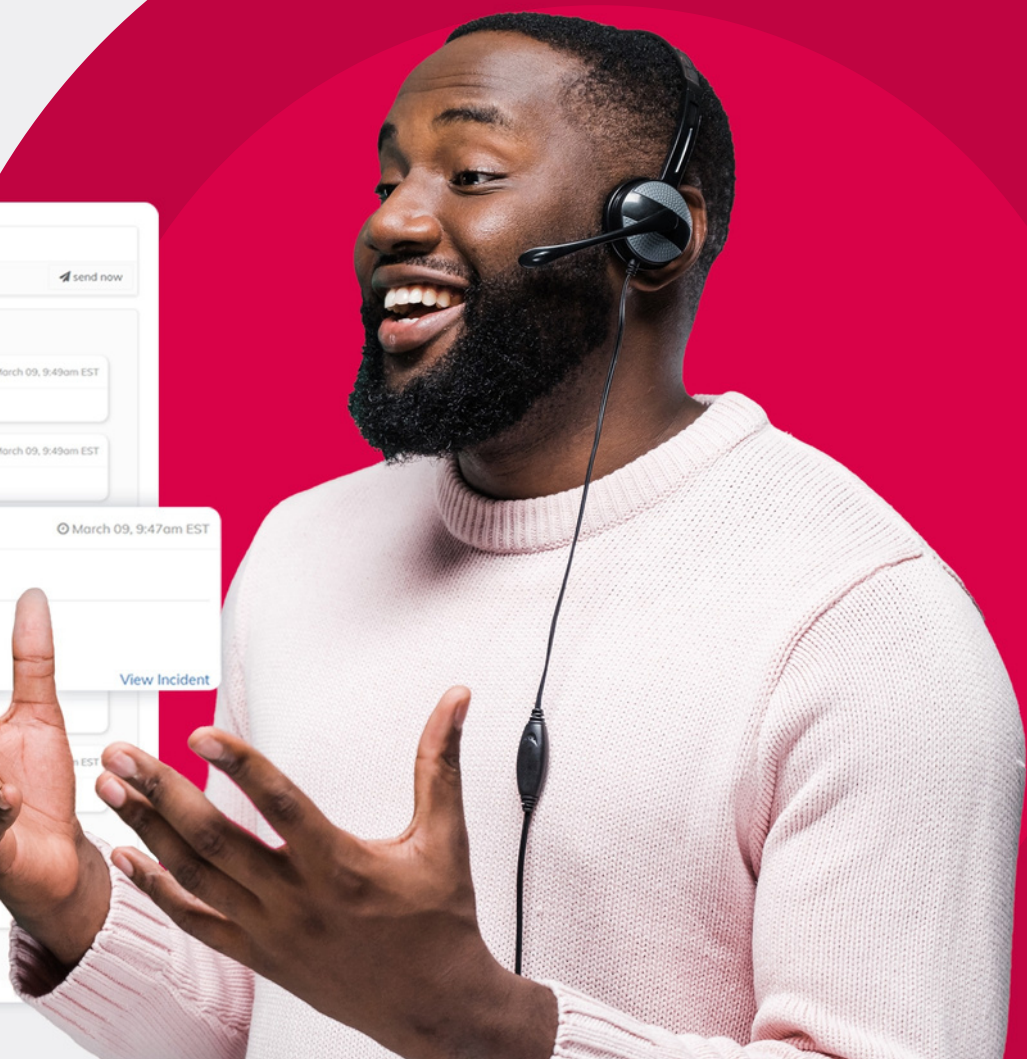
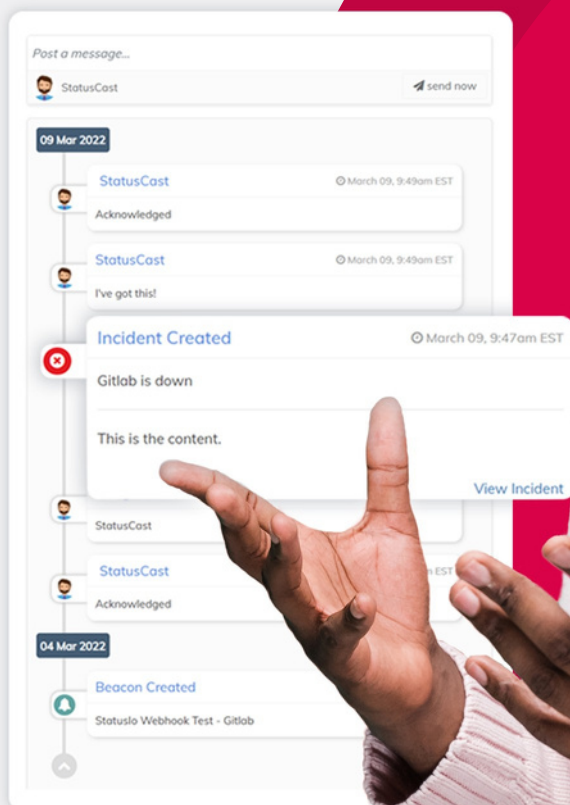




