



# CLOUDY WITH A CHANCE OF AGILE.

Six reasons to use Contour with Agile development.

For projects big and small, over the last 10 years there has been a widespread, grassroots movement with development teams toward Agile processes to manage risk, collaborate with customers and embrace change during development. And, it's easy to see why.

The fundamental principles of Agile development are good. Agile processes welcome changing requirements even late in development. Agile processes value greater collaboration between customers and development teams. Agile processes empower developers and motivate them to do their best work. Who doesn't want these things?

The challenges companies face aren't in the ideals of Agile. The challenges exist in trying to apply them in practice within enterprise organizations where the project teams are distributed, the projects are complex and high risk, or require detailed documentation for compliance.



For some companies a simple task board of cards on the wall and a daily stand-up meeting is all the management and communication tools needed. But for many others, especially for larger organizations, they struggle to fully embrace Agile. How do we connect the business side of our organization with the Agile development team? What happens to requirements? How do we communicate what's going on? How do we ensure our legal/contractual/compliance needs are met and properly tracked? Executives, business analysts, project managers, product owners and others involved in the planning of product releases often ask these questions.

Our first answer might surprise some: It's possible you don't need to add Contour or anything else to what you're already doing with Agile. For some teams or some projects, an Agile development process is all you need for success. If you're in that situation, that's great. Pat yourself on the back and skip the rest of this article.

However, before you do, you might want to run through the following checklist of questions to ensure your team doesn't suffer from the common issues organizations face when plugging in an Agile development practice into your larger project and product management process. Also, for those of you who may be thinking of moving to an Agile process down the road, these are good questions to help anticipate some possible issues you might encounter when you implement Agile.

## **REALITY CHECK: The Agile Development Self-Assessment Checklist.**

### **Where Do You Have Challenges?**

- Are you having difficulty getting agreement and buy-in on what should be built for a project or release?
- Are Sprints not getting off to smooth starts as a result of not having enough definition around what to build?
- Are there disconnects between key stakeholder expectations and what gets delivered, even with working code continuously available?
- Are you discovering requirements issues too late in the cycle?
- Do key stakeholders have difficulty organizing and planning around delivery because they don't know exactly what they are getting and when they are going to get it?
- Are there critical business objectives or non-functional goals that must be demonstrated to get certification or final customer sign-off?

If you have one or more “yes” answers, that indicates you may have needs that require more than an Agile development process and tools alone can support. Bringing a collaborative requirements management solution like Contour into the mix can help provide a complete solution that connects your entire project team to what you’re building from inception to delivery.

## Six reasons to use **Contour** with Agile.

### **ONE:** If you have trouble agreeing on what to build next.

Whether you work from existing stories parked in a product backlog or you create new stories for the next release or iteration, often the stories are bite-sized chunks that fail to provide full context for a holistic solution. A sampling of assorted stories pulled together for the next release or project can often feel like a patchwork collection of various requirements rather than a full solution to particular customer problems or to address a larger business initiative.

Of course, this is not symptomatic of Agile per se. It turns out that it’s very difficult to provide context behind what you want to build. What are the business goals? What unique user problems are being solved by the capability? How does the new functionality interact with the overall user experience and UI design? Some agile development organizations do not have teams dedicated to exploring the solution space, collaborating with the key stakeholders, and ultimately weaving in all the stories into a unified solution, transforming the stories into something more than the sum of their parts.

It’s important that everyone from stakeholders to the development team to Quality Assurance understands the big picture around what’s being built and how it all relates to the business objectives. For teams where the big picture is not understood up front, it’s a pretty good sign that the upstream part of the process is missing.

**SOLUTION:** One way teams use Contour to help solve this issue is to create business goals for the project and link them to a high-level solution document that communicates the value proposition to everyone. Starting the project with everyone collaborating on and sharing a common understanding of the overall picture and objectives, you can avoid a lot of confusion down the road.

Of course, starting out with a common understanding of the big picture is only half the battle. Once development starts, technical issues are encountered, holes in the specifications are identified and the team gets more clarity on usage with working code to try out, the initial plan may look a lot different from how it started. By having the big picture for reference to anchor the team as decisions are made, the team has the best chance to ultimately deliver something that best meets the original business goals.

## **TWO:** Sprints getting off to slow starts as a result of not having enough definition around what to build.

Agile teams sometimes suffer from not having sufficient definition within the stories in time for the development team to get started on a new Sprint. After all, stories are expected to change and will change as a result of user feedback, so there is no need to spend a lot of up-front time defining detailed requirements, right? Simplicity and lack of formality is encouraged, which indeed works well for certain teams and certain projects.

