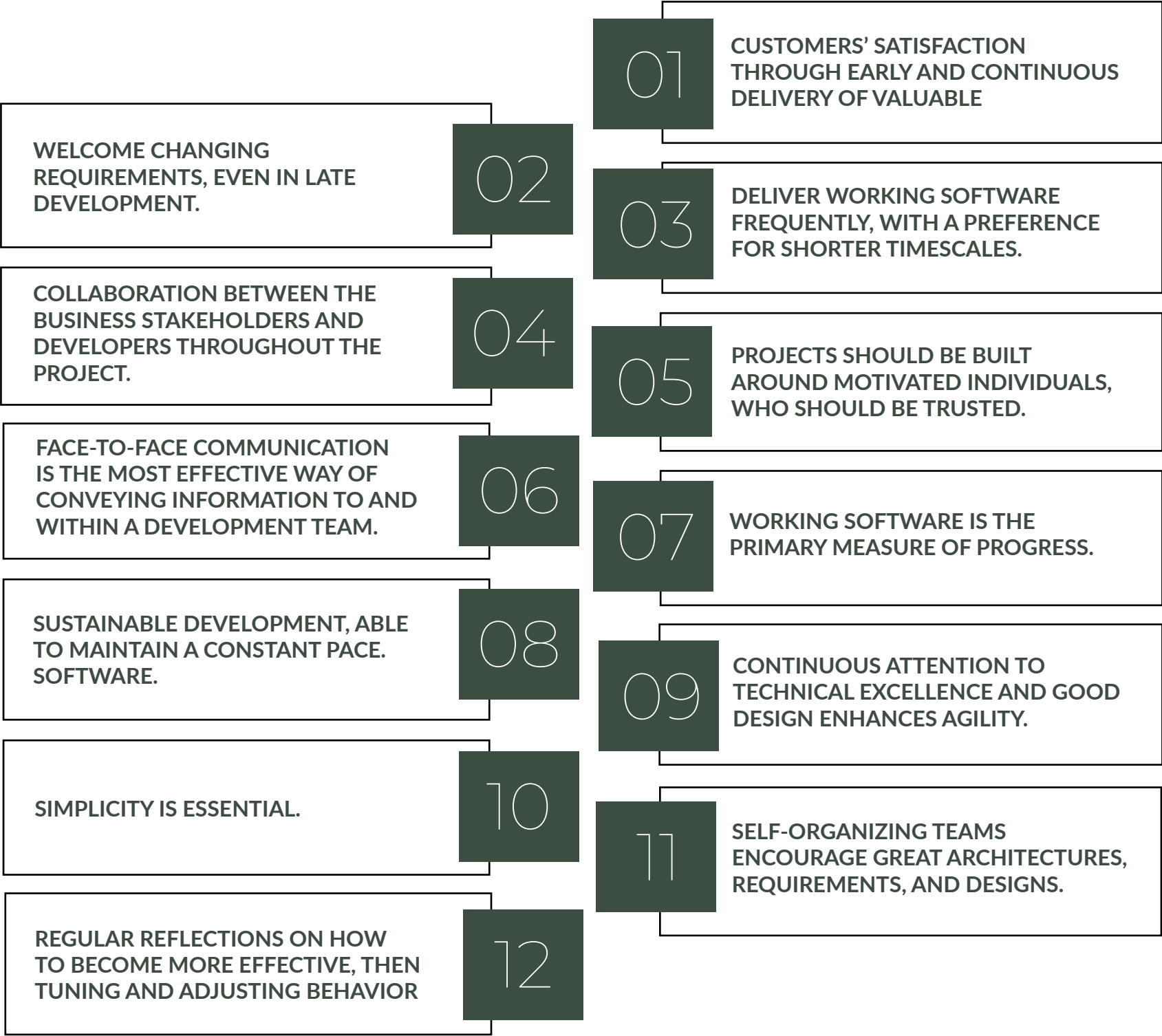
The [Agile Manifesto](#) is a collection of guiding values and principles for Agile software development. It was created in 2001 by a group of software developers who met in Utah for the purpose of discussing lightweight development methods. The Agile manifesto consists of four value statements:

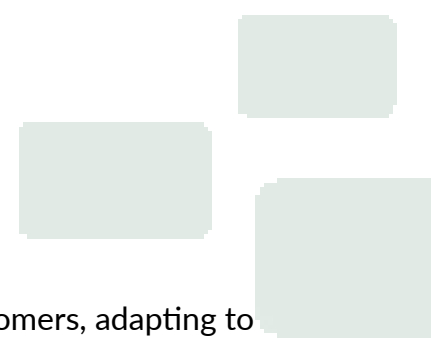


These values prioritize human interaction and collaboration over rigid processes and documentation. In 2001, this was a large shift from the traditional “big requirements document” approach and the 1-3 year development cycle for software. The Agile Manifesto emphasizes delivering working software iteratively and incrementally, encourages involvement with customers throughout the development process, and embraces change as a natural part of the development cycle. Thus the term “agile”. The principles behind the values are articulated further in the form of twelve principles, which support teams to deliver value frequently, work collaboratively, and respond to change more easily.

