How to use tags strategically to enhance prioritization and categorization

Tags can be used to attach the following essential information to your exceptions.

Checklist	To categorize	Description	Example
☐	Priority or SLA	Assign priority levels to highlight the urgency of an issue.	P1, P2, P3 & P4
☐	Environments	Differentiate between environments where errors occur	Beta, test, stage, prod etc
☐	App areas	Tag errors related to major sections of your app	key-page, checkout, payment
☐	Important steps/process	Tag errors occurring at specific steps within critical processes	step1-shoppingcart, step2-billinginfo
☐	Brand/Store Details	Identify errors associated with individual brands, stores, or merchant IDs and their locations	store123, branda-store123, store123-US
☐	Geographical Info	Categorize errors by country, state or region	USA, GB, AUS, CA or WA, FL, CA
☐	Content Security Policy (CSP)	Use tags to track and detect CSP violations	CSP
☐	Google Tag Manager Integration	Sync GTM tags with your Raygun data for seamless integration	GA - Page View

Read the full article at: http://https://raygun.com/blog/tags

Step-by-Step guide to optimizing error management with Raygun tags

- [ ] **1. Identify user-generated errors:** Recognize that not all errors are code-related; some may be user-generated or beyond your control. You could also use the "unhandled" filter in Crash Reporting to differentiate between handled vs unhandled exceptions. The best way to approach this step is to get all of your teams together (from Product, Engineering, Devops, Support, Marketing and Analytics) and put together a tags classification plan inside Raygun, so that the tags feature can support everyone's use cases.

- [ ] **2. Launch a cross functional initiative:** Initiate a company-wide effort to reduce error noise and improve response times. OpenWater called theirs the" **Raygun Error Zero Initiative"**.

- [ ] **3. Utilize tags:** Create unique alphanumeric tags for exceptions outside of your control to classify errors and determine their significance. This will make it easier for your non-technical to understand how they should triage the issue.

- [ ] **4. Sort errors in the Crash Reporting dashboard:** Use the Raygun Crash Reporting dashboard to categorize error statuses as 'Active,' 'Ignore,' or 'Permanently ignore' based on their importance.

- [ ] **5. Improve error messages:** Write more user-friendly error messages to empower users to troubleshoot common issues independently. We cover this topic extensively in this article.

- [ ] **6. Prioritize error resolution:** Focus on addressing the top 10% of recurring errors rather than attempting to fix every error.

- [ ] **7. Allocate time for error resolution:** Dedicate time to both fixing errors and enhancing error messages, with the goal of improving error handling gradually. As a guide 25% of your engineers' time should be spent on proactively fixing issues.

- [ ] **8. Regularly check the dashboard:** Proactively monitor your application's health by regularly reviewing the Crash Reporting dashboard, checking for recent activity and error spikes. Setting up custom alerts is another way to make sure you are keeping track of the health of your apps.

- [ ] **9. Connect Raygun to your existing development & support workflows:** Consider integrating Raygun to your existing error resolution and support workflows, alongside Slack, JIRA, GitHub and other support tools to manage various aspects of your software development and customer support.

- [ ] **10. Evaluate progress:** Continuously assess the effectiveness of your error reduction efforts, understanding that achieving zero errors is not the primary goal, but rather managing errors effectively. For those that are focused on improving the user experience of their apps, you might want to learn more about our Crash Free Users metric that gives you the ability to not only measure the user experience score of your website but allows you to track its trend over time.