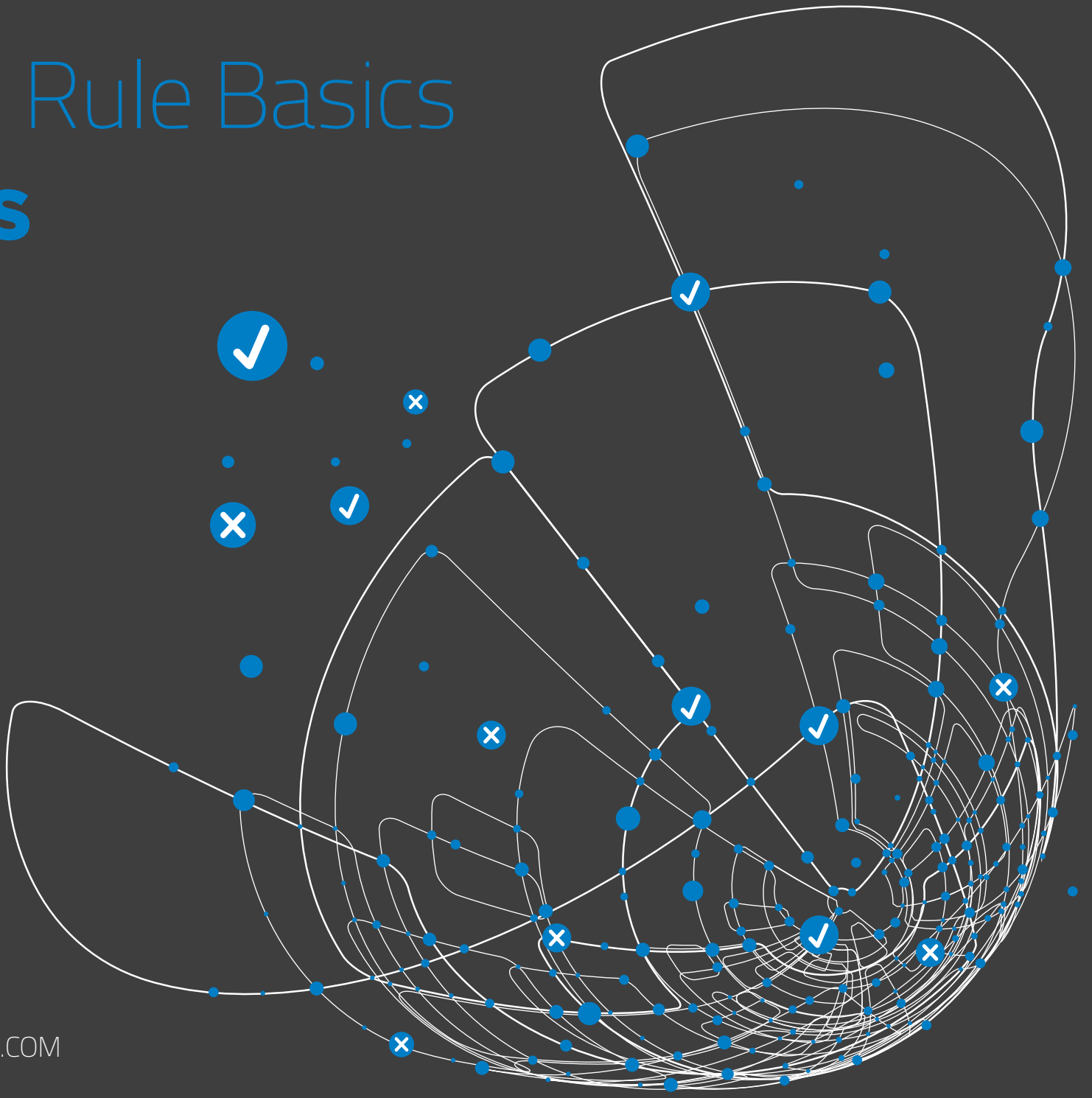