WHAT ARE THE TWELVE PRINCIPLES OF AGILE SOFTWARE?

The Twelve Principles of Agile Software are key guidelines for an Agile approach to software development, which spans across many Agile methods (i.e. Scrum, Kanban, XP, etc). The Twelve Principles of Agile Software are:






These principles promote a framework or mindset (not a process) that prioritizes delivering value to customers, adapting to change, and collaboration among team members throughout the software development lifecycle all while being sustainable.

DEVOPS VS. AGILE

Before we delve into certain areas, such as DevOps and its related terms like “continuous integration,” “continuous delivery,” or “continuous deployment,” let’s establish the foundation that Agile and DevOps are two complementary methodologies that address different aspects of the software development lifecycle. They are not mutually exclusive; rather, they work in tandem. As Agile teams become more efficient, it becomes crucial for IT Operations teams (or release teams) to match that speed. Agile primarily focuses on iterative and incremental software development, emphasizing responsiveness to changing requirements and fostering collaboration among team members. In contrast, DevOps centers around integrating development and operations teams, placing a strong emphasis on automation, continuous integration and delivery, and cultivating a culture of collaboration and shared responsibility.

When it comes to goals, Agile aims to enhance software development processes by delivering value to customers through frequent iterations, feedback, and adaptation. It emphasizes breaking projects down into smaller, manageable chunks that can be incrementally addressed.

On the other hand, DevOps aims to dissolve the traditional barriers between development and operations teams, fostering collaboration and shared responsibility. It enables continuous delivery and strives to improve the stability, resilience, and scalability of software systems.



DEVOPS IS AN EXTENSION OF THE AGILE METHODOLOGIES. GAINING VALUE QUICKLY IS IMPERATIVE TO BOTH OF THOSE METHODS AND THEY ENABLE COMPANIES TO GAIN LEARNING FROM CUSTOMER’S USAGE OF THEIR PRODUCT. THAT PRODUCT DISCOVERY FUELS THE PRODUCT BACKLOG AND THE CYCLE REPEATS. THE MORE QUICKLY YOU CAN DO THAT, THE MORE QUICKLY YOU WIN.

SPK and Associates, Vice President
CARLOS ALMEIDA

Ultimately, both methodologies share the common goals of enhancing software development processes, delivering increased customer value, and enabling teams to effectively respond to evolving requirements with high-quality and reliable software deliverables. DevOps serves as an extension of Agile methods, helping to achieve a rapid yet high-quality software development cycle.

Jira's Alignment To The Agile Manifesto

Atlassian's flagship product, Jira Software, is a popular project management tool that aligns with the Agile manifesto by providing support for Agile software development practices. Jira offers features and functionalities that promote the values and principles of Agile. Now, let's explore how these Jira capabilities align with the Agile Manifesto.

WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION WITH JIRA

The value of "working software over comprehensive documentation" is highly appreciated by both software developers and customers. When software functions as expected, the need for extensive documentation diminishes, or in some cases, becomes unnecessary. In line with this Agile value, Jira Software supports teams by providing a solid platform for planning, tracking, and delivering working software increments. It offers a wide array of Agile project management methodologies, including [Scrum](#) and [Kanban](#), allowing teams to organize their work in iterative and incremental cycles. By leveraging Jira, teams can deliver valuable software frequently and adapt to changing requirements efficiently. Furthermore, Jira's customizable project templates for Scrum, Kanban, and other methodologies empower teams to utilize Jira in a way that aligns with their preferred working style, rather than being constrained by rigid processes dictated by the tool itself.

COLLABORATE WITH JIRA

Jira Software effectively promotes the Agile value of "customer collaboration over contract negotiation" by offering features that enhance communication and collaboration among team members and stakeholders. Additionally, Jira serves as a valuable tool in making work visible to all involved parties. Through its intuitive dashboards, project kanban boards, and customizable filters, individuals can easily access and focus on the specific work that is relevant to them. This transparency enables stakeholders and teams to have a clear view of the progress and status of ongoing projects.

