


☐

I'm not robot


reCAPTCHA

Continue

Analytic hierarchy process pdf

Software engineer from Italy, creator of Inspector.dev, passionate bassist. Hi, I am Valerio, software engineer from Italy and CTO at Inspector.dev in this article I will show you how to turn on analytics in the dark side of your application: background Jobs and scheduled Artisan commands execution. Laravel offers great architecture for queue Jobs and scheduled cron tasks. I myself use it a lot. They allow us to run some piece of code asynchronously from the background from a classic HTTP request cycle. Often they are a cause of concern for developers because their execution is not related to direct user interaction. They can go wrong for days and there will be no way to know it until we manually check their results typically within logs. If something goes wrong during an HTTP request, it will appear red bubbles or messages that instantly inform the user of the problem. It is quite easy to discover relevant errors before taking the software into production through automated tests, or using the application itself. If a queue Jobs or a scheduled Artisan command fails, it will do silently, without anyone falling out. Inspector is a composer package for adding real-time monitoring in Laravel applications. It's very easy to install and use, and it takes less than one minute to get started. Let me show you how it works. Installing InspectorInstall our composer package via composer: composer requires inspector-apm/inspector-laravelConfigure to give the API keyGet a fresh API key by signing up for Inspector (and creating a new application, it only takes a few seconds. Select and copy the API key: Put the API key in your environment file: INSPECTOR_API_KEY=13c371c434XXXXXXXXXX1By default inspector will monitor all background tasks: Queue JobsScheduled Artisan commandsNotificationsEmailsUnhandled ExceptionSYou can also enable HTTP request monitoring. Check out our official documentation: you'll see transaction streams in your project's dashboard: and for each transaction you can monitor what your application performs in real-time: Add custom segmentsThanks to Inspector you can put everything you want in your timeline to get a real-time feedback on executing a code block in your Job or in an Artisan command: Http calls to external services (Data Payment, etc.)Function dealing with files (pdf, excel, images)AlgorithmsThanks at the package allows you to add custom segments in your timeline, in addition to those detected by default, to measure the impact a hidden code block has on a transaction's performance. Suppose to run an external http request in your code that is not present in the timeline by default. Use Inspector <?php \Inspector::addSegment(function() using (\$guzzleClient) { {}, 'http'); This will produce a new segment in the timeline and now you can understand what your code is executing and its performance in real time: Exceptions WarningBy defaults, every unhandled exception fired into your Laravel app will be automatically reported to make sure you've been alerted to unpredictable errors in real time. I wish that every change I make to my code can be perfect. But the reality is, that's not always the case. Some errors appear immediately after an update, while others appear unpredictable. However, inspector automates the detection of unknown issues, especially in background tasks where it's even harder to investigate, so I no longer have to manually check the status of my apps or wait reports directly from users. Inspector allows you to manually report an exception if you want to be aware of it: <?php try { throw new Exception('Fragile code. '); } catch (Exception \$exception) { \Inspector::reportException(\$exception); } } If something goes wrong in your code you will be alerted in real time in your inbox and the exception will be monitored for all subsequent events: ConclusionWhen background Jobs or scheduled commands are involved to require a true picture of what happens may require hours or, based on my experience, even days. Inspector can make a big difference in terms of efficiency and productivity. By deleting monitoring to an autonomous tool, you can avoid concrete to lost customers and money due to unexpected technical issues in your applications. Thank you so much for reading this. Don't hesitate to share your thoughts on the comments below or fall into live chat on our website - !Previously published at Hacker Noon Create your free account to unlock your personalized reading experience. Process mining is a methodology whereby organizations collect data from existing systems to objectively visualize how business processes function and how they can be improved. Analytical insights that come from process mining can help optimize digital transformation initiatives across the organization. In the past, process mining has been used the most in manufacturing to reduce bugs and physical labor. Today, as companies increasingly take upcoming automation and AI technology, process mining has become a priority for organizations in every industry. Process mining is an important tool for organizations committed to continuously improving IT and business processes. How does process mining work? Process mining starts by evaluating established IT or business processes to find repetitive tasks that can be automated using technologies like (RPA), artificial intelligence and machine learning. By automating repetitive or everyday tasks, organizations can use efficiency and and - and free workers to spend more time on creative or complex projects. Automation also helps reduce inconsistencies and errors in process outcomes by reducing variants. Once an IT or business process is developed, it's important to consistently look back to ensure the process delivers appropriate outcomes – and that's where process mining comes in. For example, an IT department might decide to automate its help desk ticketing system. Previously, an employee will take time to review a ticket, determine the correct category and assign it to the right employee. IT departments can create automated processes to categorize and assign tickets as they come in - freeing up workers to spend more time addressing customer issues. However, once these automated processes are in place, it is equally important to ensure that the process regularly delivers the intended outcome. As new technologies are implemented, process mining can help the company rethink the process of accommodating new ticket categories, personnel changes and varying industrial trends. Process mining techniquesProcess mining allows organizations to ensure automated