# Best Practices for Communicating Application Downtime

WHITEPAPER



# Best Practices for Communicating Application Downtime

Application downtime impacts everyone involved. No customer or internal corporate team looks forward to the inevitable moment when a critical application is performing poorly; or worse yet, completely inaccessible.

The exact impact of downtime on a business is unquantifiable, because it's more than a problem of time and money. Brand and credibility are also at risk during these times. How you communicate during these disruptions are equally as important as fixing the outage itself.

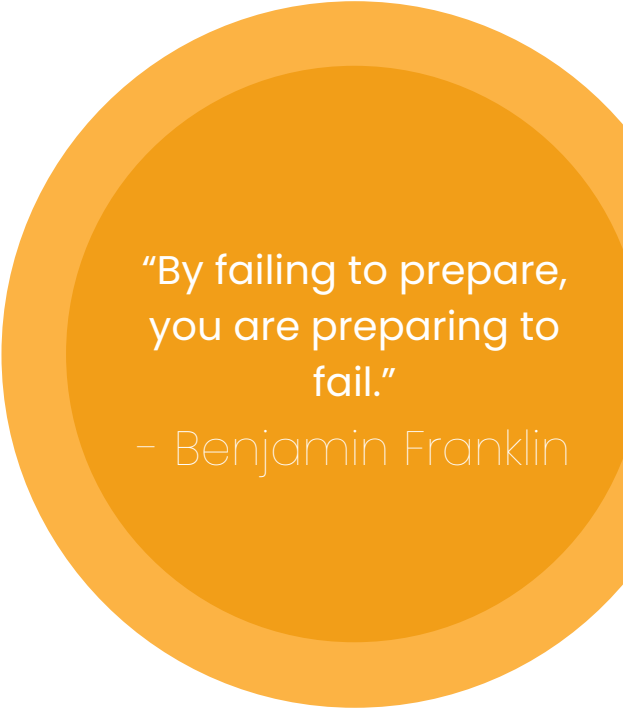
IT departments are typically given the primary responsibility for building and implementing processes to minimize downtime. Large organizations often have highly complex monitoring tools and protocols to alert personnel when mission critical applications or services become unavailable. Once these applications go down or experience performance problems, the race is on to identify the root cause, take corrective action and restore service.

But most organizations forget about one of the most important pieces of managing application downtime:

## How to communicate with the application end-users

For each downtime incident that occurs, there are several questions that have to be answered. For example:

- If your application is used by customers, which customers need to know when an incident occurs?
- If your application is used by employees, how do you reach them?
- How much should be communicated?
- What channels should we use to communicate?
- What is the process we should follow?
- Who should communicate the message?



"By failing to prepare, you are preparing to fail."