Within Jira, users can create and manage user stories, prioritize and assign tasks, and provide valuable feedback on the software being developed. These functionalities facilitate ongoing collaboration between development teams and customers, ensuring that the software aligns closely with their needs and expectations. By providing a collaborative platform, Jira enables effective communication, reduces misalignment, and helps foster a strong partnership between software developers and customers.



JIRA SUPPORTS FLEXIBILITY

Jira Software effectively supports the Agile principle of "responding to change over following a plan" by providing teams with the necessary flexibility and adaptability to accommodate evolving requirements. Teams can easily modify their backlog, reprioritize tasks, and adjust their plans as needed. For instance, let's consider a scenario where teams have pulled certain user stories or tasks into their next sprint. While working on these user stories, a customer identifies a new, high-priority bug or enhancement. Jira allows the team to promptly receive an alert about this task, enabling them to add it to the top of the product backlog for inclusion in the next sprint.

Furthermore, Jira's customizable workflows and boards play a pivotal role in empowering teams to respond swiftly to changing priorities and embrace emerging requirements. These features allow teams to configure their workflows and boards to suit their specific needs and adjust them as circumstances evolve. Whether it involves reprioritizing tasks, modifying sprint plans, or incorporating new requirements, Jira provides the necessary tools and flexibility to adapt and respond effectively to change.

Atlassian also has a tutorial to learn Scrum with Jira Software. Check it out [here](#).

Bitbucket Is The Agile Software Development In Action

Now, let's look at Bitbucket. Firstly, it is important to grasp the concept of Git and software development repositories. Git serves as the foundation for modern software development practices and forms the underlying technology for Bitbucket. However, while Git is a powerful version control system, it alone may not provide all the features that developers require to effectively adapt to the dynamic requirements imposed by customers.

While Git lays the foundation for version control, Bitbucket enhances it by equipping developers with the necessary features to navigate the ever-changing landscape of customer requirements.

WHAT IS GIT?

Git serves as a distributed [version control system \(VCS\)](#) that excels in managing projects of various sizes with speed and efficiency. It is an open-source tool widely used in software development for source code management, change tracking, and fostering collaboration among development teams. Additionally, numerous tools, including Bitbucket, are built on top of Git, further enhancing its capabilities.



At its core, Git empowers developers to create multiple branches within a codebase, enabling simultaneous work on different features or bug fixes. These branches can be merged back into a main branch, facilitating parallel development and mitigating conflicts that arise when multiple individuals work on the same codebase.

As a distributed VCS, Git ensures that each developer possesses a complete copy of the code repository on their local machine. This feature enables offline work, quick branching, and effortless collaboration.

Git's standout feature lies in its ability to track changes made to files over time. Developers can easily review the complete history of changes, revert to previous versions, and compare differences between versions. This functionality is especially valuable for teams regularly collaborating on the same codebase.

When developers need to add or modify code, Git facilitates the creation of branches to isolate their work. Once completed (typically on a daily or weekly basis), developers merge their changes back into the main branch. This approach encourages efficient collaboration, minimizes conflicts, and aligns well with the DevOps principle of [continuous integration](#).



WHAT VALUE DOES BITBUCKET CLOUD ADD TO GIT?

While Git is a powerful distributed version control system, it requires hosting on infrastructure or cloud platforms such as Azure, AWS, or GCP. Bitbucket Cloud adds significant value by providing a centralized platform for managing code repositories and collaboration. As a web-based hosting platform for Git repositories, Bitbucket Cloud extends the functionality of Git by offering additional features like issue tracking, pull requests, code reviews, and built-in continuous integration (CI) capabilities. Additionally, Bitbucket Cloud includes project management functionalities that can be seamlessly integrated with Jira.

Unlike Git, which primarily focuses on version control, Bitbucket Cloud incorporates deployment and CI/CD capabilities through its Pipelines feature. Bitbucket Pipelines allows developers to define pipelines using configuration files, run automated tests, and deploy applications directly from the Bitbucket platform. This built-in CI/CD functionality simplifies the process of integrating automated build and deployment processes within the development workflow.

Ultimately, Bitbucket Cloud enhances Git by providing a centralized platform for code management, collaboration, issue tracking, and pull requests. It also includes built-in CI/CD capabilities through Bitbucket Pipelines, streamlining the development and deployment process. This integration of functionalities within Bitbucket Cloud adds significant value to Git users.

