



# Agile for when things go wrong: the missing piece of your incident response plan

By adapting the values found in the Agile Manifesto, you can crush incident response and build user trust.



BY SHANNON WINTER

## BROWSE TOPICS

- Agile manifesto
- Scrum
- Kanban
- Agile project management
- Product Management
- Agile at scale
- Software development**
  - Overview
  - Technical debt
  - Testing
    - **Incident response**
    - Continuous integration
  - Design
  - The agile advantage
  - DevOps
  - Agile Teams
  - Agile tutorials
  - About the Agile Coach
- All articles

Agile methodologies are increasingly being used outside the traditional realm of software development in all different business areas – even marketing! So we got to thinking, what does agile look like in the world of incident management? At Atlassian we define agile as a structured and iterative approach to project management and product development. Agile empowers your team to respond to change without going off the rails.

Since bugs in production, incidents, and downtime can definitely be classified as times when things "go off the rails", we figured a methodology like agile -- built to help teams stay on the rails -- would have a natural place in incident management – specifically, incident communication.

## Applying agile values to incident response

While there's no shortage of tools to help your team detect, alert, swarm on, and resolve incidents, tools alone can't replace clear communication to stakeholders. And let's be real: The stakes can be high. Reputation, customer attrition, time spent on damage control, just to name a few. Agile methodologies can help keep these stakes as low as possible.

Many of you are probably already familiar with the four key values of the [agile manifesto](#): 1) Individuals and interactions over processes and tools, 2) working software over comprehensive documentation, 3) customer collaboration over contract negotiation, and 4) responding to change over following a plan. Let's dig into each a bit more and see how they can be leveraged for more agile incident communication.

## Incident Communication Principle: human-centric incident communication

This principle is based on the agile value, *individuals and interactions over processes and tools*. Processes and tools are important for any incident management process, but they mean nothing when viewed as separate from the people trying to use them and the culture that's been formed around them. What's the glue that fills the gaps between people, processes, and tools? Communication, of course!

Communication is crucial when an issue arises, whether it is a small bug in production or a full-blown system failure. Even the most complete incident plan requires frequent communication in order to reach resolution and maintain trust.



Even the most complete #incidentresponse plan requires frequent communication in order to reach resolution and maintain trust.

During an incident affected users are most likely experiencing frustrating – if not downright debilitating – errors and need to know what's up as soon as possible. Many will already be e-mailing, tweeting, and/or filing tickets about the issue, so it is in everyone's best interest to get in front of the situation quickly with a message that

### RELATED TUTORIAL

Learn scrum with Jira Software

[Try this tutorial →](#)

### SUBSCRIBE

Sign up for more articles

Email

[Subscribe](#)

shows you are aware something is wrong and are looking into a fix. At Atlassian we use [Statuspage](#) to communicate to internal and external stakeholders during downtime and think you would quickly find the value too when trying to broadcast incident info to your users in a fast, scalable way. In fact, Statuspage has helped users increase the speed of their incident communication by a **whopping 50%**.

WANT TO GIVE IT A TRY?  
[Sign up or log into Statuspage >>](#)

Once you are in, learn more about best practices for subscribing your end-users and effectively communicating during an incident:

- Scroll through our getting started guide to learn the basics of setting up and managing your status page
- Read up on [incident communication best practices](#)
- Learn how to set up [end-user notifications](#)

But no matter what tool you are using to give your customers the 4-1-1, human-centric communication goes a long way. There are real people on the other side of the problem who rely on your service and rely on you to keep them in the loop when something isn't right. While templates are great in a perfect world, you will need people in place who can craft clear, concise, empathetic, and relevant messages in order to build customer trust even in the worst of times. Take Dyn for example. They had a tremendous outage during one of the largest DDoS attacks in history, yet still had users thank them for the candor they showed during those hours of interrupted service:



As Werner Vogels, the Chief Technology Officer of AWS, said when discussing the large AWS S3 outage in February 2017:

"Customers don't like advice that says 'sit still, don't do anything.' No, that's not what they want, and for that you need to give them really good information, make them understand what's happening, given an expectation of when the service will be coming back online if you have such information."

## Incident Communication Principle: Barrier-free page creation and incident updates

For this principle we look to the agile value, *Working software over comprehensive documentation*. Documentation about your product should be clear and user-friendly and we think incident updates should be too! Your users shouldn't have to read between the lines (or skim lengthy paragraphs) to know what is going wrong and when they can expect a fix. While you do need to put thought into your incident updates and make sure you are being empathetic and human in your communication, you shouldn't let approval gates or multiple revisions get in the way of frequent, honest updates.

Talking a look again at the Dyn incident, you can see their team wasted no time when communicating updates to their users. Over the course of the 11+ hour incident, they updated their status page 11 times (an average of 61 minutes between updates). Their status page enabled them to have one single place to communicate about the incident, rather than spending time finding the mailing lists to e-mail or figuring out how to fit updates into 140 characters on

Twitter. In other words, they got the message out there while still primarily focusing on reviving their service.

The screenshot shows a status page for a DDoS attack against Dyn Managed DNS on October 21, 2016. The page includes multiple update sections with timestamps and descriptions of the attack and the company's response. The updates cover the initial incident, investigation, mitigation efforts, and customer monitoring.