– Benjamin Franklin

**THIS DOCUMENT PROVIDES SOUND BEST PRACTICES YOUR ORGANIZATION CAN USE TO EFFECTIVELY COMMUNICATE SCHEDULED AND UNSCHEDULED DOWNTIME.**

## Plan for the inevitable

A little planning goes a long way when dealing with downtime. Develop a communication plan that outlines your strategy for when a situation does occur.

Your plan should outline the process for communicating to internal and external stakeholders. Internal approval processes should be identified along with the communication channels. It should outline the strategy around messaging and timing according to the severity of the situation. The point is to avoid reactionary responses when problems occur.

Minimally you should consider the following concepts when developing your strategy:

- How to keep an up to date list of the Employees, Customers, 3rd parties requiring communication.
- Predefine appropriate messages for each potential incident.
- Determine which messages need to be approved before disseminating the information.
- Determine which department is responsible for delivering the message depending on the type of incident.
- Given the length of the outage, determine how often your communications should occur.
- Determine the best channels of communicate for your constituency.
- Developing an emergency plan helps educate stakeholders and set guidelines for engaging customers.

## Know when you have a problem

End-users are quick to notice when a mission critical service experiences downtime (often times beating even your most sensitive monitoring services to the punch).

Ideally you want to know about a problem before they do. Companies tend to rely on system management and monitoring systems to provide real-time detection of an application problem. Have a clear understanding of how you are measuring application availability and performance. For most complex systems the process to determine an outage or poor performance is a combination of automated monitoring and manual verification.

Develop and document the process to determine how you handle an incident. This process should include answering these questions:

- What is your source of truth?
- More importantly, what will be the perceived severity of the situation by your end-users?
- What is the actual severity of the situation?
- What is the process to verify before escalating?

**Consider which incidents be escalated to senior management for review before notifying customers. Ensure that everyone understands the protocol and who will be the voice to the customer.**

## Communicate early – Communicate often

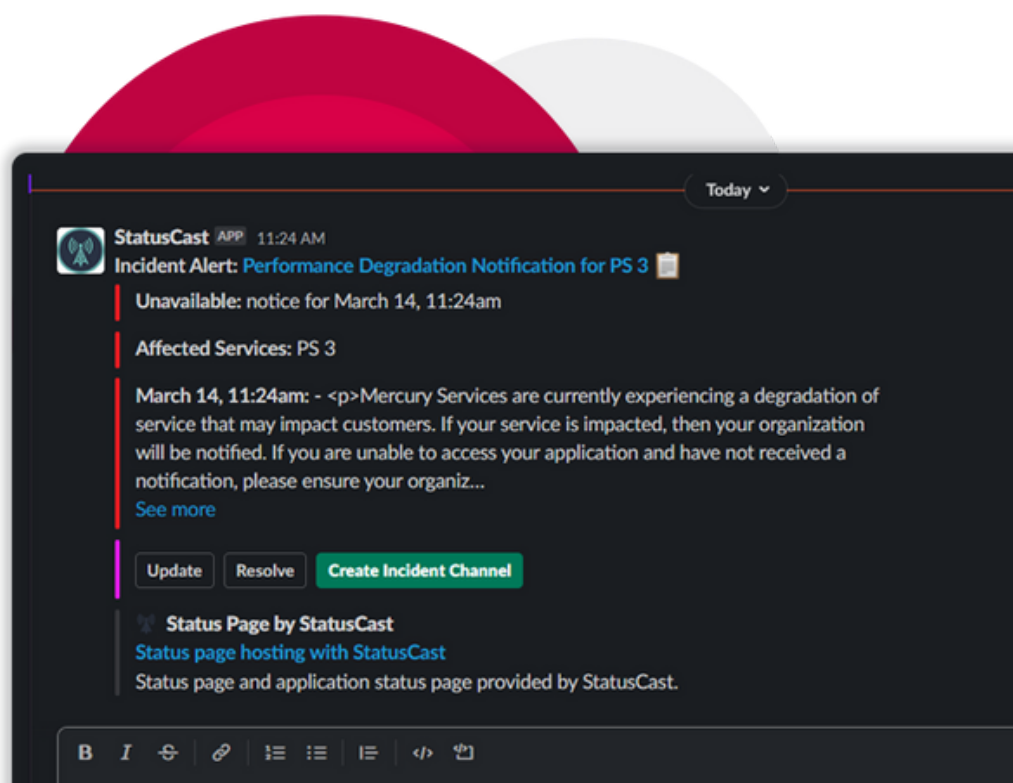
You want to break the news before someone else has the opportunity, otherwise it will likely be a customer. Communicate as early as you can, even if it's a simple message that you are aware some users may be experiencing intermittent problems, and that you are looking in to the issue. You may not have all the facts yet, and that is OK, rapid communication and honesty keep end users loyal. In some ways this is the most delicate moment in the communication process.