However, if you are finding that the development team is making assumptions, stretching to close information gaps, idling when trying to get started on a new Sprint, or that Sprint planning meetings run long and raise more questions than answers, and the big picture is not getting transferred to the development team clearly, then these may be signs that you aren't spending enough time up-front defining stories.

Ever been involved in a painful Sprint Planning meeting? Think about the root causes for what went wrong. Sometimes it's that the team is new to Sprint Planning or hasn't established working rules to help make the discussions productive. Sometimes it's an inexperienced Scrum-master or Sprint Leader who is having trouble reining in the team and keeping the focus. But more often than not, the main cause is not being adequately prepared with the story definitions for the Sprint Planning meeting.

Telltale signs that the stories or features during a Sprint Planning meeting are not sufficiently defined are long discussions about "What about this?", brainstorming about UI design options or alternative UI flows, questions about technical feasibility and non-functional requirements like performance.

When there are large variations in estimates among the team or when there are a fair number of stories that are pushed because the estimates are too large or too varied.

Lack of preparedness in product definition can have an adverse impact on the development team well beyond the Sprint Planning meeting. Development teams may be slow to get started with the Sprint underway, since a good number of stories are being reworked. Even the stories that initially seemed good enough may need more work as development tasks get blown out and critical gaps are discovered. It's easy to lose momentum from one Sprint to the next if the team is not ready.

**SOLUTION:** The key to avoiding bumpy Sprint transitions is getting the upstream stories into Contour early, designing and building out the details as part of the story

With a clear understanding of the big picture, the development team can make better decisions during design and coding.



definition process. The overall solution and supporting stories can be reviewed early and often by stakeholders and development team alike to call out the gaps and inconsistencies and issues with the specified behavior and usability or design. The development team can raise awareness of areas that are not technically feasible or are technically difficult to do and in follow-up the team can collectively discuss options and come up with a feasible solution. Using Contour for collaboration and review, stories can be ready for the next sprint with minimal slowdowns and churn. With a more clear understanding of the big picture, the development team can make better decisions during design and coding.

### **THREE:** Big disconnects between key stakeholder expectations and what gets delivered, even with working code continuously available.

Nobody likes surprises. Not stakeholders, not the project lead, and certainly not the development and QA teams – especially at the end of a release or project. Even with an Agile development process, there can be a divide between stakeholders outside the development team and the development team. Specifications get thrown over the fence. The development team does its best to break up the work into iterations based on priorities. Scope and time are negotiated each Sprint. Continuous builds are made available to the stakeholders for eliciting feedback. And yet, it's very possible for stakeholders outside the development team to be unsure of what they're going to get up until the very end. How could this happen?

When reduced to lower-level details, it's easy to lose sight of what's really important - that the sum total of parts will meet the initial business goals.

For one thing, key business objectives can get lost in the decomposition of initiatives into epics, stories, sub-stories and tasks. When reduced to lower-level details, it's easy to lose sight of what's really important - that the sum total of parts will meet the initial business goals. An Agile process can help development teams verify that what they are building works from a user perspective, but this is not the only measurement that applies. While it may present value to the end-user, if it fails to meet critical business goals, the project has not provided as much business value as was originally expected.

**SOLUTION:** Providing upstream context for everyone on the team to reference in Contour helps teams keep their eye on the big picture

throughout the twists and turns of development. Conversations around story details shared in Contour with the ability to trace back to higher-level project goals for

reference provides the right level of access and context to keep everyone involved and aligned.

Another issue is that there is little visibility into the negotiations and decisions continuously being made during the development cycle by stakeholders outside the development team. Sure there are published burn-down charts and boards of various flavors, but most stakeholders are not comfortable translating those into what it really means for the project. The stakeholder wants to know what's changed and how it impacts the big picture or business goals. Are there any issues or changes that they should care about?

Using Contour as a central place of record for all issues and decisions during the life of a project provides visibility into what's going on during development and reduces the risk of surprises for stakeholders as delivery time approaches. Important issues and decisions can be highlighted enabling all stakeholders to stay connected to what's going on during development and help avoid the "Why does or doesn't it do that" or the "I thought we agreed it would do X instead of Y".