USING BITBUCKET AND JIRA

As a software developer, you can use Bitbucket in conjunction with Jira to resolve a ticket/story about a bug in the software or to add a new feature. Here's a step-by-step guide on how you can use Bitbucket to address the bug:

1 Link the Jira ticket with a Bitbucket repository

Start by linking the Jira ticket to the corresponding Bitbucket repository. This allows you to easily track the changes made for resolving the bug and associate them with the ticket. In Jira, navigate to the ticket and find the "Development" panel. Click on "Link repository" and select the appropriate Bitbucket repository.

2 Create a branch

In Bitbucket, create a new branch specifically for fixing the bug. The branch name could be related to the bug or the ticket number. Switch to this branch in your local development environment. It is also recommended to think of a branch as a specific Jira ticket. That way, you're working on a specific branch for a very specific ticket.

3 Write code and make changes

Make the necessary code changes to resolve the bug or add the feature in your local development environment. Use your preferred development tools and IDEs as you would normally.

4 Commit and push changes

Once you have made the required changes, commit your changes to the local branch. Provide a clear and concise commit message that describes the bug fix or adding the feature. After committing, push the changes to the remote branch on Bitbucket.

5 Create a pull request

In Bitbucket, create a pull request to merge your bug fix branch into the main branch. Provide a detailed description of the changes, referencing the Jira ticket, and include any relevant information or context.

6 Review and address feedback

Collaborators or team members can review your code changes and provide feedback or suggestions on the pull request. This Agile concept from XP (eXtreme Programming) will help increase quality by answering any feedback or comments raised during the review process.

7 Approve and merge

Once the changes have been reviewed and approved, merge the pull request into the main branch. This integrates your bug fix into the main codebase.

8 Update the Jira ticket

In Jira, update the status of the ticket to reflect the progress. Provide any relevant details, such as the branch name, pull request, and merge status. You can also create automation rules that will update the Jira ticket based upon the information you provide in Bitbucket.

Delivering Value With Bamboo

Agile Principle three emphasizes the importance of delivering working software frequently, with a preference for shorter timescales. To achieve this, companies should adopt a DevOps model and utilize tools that enable developers and IT operations staff to develop, test, and deploy code regularly. Deployments on a quarterly, semi-annual, or yearly basis do not align with the Agile principle of delivering value quickly. In such cases, it is crucial to explore Continuous Integration/Continuous Deployment (CI/CD) processes and tools to support faster and more frequent deployments.

One such tool that supports the quick delivery of software value is [Bamboo](#).



WHAT IS BAMBOO?

Bamboo facilitates continuous integration by automatically building, testing, and merging code changes as they are committed to a repository. This approach helps identify and address integration issues early, ensuring a stable codebase and a smooth development process. In the past, it was common practice to write code for a software project over several months and then spend additional weeks or months integrating the code through build and testing processes. However, such lengthy integration cycles increased the chances of errors going unnoticed for extended periods, which led to additional challenges, especially if there was developer turnover. Implementing continuous integration on a daily or weekly basis allows for quicker error detection and resolution, enhancing sustainability and minimizing the complexities of fixing code written months or projects ago.

Furthermore, Bamboo offers a range of automation features that streamline build processes. By automatically triggering builds based on code changes, Bamboo minimizes manual effort and accelerates the software delivery process. This automation aligns with Agile and DevOps principles, particularly the emphasis on continuous attention to technical excellence and good design enhancing agility.

Given the importance of testing and delivering a quality product, Bamboo integrates with various testing frameworks and tools. This integration allows teams to automate different types of tests, including unit tests, integration tests, and acceptance tests. By automating the testing process, Bamboo enables rapid issue identification and resolution, ensuring high-quality software releases. This aligns with Agile principles, including the core principle of delivering working software.

Lastly, Bamboo supports the creation of customizable deployment pipelines that automate the release process. Through its intuitive dashboard and reporting features, Bamboo provides visibility into the entire software delivery pipeline. This enables quicker feedback cycles, improved code quality, and faster time-to-market for software products, aligning with Agile principles.

JUST ENOUGH DOCUMENTATION WITH CONFLUENCE

Confluence, Atlassian's wiki product, serves as a centralized knowledge base for teams, enabling the creation, storage, and sharing of information. It aligns with Agile principles by emphasizing documentation that is "just enough" and fostering collaboration and transparency among team members.



Teams can utilize Confluence to store project documentation, requirements, user stories, meeting notes, and other relevant information. Additionally, Confluence offers built-in templates or customizable pages for creating and managing user stories, requirements, and product backlogs. This helps teams capture and organize user needs and effectively prioritize work, a crucial aspect of Agile development.