A good example of this is a delayed flight. Just think about how you would feel as the plane pulls away from the gate and then just stops and sits on the tarmac. You look at your watch and see that it's well past your scheduled departure time and you're just sitting there – and worst of all there isn't any word from the crew about why you're not moving or how long you will be delayed. No announcement -- just waiting. People start to talk and ultimately get upset. An hour goes by. Finally the captain makes an announcement that maintenance has been called. The flight is delayed for a few hours. By this time everyone is outraged. If customers were told upfront they would be much more likely to sympathize with the problem.

But instead, you have a bunch of unhappy customers who are now using email/text/Twitter/Facebook to vent their frustration and share their experience with the world. You have the same situation when your application experiences a problem. Get in front of it early and commit to providing updates.

Once you establish communication about an incident, commit to regular updates. The frequency of updates should largely depend on the type of incident and its time frame. For example, you may choose to provide updates every half hour for a long running outage.

Smaller incidents may not require more than the initial acknowledgement and an updated notice when the problem has been resolved. The most important thing you can do is to let your customers know when they should expect an update. Don't keep it open ended.



## Dealing with Scheduled Maintenance

Often, outages are planned. In fact, in most organizations, outages are typically not caused by something going wrong, but because some kind of IT operation required your team to take a system temporarily offline.

Communicating scheduled maintenance is just as important, if not more important than alerting users to unplanned outages.

### CASE IN POINT:

A few years back, Yahoo was on the hunt to be the comeback king, but in December 2013 they had an epic fail with communication to their users.

A twitter user tweeted, "Very frustrated!!! My account has been down since Monday. Apparently due to "maintenance", with no heads up. Terrible!". Herein lays the problem with companies that have only implemented application monitoring tools– there is no end user communication or no "heads up", as Wendy put it – at all

**With scheduled maintenance, there are some additional factors that you need to consider when communicating with end-users:**

- How much advance notice is required? Many times, SLA agreements have this metric in place so make sure that your process takes in to account all of your active contractual requirements.
- How many times between the first notice and the actual maintenance event should you remind the customer?
- Is any downtime actually expected? Many of today's environments allow for the deployment and upgrading of system components without effecting your end users. Before performing scheduled maintenance where no downtime is expected, but could occur; a risk assessment needs to be done.

If the chance of downtime exceeds your organization's comfort threshold, consider notifying your customers of the "planned system maintenance, with no expected downtime". However, warn them that unforeseen events may cause something unexpected to happen.



## Be a journalist

Journalists evolve the story as they collect additional information. The final version of the story may not reflect the original version.

People generally understand the fluid nature of unexpected problems. In the absence of a solution we look for truthful information.

Let your end-users know that a problem is occurring even when you don't have the details.

Ultimately they will need details and you need to accommodate them during and after an incident. This is where you need to think like a journalist. To do this you need to answer the 5W's: Who, What, When, Where and Why.

### Who

Is every customer or end-user impacted or only a certain subset? Clearly there is no need to communicate with everyone if only a few are affected. Make sure you understand the scope of who is involved and communicate with those users.