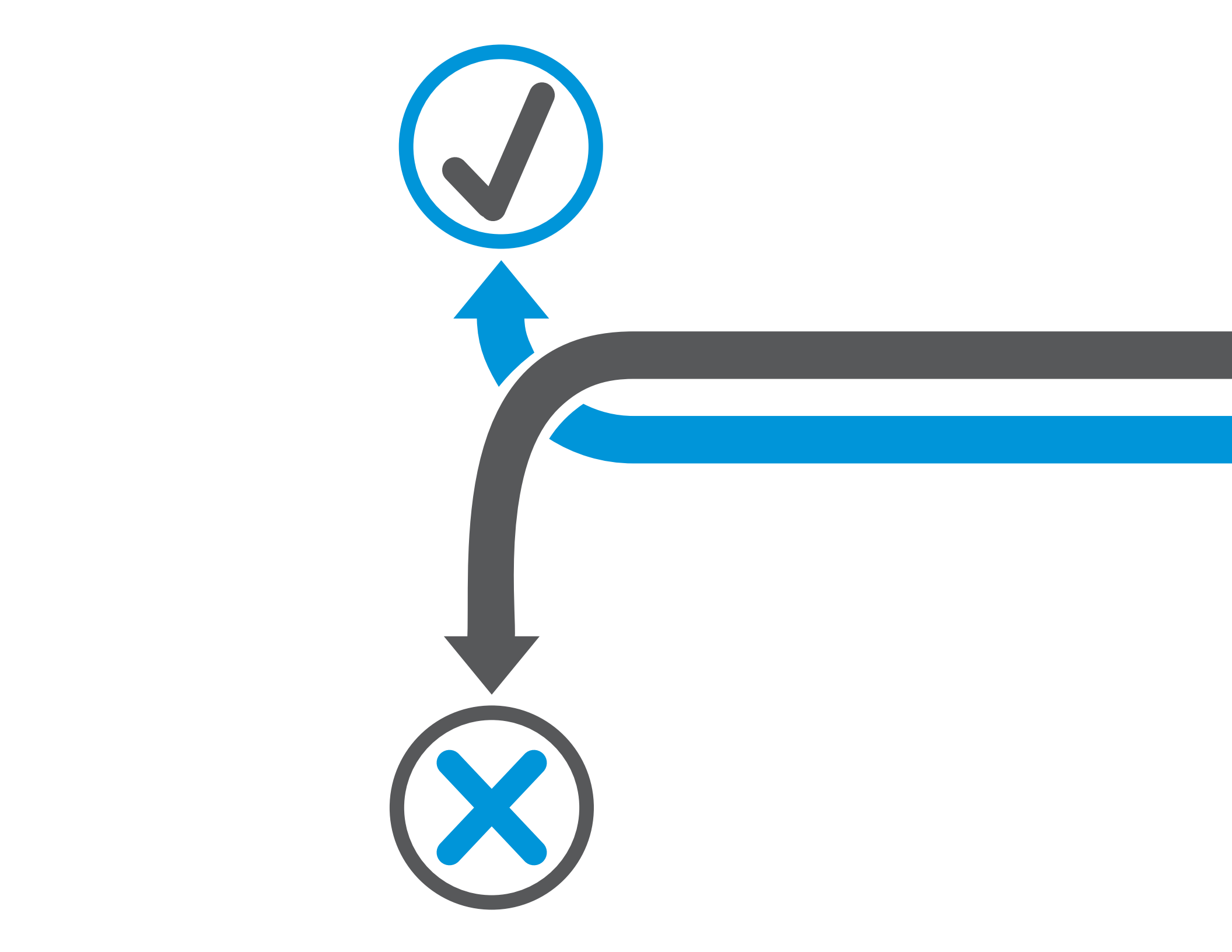
Business Rule Basics