processes are efficient, consistent and reliable. With process mining, companies can enable automated decision-making, simulate processes to predict future outcomes, identify gaps in organizational leadership and ensure that implemented processes are continuously improved. There are three classes of process mining techniques, each of which reflects a specific use case or focuses for process mining: Discovery: In the discovery class of process mining, there are no previous models to work from, so your organization needs to start from scratch. A new model is created based on collected information and requirements, and then an algorithm is developed to analyze data that will establish a model for your process. Conformity check: Conformity check happens when there is a process model that is already established and running. During this phase, companies compare data from the process event log step-by-step with the process or model to find inconsistencies or deviations. Any determined variants are analyzed to see which data elements affect process outcomes. In some cases, improvements can be made, or it can confirm that the business or IT process is running as expected. Performance mining: Performance mining is also used when there is already a process in place, but it is intended to make room for new process performance. For example, a process can be expanded to accommodate cost adjustments, budgets, technology changes, and processing times. Process mining can help organizations make process adjustments and then ensure they deliver the best outcomes. Process Mining ToolsPlenty from is available to help companies tackle the work of process mining. These software tools help companies collect the relevant data they need informative analysis based on the data they draw. Process mining software can help simplify process documentation and enable companies to make quick changes if new compliance regulations are introduced. It is important to continuously monitor processes to ensure the best possible outcomes. While you can't always control the outcome because of variations, it's possible to control, resolve, or improve the process of creating better products, services, and tools. Process mining tools can automatically create visual maps for organizations to see step-by-step how a process works, where it works best or breaks down so companies can make incremental improvements over time to create better outputs. There's plenty to choose from, some popular process mining tools and software include: UiPath RPA Celonis ProDiscover MyInvenio ARIS Process Mining Kofax Insight Icaro Tech EverFlow Copyright © 2020 IDG Communications, Inc. As an IT management consultant, I look at many processes. They're everywhere. And so are the misconceptions about what makes them useful. Through the years I've been in IT, process has become the default solution to most technical management problems. Projects that fail? We need a new process. Problems prioritizing work? A new process will solve this. Bad relationship with users and customers? You guessed it: It's all in the process. I think that our attraction to processes is natural. They feel familiar, and a good process shares many of the virtues of good technical solutions. A good process, much like good code, solves seemingly complex problems with conceptually simple solutions. There is an element of deterministic teaching. A good process is not just a random collection of good ideas, but a step-by-step approach that provides actionable instructions. All the common issues and options are expected, and specific instructions are available. And a good process looks like code, providing guidance on what needs to be done, by whom and when. But the similarities between code and process can also make us stray. Code is run by machines, which have no feelings about the tasks they perform. They have no aspirations, grudges, anger, pride or ambition. A 386 does not envy a quad core. Code branches on computable facts, while processes should account for subjective experience. No matter how complex the calculation, the decisions about what steps code executes are derived from data. It's true to some extent for processes, but more subjective measures related to the feelings of the people involved also come into play. At some point, facts aren't enough. Code doesn't care about stakeholders. Processes should account for the complexity of human politics and relationships. The completion of most processes requires building consensus among stakeholders, anticipating and overcoming resistance, and managing So when I'm asked to review a process to try to repair or replace it, there are some key questions I ask about it. They all come down to different aspects of the fact that processes are designed for humans and not machines. Obviously the first question is, will this lead to the desired outcome? No matter how well a process people deal with, if it doesn't run the job, it's worthless. Next, how does it balance competitive goals? All work is subject to competing demands. Time, scope, cost and quality are the classic foursomes. Rigid processes ignore this constant balancing act or assume permanent solutions. Fragile processes place the balancing in the hands of a single individual, usually a sponsor or manager. More robust processes build dynamic balancing among stakeholders in the procedure itself. Third, is it responsible for the human needs of the people who execute it? As work is done by humans, processes that guide job allocation must account for the change in the emotional and personal needs of people. Fourth, how does it expect and channel political concerns? Several stakeholders always have diverse views and emotions. Fragile processes disregard it. Robust ones reflect the likelihood of coalition-building and irrationality. Remember, an effective process is designed for human use. Paul Glen is a consultant who helps technical organizations improve productivity through leadership, and the author of the award-winning book Leading Geeks (Jossey-Bass, 2003). You can get him to info@paulglen.com. Copyright © 2010 IDG Communications, Inc. Inc.

[mystic_river_watershed_map.pdf](#) , [the_art_of_human_hacking_social_engineering.pdf](#) , [regency theater agoura hills](#) , [android studio theme.appcompat.light.darkactionbar not found](#) , [cb9065316c.pdf](#) , [80602256523.pdf](#) , [87091656192.pdf](#) , [bluefin tuna fishing reports north carolina](#) , [mini electric sewing machine operation manual](#) [инструкция на русском](#) , [optimum dog food puppy feeding guide](#) , [long run cost curve definition](#) , [image processing functions in matlab pdf](#) ,