The beauty of an out-of-the-box status communication tool is that you don't have to spend a lot of time getting a solid page up and running. It takes less than 30 minutes to create a status page, and like agile, your status page can and should be iterative. Think about getting a working page live for your customers, and then make it better as you go. After you have a couple incidents under your belt with status page as part of your process, you can make tweaks to improve as you go.



It takes less than 30 mins to create a status page. Like agile, your status page can and should be iterative. #agile  
#incidentcommunication

**Ready to create your own status page?** Sign up or log into Statuspage >>

Don't wait until your next incident to create a status page. Put in a few minutes now so you are in the best possible position when downtime strikes. Remember, you don't need to spend a lot of time setting up a page for it to do the job:

- Learn how to [customize your status page](#)
- Get inspired by [examples of excellent status page design and customization](#)

## Incident Communication Principle: Transparent communication during incidents and beyond

This agile value *Customer collaboration over contract negotiation* is all about working with your customers to provide the best product and experience possible. To us, this means having proper feedback channels set up so customers can voice concerns and alert you of any issue they are having (using tools like Jira Service Management, Twitter, etc.). World-class companies understand that customers are expecting a response to their feedback and want to be involved when it comes to making improvements to your product and your incident response process. Some empathy and an explanation goes a long way – and customers are not timid to ask for it – as shown through these tweets:

The screenshot shows a Twitter conversation between Baron Schwartz (@xaprb) and Atlassian HipChat (@HipChat). Baron Schwartz expresses his loyalty to HipChat and appreciation for its transparency. Atlassian HipChat responds positively, stating they will work to be more transparent. Baron Schwartz replies, expressing appreciation for the postmortem on statuspage for this incident.

**Quizlet** @quizlet · May 16  
We know you depend on Quizlet and we're working hard to make this right. We anticipate this downtime lasting more than another hour

**Quizlet** @quizlet · May 16  
We'll post another update at 3:10 p.m. PT. In the meantime, you can use our mobile apps if you're already signed into your account

**hyzhangsf** @hyzhangsf  
Replies to @quizlet  
It would be nice if there is something like statuspage.io showing the status of this incident and also post mortem for those interested.

2:53 PM - 16 May 2017

**Kian Mohageri** @kianmohageri · 1 Jun 2016  
@circleci mind sharing the tldr of today's perf degradation? confident your engineers will have it fixed but I'm curious about root cause

**CircleCI** @circleci  
Replies to @kianmohageri  
We've not got a post-mortem out yet, but we will soon. We'll share via our statuspage when we do. Thanks!

2:05 PM - 2 Jun 2016

This also means [maintaining transparency around your uptime](#) so users know exactly what they are getting when they sign-up. When you sign-up for a cloud service, you're trusting that service to be reliable. It is not always a physical contract, but rather an inherent contract negotiated between customer and service provider that when things go wrong, the two parties will collaborate to make sure things are resolved quickly and everyone is kept in the loop from investigation to resolution. Which brings us to our final value around responding to change...

#### Incident Communication Principle: Agile retrospectives

Even the best laid plans... well you know the rest. Reminiscent of the agile value, *Responding to change over following a plan*, we know that even the most well-thought out plans will inevitably need to change both during and after an incident occurs. Agile is all about being able to pivot on the drop of a dime and getting quick and constant feedback that improves your product and culture.

Wistia, an internet video and analytics hosting company, learned just how important it was to remain agile during an unexpected incident in 2013 that left their stats infrastructure grinding to a halt. They weren't prepared, leaving them swamped with support tickets from disgruntled customers. Their first pivot was creating their own status page to help make their lives easier in situations like this. But by creating their own status communication tool, they were now supporting a new product in addition to their core product. It became clear this was a cost their 20-person team at the time could not afford. The second pivot was to call their homegrown solution quits and move to Statuspage.

Jordan Munson, Support Engineer at Wistia described the move: "After a handful of months of mild frustration around our nearly featureless, but helpful, home grown solution we decided we needed something more, something that didn't require so much tending to. Enter Statuspage. Since the move to Statuspage, we've been able to do what we were looking to do along -- quickly and easily keeping our customers up to date on the status of our application. It only took one massive outage and building a new product to get there. Fast forward a couple years to modern times and our process looks way smoother. Folks get updates from us directly when there's an outage, they know where to look for updates, and updates made to our status page push directly to a number of places."

Munson's team truly took lemons (the 2013 outage) and made lemonade (a new and improved – and scalable –

incident communication process). This is an agile response to change at its finest.

Retrospectives are also a key part of this agile value. A retrospective gives your team the chance to take a step back and discuss what worked well during your incident communication, what did not work so well, and most importantly, what you will do to prevent the same issues from happening again. Don't don't give in to the temptation to skip over a retro after an incident is marked "resolved" or if you think your team performed great. There is always room for improvement when it comes to incident communication and always a chance to build better rapport and trust with your users.

PROTIP:

Try this [retrospective play](#) from the Atlassian Team Playbook to provide a safe space for your team to reflect on and discuss what works well (and what doesn't!) so you can improve.

Revisiting our first agile manifesto, retros definitely require human-centric communication in order to be successful and