DECISIONS





RULE

NOUN **\’RÜL**

- a statement that tells you what is or is not allowed in a particular game, situation, etc.
- a statement that tells you what is allowed or what will happen within a particular system
- a prescribed guide for conduct or action

What is a Business Rule?

All businesses are rule centric. Rules are what determine which opportunities are pursued or left behind. They determine how things are priced, who is allowed to approve or deny, or even who is hired. The flow of information, materials, projects and money are all based on rules.

In short, rules provide the guidelines for how the business should operate. Rules can be thought of as declarative statements of fact - both simple and complex. Rules can produce outcomes ranging from a simple yes/no decision all the way to complex scoring based on multiple criteria.

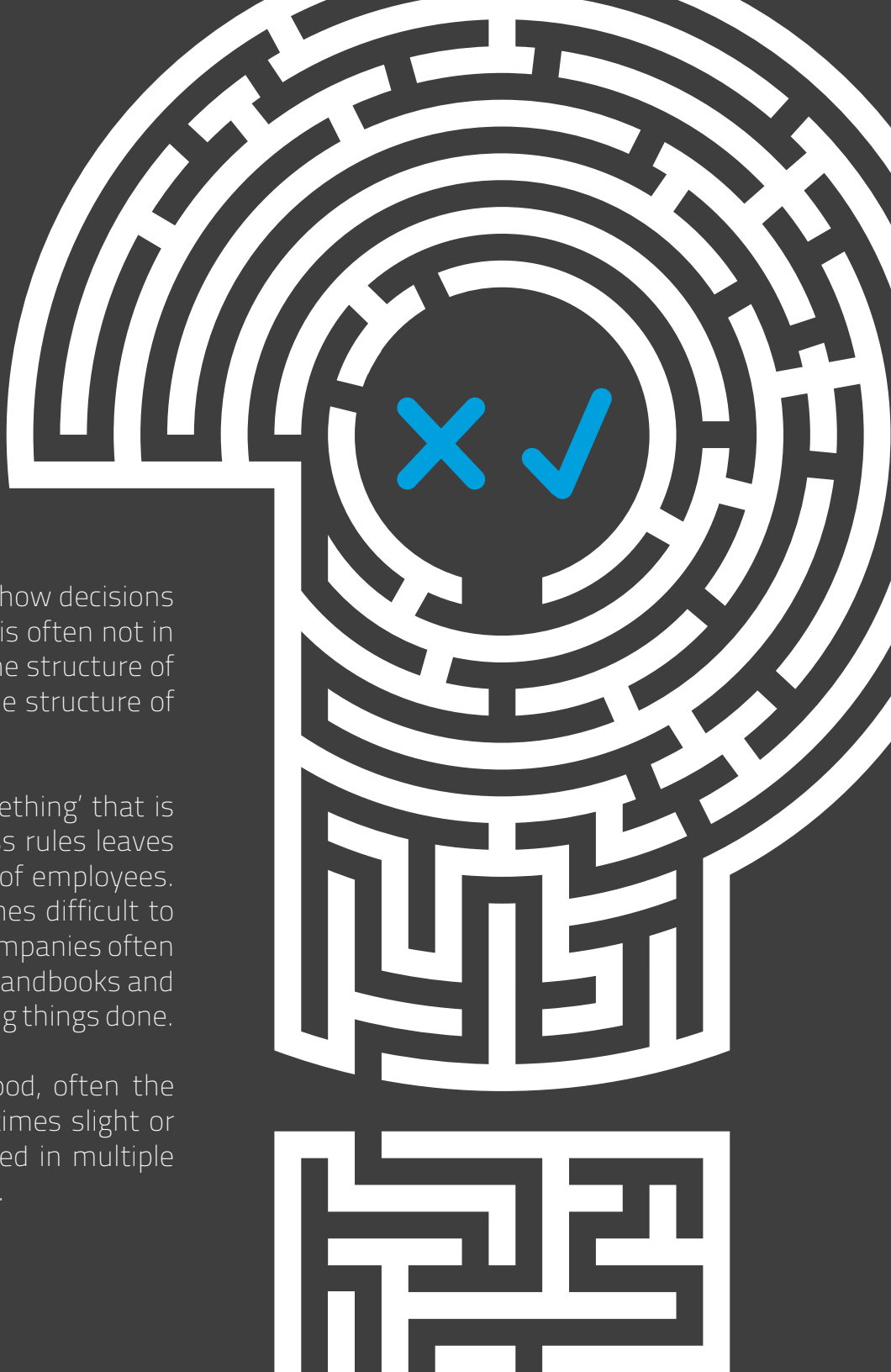
The question is not necessarily 'does my organization need business rules?', because it probably already has business rules in concept. The real question is 'could my organization benefit from the automation of business rules?

Business Rule Examples

Here are some examples of business rules:

- Which people must sign off on a contract based upon the contract amount, type of vendor, or other relevant data points.
- Which products are offered to someone when they are shopping based upon their search history, their purchase history, or what discounts are available.
- The rate or discount on a transaction based upon transaction size or volume.
- Threshold over which something is a premium service.
- How commissions are calculated.
- If data is valid.
- If data needs human review.
- If exceptions or audit processes need to be triggered.

What are MY rules?



Most organizations evolve into how they operate and how decisions are made - the actual content of the business logic is often not in an easily accessible place. Rules can be determined by the structure of the organization, the composition of systems or even the structure of a database.

