If you already own a business yourself, such as an online business you probably noticed that incident classification helps a lot. It doesn't matter if you are using your own incident classification system or tools external tools. Essentially, incident classification allows you to **select issues that require immediate attention** against those that can wait. Picking up such elements during sprint is not really compatible with clean Scrum, but can be compatible with other agile methodologies like Kanban. Because of that the perfect setup is when incident response team is completely separate from development team. If it is not, you might be forced to use rather Kanban or Scrumban instead of clean Scrum to run both proper incident management and develeopment simultaneously by the same team.

Incident Definitions First

No matter which aspect you are looking for in this article – if you are someone who wants to sign SLA agreement, or you want to establish how fast software house should respond to incidents, or if you need to introduce some standarization into your business internally – first thing you need are definitions. In my opinion in minimum setup definitions should have following elements:

1. Short tag for definition describing it's urgency
2. Long descripiton how issue impacts system to fall into category definition
3. Expected resolution time – in what time issue should be solved on production

But it would be more complete if you add a bit of taste to that aspect, such as:

1. Examples for each tag – to better understand definition
2. Expected first response time – that is not resolution, but an indication that "we're working on it" or "since it's a security issue, we've temporarily disabled system"
3. Escalation rules – if incident is wrongly classified, what needs to happen to escalate it to higher level
4. Basic communcation channel
5. Communication fallback channel – if nothing happens after due "expected first response time", where to call

A typical definition table could look similar to this:

Category	Descriptive Incident Definition	Incident Examples	Response time	Resolution Time	Default Assignee
Critical	The impact of the reported incident is such that no customer is able to use the Software. Full loss of function or performance affecting all numbers of users. No workaround exists.	Example 1: Website is hacked and homepage content of the website is replaced with hacked image. Example 2: It is detected, that user found security breach and can get data of all other users. Example 3: Due to error in DNS change website is not available worldwide	1	4	criticals@example1.com

Category	Descriptive Incident Definition	Incident Examples	Response time	Resolution Time	Default Assignee
Major	The impact of the reported incident is such that a number of customers are unable to either use the Software or reasonably continue work using the Software. Major loss of function or performance affecting all/ large number of users. No workaround exists.	Example 1: End user in shopping system cannot pay for what's in his basket with one of the means (example: payment by credit card is not working). Example 2: End user cannot create new account. Example 3: Manager user cannot take down product, that is out of stock	2	8	majors@example2.com

Category	Descriptive Incident Definition	Incident Examples	Response time	Resolution Time	Default Assignee
Medium	Important features of the Software are not working properly but there are some alternative solutions. While other areas of the Software are not impacted, the system is impaired – some functionality is unavailable.	Example 1: End user cannot leave a comment for his order, but can write an email for support, or will be able to perform such action later on. Example 2: Manager cannot change content or product description in any way, but can temporarily disable product, so problematic content is not visible.	4	48	mediums@example3.com

Category	Descriptive Incident Definition	Incident Examples	Response time	Resolution Time	Default Assignee
Minor	Minor features of the Software are unavailable, or good alternative solution is available, or non-essential features of the Software are unavailable. The customer impact, regardless of product usage is minimal.	Example 1: Manager wants to change the picture on the slideshow, but this picture is not scaling properly. Manager can switch back to the original picture. Example 2: After manager changed style on product, style is not displayed properly, but text is visible. Example 3: User did not receive email with invoice on his email.	24	120	minors@example5.com

Category	Descriptive Incident Definition	Incident Examples	Response time	Resolution Time	Default Assignee
Support	Client submits an information request about a possible support task which has no operational impact. The implementation or use of the Software by Client or its customers is continuing and there is no negative impact on productivity. This includes also all tasks that can be managed inside CMS without any new feature development.	Example 1: Research if issue submitted by Client is an incident. Example 2: Change of text or product description on page. Example 3: Change of image/graphic on the page. Example 4: Perform system update or plugin update. Example 5: Manager needs instructions on how to change product price	48	120	supports@example6.com

Category	Descriptive Incident Definition	Incident Examples	Response time	Resolution Time	Default Assignee
Feature	Client submits an information request about possible software development or documentation clarification which has no operational impact. The implementation or use of the Software by Client or its customers is continuing and there is no negative impact on productivity.	Example 1: Client would like to add section with slideshow on homepage. Example 2: Client would like to add new method of payment, while current payment methods are working properly. Example 3: Text is too large on mobile forcing user to scroll. Example 4: Additional subpage with contact form is needed by client	168	336	features@example7.com

Now if you have incident definitions, we can move on to use cases.