### Where

Is your application distributed in multiple locations? If so, identify which location or locations are experiencing the issue.

### What

Outline the problem. Is your entire application down or specific functions? Often customers experience an issue with a specific feature and assume the entire application is not working properly. Communicate what is working properly and what areas of the application are experiencing issues.

### Why

Provide end users with an understanding of why the incident happened and any circumstances that contributed. Was it avoidable or outside of your control? You may not know at the onset of the incident. Keep your users informed as you gain more information as to why the incident occurred.

### When

End users need to understand when a problem started and when it ended. Otherwise you run the risk of other issues being attributed to the wrong incident. Be clear about when you believe a service incident started and when the issue was resolved. Provide a range if you are unsure.

## Establish multiple lines of communication

Make sure you are heard. People expect to learn about important news through multiple channels. Push out the message on as many mediums as possible including private and, if appropriate public, feeds: Twitter, Facebook, e-mail, SMS. Give your customers the choice to be informed on their terms.

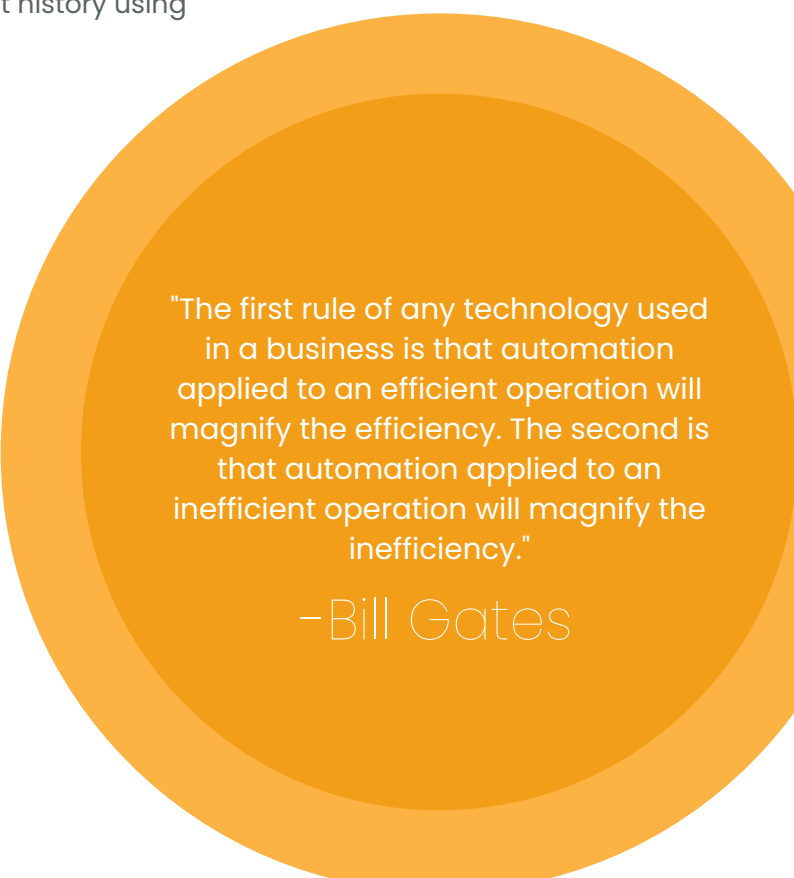
## The Application Status Page

Most importantly, every application should have a status page made available to your end users. Many large Software-as-a-Service vendors already provide such a page, your applications should have one as well.

Implementing a self-service status page provides customers with current and historical application uptime information. Use a system that pushes the user a notification to check your corporate status page. Encourage your end-users to go there first if they suspect an issue. Providing transparency causes your support requests to go down. Consider including historical uptime percentages on your status page. This gives your customers an overall perspective of the health of your applications. A great example JungleDisk.<sup>1</sup> They provide uptime metrics and application incident history using StatusCast's status page service.