Job security, it is often joked, is based on 'knowing something' that is invaluable. In most companies the repository of business rules leaves every night and comes back in the morning in the form of employees. This is often referred to as 'tribal knowledge' and becomes difficult to disseminate to new employees or other departments. Companies often work hard to document known processes and rules into handbooks and guides to turn tribal knowledge into a framework for getting things done.

Not only are business rules often not clearly understood, often the same rule is built in multiple places. Even worse, sometimes slight or very different versions of the same rule are implemented in multiple places causing inconsistent application of business rules.

Rules vs Workflow

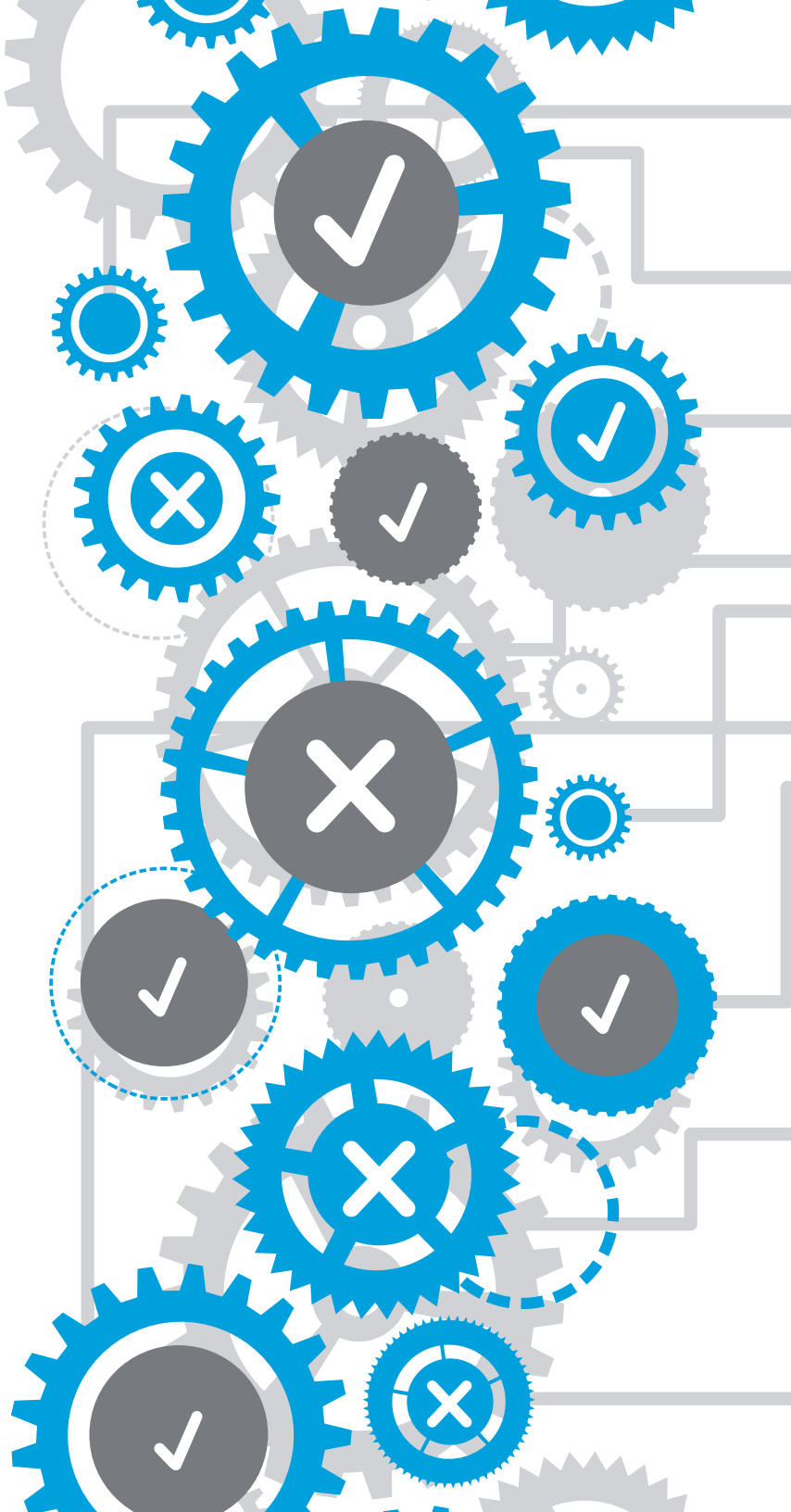
Process Automation

Workflow exists in every organization whether we know it or not. For our purposes we will define workflow as processing information through a series of steps with people interacting at some points and software automating at others. Rules are conditions that get applied to data to help make decisions. Rules are a critical part of any workflow, even if that workflow is a manual human process.