An essential aspect of Confluence that aligns with Agile principles is its ability to enhance collaboration and communication. The platform provides collaborative features such as comments, likes, and mentions, facilitating real-time discussions, feedback provision, and collaboration on project-related content. This promotes the Agile principles of face-to-face communication and continuous collaboration among team members.

While not explicitly stated in the principles, the ability to create product roadmaps is an essential part of Agile software development. Confluence offers features to support the creation of roadmaps, enabling stakeholders to collaboratively generate ideas through brainstorming sessions using Confluence's whiteboard feature. Additionally, finalized roadmap plans can be stored for review and reflection by business stakeholders. This supports the Agile principle of collaboration between business stakeholders and developers throughout the project, as regular update meetings can be recorded and shared with all stakeholders.

Essentially, Confluence provides a flexible and collaborative environment that aligns with Agile principles. It enables teams to document, organize, and communicate project information efficiently, supporting iterative development, transparency, and collaboration within Agile software development processes.



BONUS: Can Non-Software Teams Be Agile Too?

Absolutely! Non-software teams can indeed adopt Agile principles and benefit from tools like Trello or Jira Work Management to enhance their work management and collaboration.

For marketing teams, the growing complexity of their work, especially with the use of various digital tools, can make managing tasks and visibility challenging. By utilizing project management tools like Trello or Jira Work Management, marketing teams can efficiently plan and manage marketing campaigns, streamline content creation processes, handle event management, and effectively manage marketing requests from other departments. These tools provide visibility, collaboration, and workflow management capabilities that support Agile principles in marketing team operations.

Similarly, human resources teams can leverage Jira Work Management or Trello to streamline their processes. These tools can be utilized for tasks such as employee onboarding, leave management, performance reviews, and managing employee suggestions and feedback. By embracing Agile principles and utilizing these tools, HR teams can improve visibility, collaboration, and efficiency in their work.

The key point is that Agile principles can be applied beyond software development and tailored to the specific needs of non-software teams. Tools like Trello or Jira Work Management offer the necessary features and functionalities to support Agile practices, enabling non-software teams to enhance visibility, collaboration, and efficiency in their work.

Conclusion

Agile software development, with its focus on collaboration, adaptability, and customer-centricity, has transformed software development. Implementing Agile methodologies goes beyond just following a set of practices; it requires a mindset of collaboration, trust, and a commitment to better business outcomes. Atlassian's suite of products plays a vital role in supporting Agile principles and empowering teams to embrace Agile methodologies and enhance their development processes.

The integration of Atlassian's products creates a solid ecosystem that fosters transparency, communication, and visibility across development teams, stakeholders, and customers. Jira Software facilitates agile project management, Confluence enhances documentation and collaboration, and Bamboo provides continuous integration and delivery functionalities. Together, they establish a comprehensive framework that supports Agile software development from start to finish.

We've highlighted the numerous benefits of using Atlassian's products in Agile software development. Moreover, these tools offer high customization and scalability, accommodating the diverse needs of teams and organizations of all sizes. With extensive integrations, APIs, and a thriving [marketplace](#), teams can further enhance their toolset by integrating additional tools and extending functionality as required.

In summary, Agile software development, combined with Atlassian's suite of products, empowers teams to embrace agility, collaboration, and efficiency in their software development lifecycle. By adopting Atlassian's tools, organizations can successfully implement Agile methodologies, improve team productivity, and deliver high-quality software at a faster pace, effectively meeting the dynamic demands of the software industry. For more detailed information on Agile software development, you can [refer to Atlassian's articles on the subject](#).



To explore how SPKAA can support your Agile software development journey and maximize the benefits of Atlassian's suite of products, visit www.spkaa.com. Our team of experts is ready to provide tailored guidance, training, and consulting services to help you implement Agile methodologies, optimize your software development processes, and leverage the full potential of Atlassian tools. Take the next step towards Agile excellence and [contact us](#) today.



SPKAA.COM



(888)-310-4540



INFO@SPKAA.COM



**5011 SCOTTS VALLEY
DRIVE
SCOTTS VALLEY, CA
95066**

SPK and Associates is focused on improving Engineering with smart information technology solutions. SPK understands the systems, processes, data and applications critical to successful product development, and dedicate ourselves to helping clients build, test, and release products faster and better. For 25 years, we have helped our customers harness technology to optimize engineering and accelerate product delivery.