Use Cases for Incident Definitions by Business

I am Business Owner working with Software House

If you are a business owner, and software house is delivering some kind of software to you, then your users at some point will encounter some kind of issues. In that cases you would expect software house to act fast and reliably depending on how important the issue is, so that interference with current sprint is minimum. That’s why overestimating incident level will be counterproductive and will just result in

software house reducing issue priority.

On the other hand underestimating issue importance is not good aswell. For example if you see security issue that may lead to data leak, then it’s definitely a critical issue. Underestimating in such case may lead to legal issues and even more problems. So it would be best of course if classification would be as close to true, as possible.

When you know you need to post an issue, you should also know where or how to send it to software house. Some may provide special form (possibly even with AI that helps you to categorize issue level), other want to have it pushed via Slack or Asana, but on critical it may be best just to call or send email to designated email address. If you have SLA but you don’t know channel – ask your software house (or it might be indicator that such channel does not exist, thus you should look for another software house).

I am Business Owner handling User Issues

If you are runninng your own online service, then you will at some point receive some kind of user complaints. You better be prepared for this – even if users do not have SLA, you should provide some kind of support and handling user requests – even if they mostly won’t send you critical issues, but rather requests for help.

One way to handle it nicely is to connect to external ticket handling system. These are nice and easy to start with, but there is one problem: they take you outside your platform so user immersion drops (user is taken outside our space and may loose interest in your platform or with filling issue details). It is better if you can keep your user on platform, but this requires custom integration or developing such ticketing system. Third way – if you are looking either for budget option or customizing support flow – might be N8N automation in connection with NocoDB.

One thing to remember is despite user sometimes even best intentions they might not be able to to describe issue precisely enough. That’s why two elements are important. First: ask for conact or tie issues with users. Second: make sure you are using Sentry and Clarity or Hotjar so you can track down the issue.

I am Software House providing Support or SLA

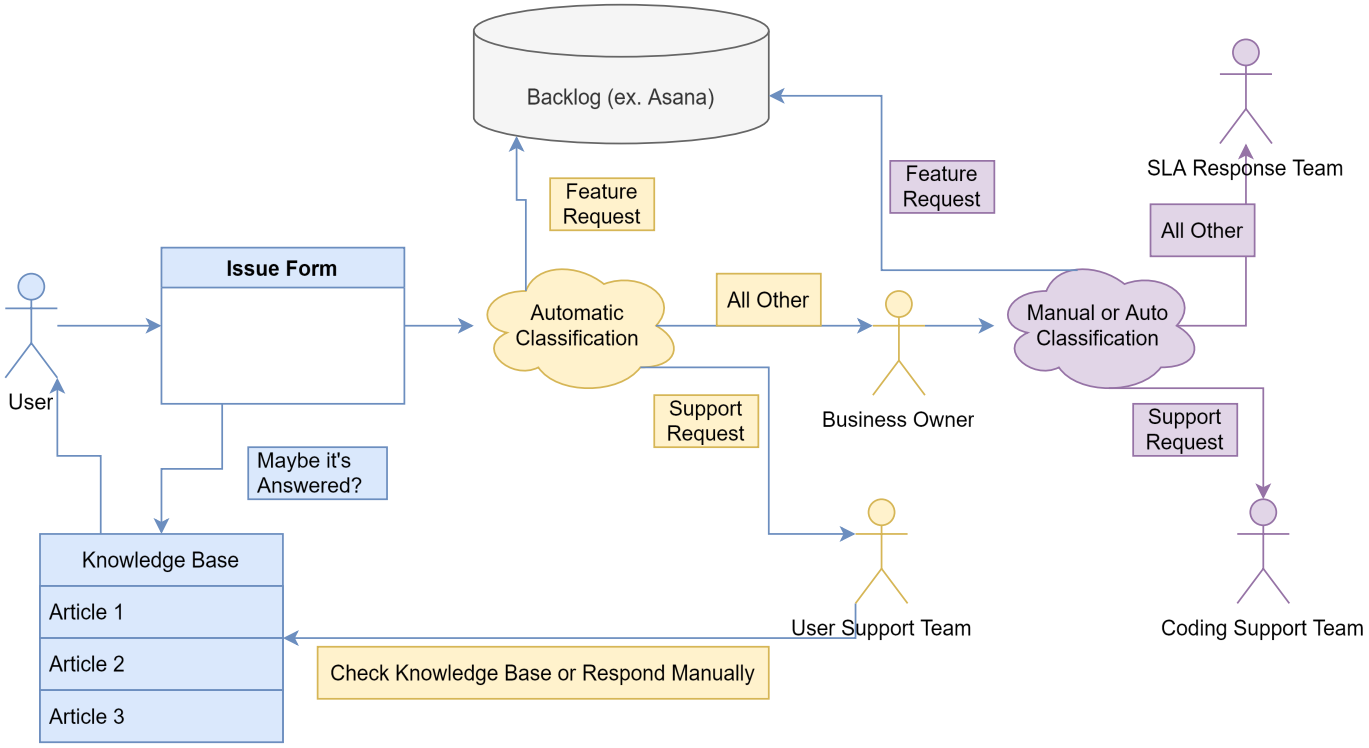
In such case you should esure that your SLA or support agreements are clear and