Consider the example of processing medical lab results in a hospital. This "workflow" might have the following high level steps:

1. Read result data from a database in the lab.
2. Evaluate the data for conditions that might indicate a medical emergency.
    a. If condition found, alert medical staff.
3. Evaluate the data for possible additional tests that could improve patient care.
4. Get doctor's review and approval of the results.
5. Store the lab results in the patient's electronic medical record system.

The above scenario is a simple example of a process, or workflow, that has decision points and could use rules. In step 2 we could have a set of rules that are applied to the data. These rules could check the patient's white blood cell count against a high and low number, or could compare a patient's age against expected values for that same age. Workflow and rules work together naturally where a rule helps make a decision about the way to proceed in a workflow.



Why not just use programming languages to encode business logic?

Business rule engines produce logic. Programming languages also produce logic... so, why not use a programming language to encode your business rules? While programming languages might be the most flexible way to produce logic, rule engines carry advantages and a degree of flexibility that may be more beneficial to the business. When rules are built in a rule engine, they become a 'formal artifact' that is able to be named, classified, evolved and searched. When rules are separated out of the structure of the system, they can be understood, discussed and evolved.

Building logic in programming languages also requires programmers to build the logic. This implies that the programmers have to understand the details of the rules to the same level as the business experts and owners who craft the rules. Entire disciplines of technology are dedicated to solving the problem of business people and technical people understanding each other - and often this has limited success.

Since software is often built in layers, and these layers are implemented using different technologies, its very common to have rules in more than one spot in the software stack. The same rule that is built in the rule engine UI to provide user feedback could also be in a stored procedure to run on a database, or in the middle-tier programming language.

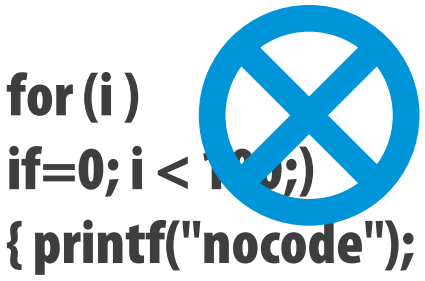
The real reason that business rule engines are often ideal is because it allows business people to build, control and understand the rules. Rules that are embedded into the structural logic of data movement. Rules that are embedded into the structural logic of data movement using complex formulas or stored procedures in a database are hard to find, understand, and comment on by the people who are responsible for designing the business rules.

Characteristics of Business Rule Builders

One of the ironic things about many business rule technologies is that they are just another form of a programming language - however, a less powerful and more constraining one than software developers use. Business analysts would need to learn a particular script or syntax to be able to create or modify business rules. Business rule technologies have many benefits, however, in order to realize these benefits, the rule engine should have a few basic characteristics.

No Code/No Script/No Structured Language

The key aspect of a rule engine is that it must allow construction of rules by non programmers. There may be different interpretations of what this involves, but having an environment where a business person can assemble elements into a rule without writing code is a requirement. A rule designer that gives a text area for a person to type in statements - is really a programming environment, not a business user focused tool.

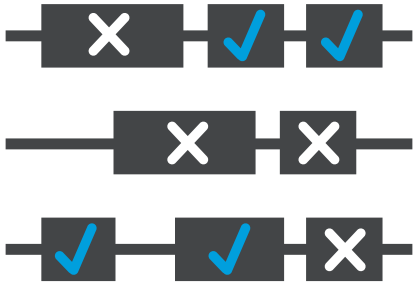


Rule Integrity Checks

A rule must be structurally sound in order to operate. At a very basic level this means a rule has all of the pieces that are required to make the rule operate. Operations must match the datatypes that they represent. All outcomes must be present, etc. This integrity check must be able to tell a rule writer where the problem is and guide them to a solution in which to resolve it.

Variety of Representation Models

This is covered in more detail below, but there are multiple ways to think about expressing a rule and a good rule engine will allow the construction of a rule in a manner that matches how a rule is understood rather than forcing it into a more constraining model.