It is true that an Agile development process provides all stakeholders, including customers, with early and continuous access to the latest working code. And, we agree that feedback based on kicking the tires of the actual working code is incredibly valuable. If all of your stakeholders have the time and are comfortable with routinely trying out the latest and greatest version on a continuous basis, then it's not as important to communicate changes through Contour, as stakeholders will see the changes by exercising the code. In our experience, stakeholders typically don't have the time to keep checking out the updated code or they may miss key changes when they do spend a little time with the latest code. Also, they may not be knowledgeable enough about using the tool to pick up on the key changes.

For example, the head of the Sales team may not know how to access the licensing module to discover a key change to the access rules for VIP users, but she definitely has an interest in those rules. She should not be excluded from the decision-making simply because she does not have time or does not have deep enough knowledge of the licensing module to discover the key changes on her own. Employing Contour enables all stakeholders to be included in the discussion around key decisions.

## **FOUR:** Critical disconnects discovered later than they should be discovered.

As mentioned in the previous section, at Jama, we are big fans of the Agile concept that emphasizes the delivery of working code over formal lists and documents. However, many teams can ill-afford to make changes after working code has been delivered based on feedback that could have been elicited much earlier in the process when concepts, use cases, scenarios and UI are being figured out. Paper prototypes,

clickable mock-ups, wireframes can all be used effectively to get stakeholder feedback earlier rather than later.

In some cases, getting critical feedback after working code has been built does not leave enough time for the team to take appropriate action around that feedback. You are left with the prospect of either extending the delivery date to address the concerns or delivering with questions marks and gaps. These late disconnects can also lead to the addition of stories in subsequent Sprints that are enhancements or fixes to the original stories built in prior iterations, which can be a costly way to deal with a miss in the definition that might have been caught before the first Sprint.

**SOLUTION:** Contour offers a solution to help push discovery of key gaps and issues earlier in the process. While it is true that no plan survives contact with the enemy, it is well worth the effort to define and design your solution up-front to enable your team to test out assumptions about business goals and user experience early. Issues discovered testing these assumptions can be expensive to deal with after working code is built so it pays to put your team in a position to test them early.

## **FIVE:** Key stakeholders have difficulty organizing and planning around delivery because they don't know exactly what they are getting and when they are going to get it.

You may have a project where various teams outside the development team are impacted by important changes to the scope or dates. These are teams like Executive management, the Board, Product Marketing, Sales, Support and they need to know "What are we going to get?" and "When are we going to get it?" Very often, these teams of stakeholders have no way to continuously track and influence that on Agile projects. To be fair, they typically have no way to track or influence key issues and decisions once a project is in development following any development process.

For these teams, the fluid nature of an Agile development process can sometimes be a challenge. They have dependencies on what's being built and the delivery date. They need to know about changes that affect them at the right level and need to have a place to voice the level of impact and thus have a hand in those decisions.

For example, a UX team needs to be involved in the decision about whether or not to drop a certain story if they designed the navigation assuming that the story would be included. When the idea to drop the story first gets voiced, the UX team should be notified and should participate in the discussion so that the net effect to the design is considered. A Product Marketing team would want to know about story changes that might affect the messaging around the new capability for a product launch. And, both the Product Marketing and Sales Teams have a vested interest in the delivery date to customers, and should have a say in any decisions that might push out the

date. They may need to lock-in a date for the launch campaign or some related media event one month before the final delivery date.

As mentioned earlier, these kinds of stakeholders typically don't know how to read the tea leaves in published burn-down and task boards to glean what they're going to get and when for a given project. While everyone understands to manage releases or add new features, you can either reduce the existing scope or extend the delivery date. Teams like Product Marketing and Sales need to know how to plan their own activities around the delivery, and they have very real needs to understand if the project will be delivered on time and with the right stuff to meet key initiatives they are depending on. These particular stakeholders are usually the least likely to take advantage of a continuous build to play on, for the same reasons as outlined earlier. They don't have the time or the specific usage knowledge to get what they are looking for from exercising the working code.

**SOLUTION:** For these teams, it is essential to have access to a product like Contour that provides a window into key decisions and activities during product definition, and more importantly product development. They have a real need to stay close to and connected with what's happening. Sure they can join daily Scrums and stand-up meetings, but typically these forums don't provide the kind of high-level information that allows these stakeholders to grasp how this affects them and impact of changes.