precise, and that clients are using proper channels to communicate with you. Also make sure that you are using proper tracking tools, such as Sentry or MS Clarity, so that you can reproduce given issue. Also Sentry is important in that aspect that you can track down issues which users won't report. In my opinion it is absolutely essential tool.

Streamlining Issue Distribution Process

You can see some kind of pattern here: user sends data to business owner, who sends data to software house. But there is a catch: not every issue **is** an issue. It can be a support ticket or it can be a feature request aswell.

And no matter if you are a software house or business owner, you should understand **whole** process that issue is pushed through. It may be helpful to review full support path and visualize it, possibly with consultation of the second party. Such process may look similar to this one (note colors that are identifying area of responsibility!)

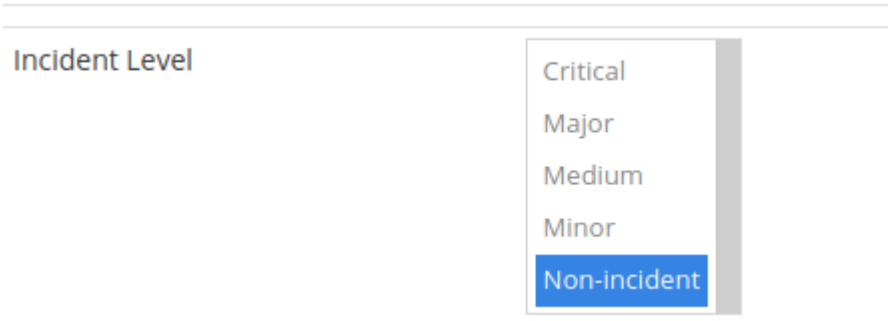


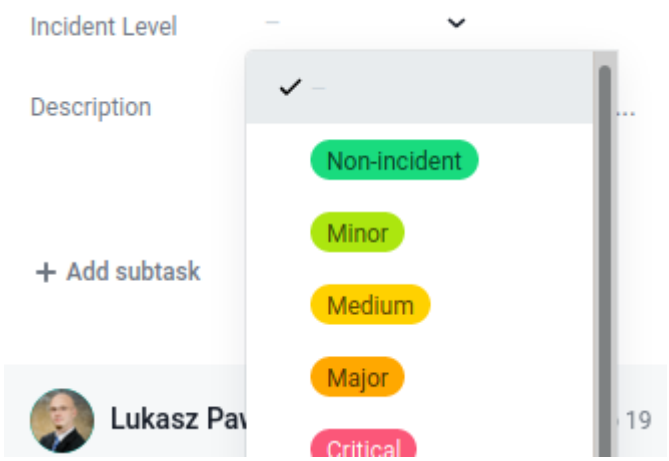
Incident Classification in Sailing Byte

Do you have any problems with Incident classification that we have introduced in SLA packages? Incident classification is valid as a method of specifying submitted tasks in via Asana or other agreed method.

Please remember that this description is only a suggestion and ultimately is up to our support team to decide:

- if the incident actually occurred or it’s only a non-incident support task;
- which level of severity occurred;
- is there a possibility to lower incident levels thus give us a longer resolution timeframe.





Non-incident Support Tasks

Client submits an information request about a possible support task which has no operational impact. The implementation or use of the Software by Client or its

customers is continuing and there is no negative impact on productivity. This includes also all tasks that can be managed inside CMS without any new feature development.