If you provide a public facing application you need to embrace social media. Your customers expect it. Twitter has become a key platform for vendors and customers to connect. Set up a customer service or IT Operations account and use a unified communications platform to communicate operational issues to all social channels, SMS and e-mail at once

Consider whether you will mandate a broadcast email to everyone or provide an option for customers to subscribe. Your status page is a great place for customers to sign-up for future updates on downtime and scheduled maintenance.



"The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency."

–Bill Gates

<sup>1</sup> JungleDisk, public status page can be found here:  
[status.jungledisk.com](https://status.jungledisk.com)

## Automate as much as possible

The technology sector has evolved into continuous decisions around “build vs. buy” and “automation vs. manual” processes.

Downtime communication has largely been an area where both concepts have been underserved. As a result, most organizations rely on internal solutions and manual processes.

Can you use alerts from your IT operation’s processes to initiate the appropriate communication workflow? Do you have a solution in place to efficiently communicate with your customers across communication channels? Review your internal processes and systems to determine how you can automate, integrate and expedite communication. At the same time, do a build/buy analysis. Most organizations are not aware that solutions exist to solve this communication problem.

## Wrapping Up

Get the word out to your customers and end users when you experience downtime or performance issues.

Be prepared ahead of time. Implement an efficient process and communicate truthfully. Know that this is not only an IT issue, and that it is important that a system is in place where customer service has some control of the messaging. It matters to your brand, your customer relationships and to your bank account.



PO Box 1141 Columbia,  
MD 21044

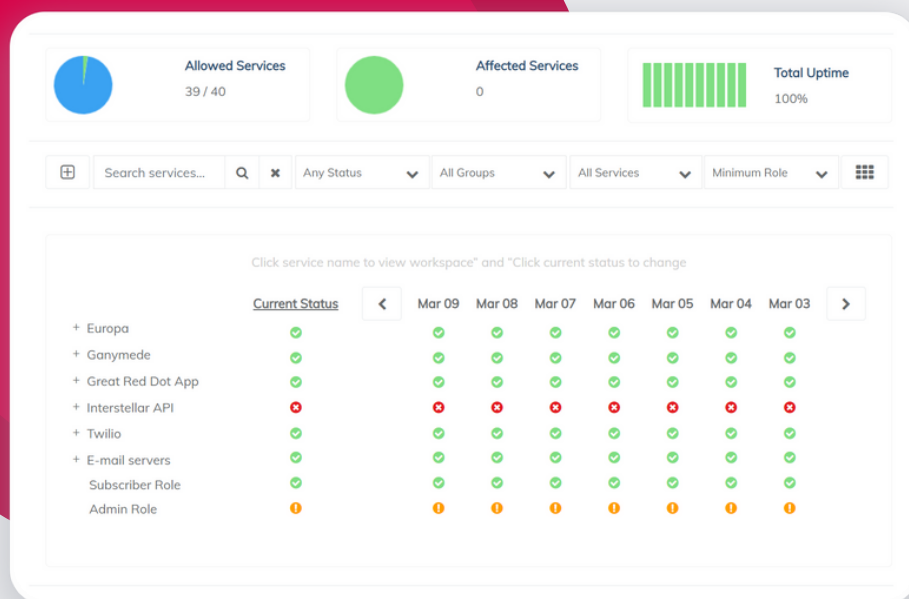


833.STATUSC(AST)



[sales@statuscast.com](mailto:sales@statuscast.com)





IDC Report suggesting  
Fortune 1000 companies lose  
**\$1.25B – \$2.25B per year**  
in application downtime.





This whitepaper was provided to you by StatusCast, Inc. Established in **2013**, StatusCast is a leading status page vendor that offers **SOC-II compliant** enterprise class services to both **Fortune 500 organizations**, and small businesses.

If you would like to learn more about how our incident management and status page platform allows your IT Team to bring their 'A' game every day, please feel free to contact us:

**[statuscast.com](https://statuscast.com)**