Further, many Agile development teams make some of the most critical decisions and changes during the course of design and coding tasks, at internal demos, and in demos to external users. It's not uncommon for these decisions and activities to get lost on whiteboards, in emails or sidebar discussions at someone's desk. Either the development team is too busy to document every offline conversation or there is no good place to publish these events in such a way that stakeholders can stay appraised. Having a central place like Contour to go for all discussions, decisions, and events of interest around releases and projects from low-level scope details to release changes is critical to keeping everyone connected for ultimate project success.

## **SIX: You have critical business objectives or non-functional goals that must be demonstrated to get certification or final customer sign-off.**

For some projects, this is not a factor. Delivering working code that satisfies the user needs is good enough and the Agile process alone can get you there. But, there are fixed-bid projects, projects that require compliance to published standards or projects to build critical systems that impact human life such as medical devices and aircraft that require more documentation than simply the working code.

It's important to document the initial scope as agreed to at the beginning of the

contract, then document, review and get approval for changes along the way and report traceability from initial higher-level requirements to lower-level specifications to test results that describe what is being delivered and verifies what is being delivered meets the final specifications. Documentation isn't always necessary and can introduce overhead in a dynamic environment, but for some projects the consequences from not documenting those changes can be disastrous. For these types of projects, you need to show what you initially planned and agreed to deliver, what's being changed and why and get agreement on that. Also, assess impact of changes to help evaluate time and cost risks due to changes.

There are other times where you need to demonstrate compliance to certification standards and requirements, and while the development team can follow an Agile process to manage the risks and changes, Agile by itself can't provide the connectivity and details needed to demonstrate compliance. Similarly, for projects concerned with critical systems where failures could lead to injury or loss of life, financial theft, or confidential data theft, teams need to perform hazard and impact analysis to identify, link, manage and mitigate risks, as well as evaluate impact of changes and decisions to risks.

**SOLUTION:** Contour provides the ability to create unwanted events and risks as items to test and link to requirements and mitigation steps to ensure you have the necessary coverage.

Unit testing alone is not enough for protecting against these kinds of negative outcomes. Special verification activities are derived from these typically non-functional requirements around safety, security, performance to ensure that risks have been adequately mitigated and these items need to be linked for traceability, coverage and impact analysis during change management. Contour is a central repository for defining, tracking, and reviewing non-functional requirements that are every bit as critical to project success as the functional requirements.

## In summary.

Agile provides many benefits to organizations, but it's not without its challenges especially when it comes to the product planning phase and keeping everyone in sync on the details, discussions and decisions that are made constantly as the requirements evolve during the development process. If you are experiencing one or more of the six scenarios we outlined in this paper, then I encourage you to explore Contour with your team on a real project. We offer [videos](#), [free trials](#) and tech support to let you have the keys to experience the tool first-hand. The goal isn't to replace what you're doing, the goal is to support your development team and help them collaborate with other teams and stakeholders more effectively to embrace the benefits of Agile company-wide.



In the mode of transparency, we use a modified Scrum process at Jama and use several tools including Contour, Balsamiq and JIRA with Greenhopper to aid in our Agile product planning and development process. We have a task board and do a daily stand-up with our teams. We hold a retrospective at the end of releases and constantly look for ways to improve how we develop our software. Regardless of the terminology, process and workflow you use today, it will likely evolve over time and we encourage you to adopt tools that will evolve with you.

If you have any feedback for me on the paper, please shoot me an email any time. We're always interested in your feedback: [fcharron@jamasoftware.com](mailto:fcharron@jamasoftware.com).

## About the Author



### **Frank Charron, Director of Engineering**

Frank leads the development team at Jama Software. Frank's experience in software development spans nearly 20 years. Prior to Jama, he worked at K12, Cigital (formerly known as Reliable Software Technologies) and BDM. Frank holds an M.S. in Operations Research from The George Washington University and a B.A. in Mathematics from the University of Virginia. When not writing code, Frank is an avid cyclist in and around Portland, Oregon.

## About Jama Software

Thousands of users worldwide. Billions in R&D projects managed within Contour.

Jama Software is the leader in collaborative requirements management solutions for improving enterprise collaboration and managing complex software development projects. Its Web application, Jama Contour, helps organizations manage the entire requirements management lifecycle through an intuitive, easy-to-use interface that brings people, process and data together to ensure software quality is delivered as specified.

Customers, from agile start-ups to the largest and most sophisticated technology and IT organizations in the world, turn to Jama to help drive innovation, improve the decision-making process and harness the collective genius of all stakeholders involved in building great software. For more information please visit: <http://www.jamasoftware.com>.