Examples:

- Research if issue submitted by Client is an incident
- Change of text or product description on page
- Change of image/graphic on the page
- Perform system update or plugin update
- Manager needs instructions on how to change product price

MINOR incident

Minor features of the Software are unavailable, or good alternative solution is available, or non-essential features of the Software are unavailable. The customer impact, regardless of product usage is minimal.

Examples:

- Manager wants to change the picture on the slideshow, but this picture is not scaling properly. Manager can switch back to the original picture.
- After manager changed style on product, style is not displayed properly, but text is visible.
- User did not receive email with invoice on his email.

MEDIUM Incident

Important features of the Software are not working properly but there are some alternative solutions. While other areas of the Software are not impacted, the system is impaired – some functionality is unavailable.

Examples:

- End user cannot leave a comment for his order, but can write an email for support, or will be able to perform such action later on.
- Manager cannot change content or product description in any way, but can temporarily disable product, so problematic content is not visible.

MAJOR Incident

The impact of the reported incident is such that a number of customers are unable to either use the Software or reasonably continue work using the Software. Major loss of function or performance affecting all/ large number of users. No workaround exists.

Examples:

- End user in shopping system cannot pay for what's in his basket with one of the means (example: payment by credit card is not working).
- End user cannot create new account.
- Manager user cannot take down product, that is out of stock.

CRITICAL Incident

The impact of the reported incident is such that no customer is able to use the Software. Full loss of function or performance affecting all numbers of users. No workaround exists.

Examples:

- Website is hacked and homepage content of the website is replaced with hacked image.
- It is detected, that user found security breach and can get data of all other users
- Due to error in DNS change website is not available worldwide

Why do I Need to Pay for Incident Handling Additionally?

When we are doing development, you are paying for development, and that's what's included inside agreement. Also, as a software house, we are often sining "maintenance" agreement, that helps to develop small features and respond to issues without legal or financial consequences on specific due dates. Signing SLA enforces to have SLA response team ready and it costs additionally to run such team and keep it ready, along with any procedures that need to be kept in place. Although, there are benefits of paid SLA packages aswell.

Ensured Fast Response and Resolution

Paying for Service Level Agreement (SLA) incident handling guarantees that when a problem arises—such as a system failure or software bug—you have direct access to specialists who are contractually obligated to respond and resolve issues within a specified timeframe. This means your business can minimize downtime and quickly restore normal operations, which is crucial for maintaining process continuity and customer satisfaction. If prolonged wait for response can cause you legal or financial problems, then SLA is probably a good direction.

Clear Accountability and Defined Expectations

An SLA sets clear expectations for service quality and response times, establishing accountability between your company and the software provider. If the provider fails to meet these standards, the SLA offers a framework for addressing the issue, including potential penalties or escalation procedures. This clarity helps avoid disputes and ensures both parties understand their responsibilities. You can see such aspects in incident defintions mentioned above.

Reduced Business Risk and Financial Impact

System outages and unresolved incidents can lead to significant financial losses. By investing in SLA-backed incident handling, you reduce the risk of extended outages, protect your revenue, and safeguard your reputation.

Continuous Improvement and Proactive Monitoring

SLAs often include regular reviews and performance reporting, encouraging providers to continuously improve their services. Proactive monitoring and automated incident tracking help detect and address issues before they escalate, ensuring your software remains reliable and efficient.

Access to Expert Support and Consultation

With an SLA, you gain guaranteed access to technical support, including hotline assistance and expert consultation. This ensures that your team can get help not only during emergencies but also for everyday operational questions, further reducing the risk of disruptions.

SAILINGBYTE

Summary Benefits of SLA Packages

Benefit	Why It Matters
Fast Response & Resolution	Minimizes downtime, restores operations quickly
Accountability & Expectations	Defines roles, reduces disputes, ensures compliance
Risk & Financial Protection	Prevents costly outages, protects revenue
Continuous Improvement	Drives service quality, proactive incident management
Expert Support	Ensures access to specialists and technical guidance

In summary, paying for SLA incident handling by a software house is an investment in business continuity, customer trust, and operational excellence. It ensures that when problems occur, you have the support, structure, and accountability needed to resolve them efficiently and effectively. In comparison to “maintenance” agreement, it helds accountability on resolution time and response based on definitions.