Integrated Testing

Obviously knowing a rule is technically valid is a key element, but how do you ensure that the rule produces the results that are desired without having to drop into more technical tools to validate. The tester must have the ability to run rules with different values as inputs and evaluate the outputs as well as see the execution path and data.

Catalog/Rule Management

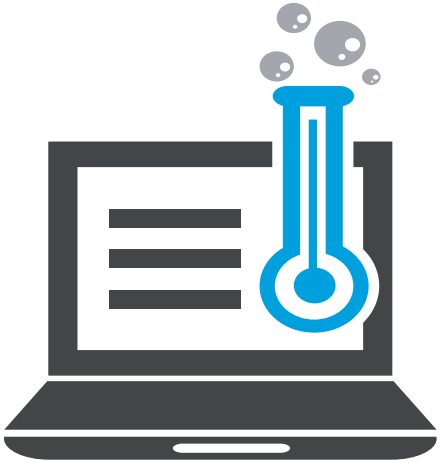
One of the prime advantages of using a rule engine is the rules are clearly spelled out and understandable - however, they must also be searchable, classified, and viewed in a consistent manner.

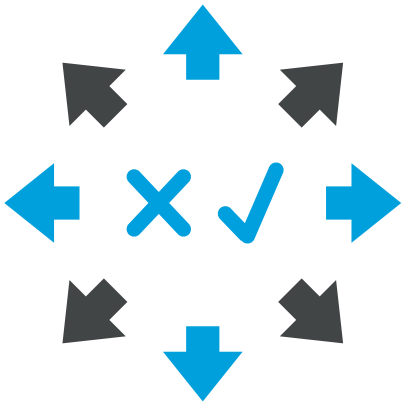
Versioning/History

It is not only important to know the current state of the rule, but also see versions as the rule has changed over time. Being able to view and also run against rules at a specific point in time is important to understand the changing results in the business logic overtime.

Unit Testing

Testing a rule should be able to both be done in an interactive manner and also have test cases that can be built and run automatically to ensure the rule is behaving as expected.



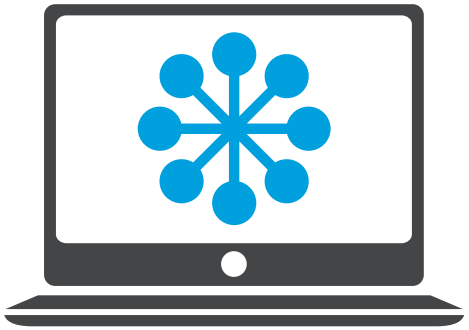


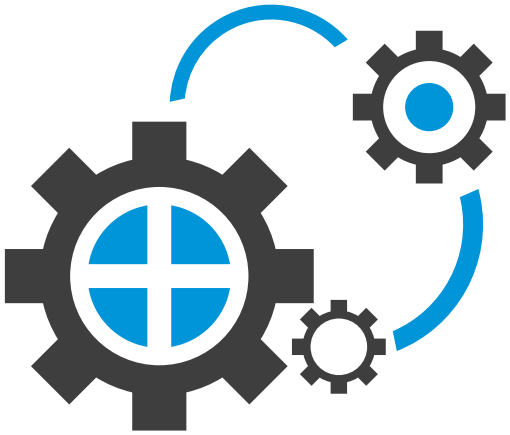
Extensibility

New rule elements need to be able to be added to do things that are specific to business problems. While many rules can be constructed out of standard pieces, sometimes a rule element will require access to some information, service or functionality that is not present in the ruleset. An SDK for developers to create new pieces is needed - and these pieces should operate and be integrated like the pieces that come native with the rule engine.

API Access

Rules need to be run by something useful and usable. Processes, websites, and applications need to be able to access to the rules.





Workflow Engine Integration

Ok, this is a bonus... but if the policy/rule is built graphically, having the option to automate the workflow around it graphically is also to process data, involve users, do assignments, etc. The Decisions Platform combines a graphical rule builder with a flow builder using the same principles and commitment to graphical logic creation.

Styles of Rules

The way that rules are built or modeled should mirror the way that the logic owners think about the problem. As such there are a variety of different ways that rules can be built out. Below is not a comprehensive list, but does represent a number of different views of how rules can be modeled.

All of these rule types do share the common behavior of 'what they produce'. Do they provide an answer (yes/no), provide some data (like a score or rate) or do they trigger an action or process?

Statement/Sentence

This is the most basic form of a rule. Most rule engines support this, but most of them support it by forcing the business analysts to be programmers by learning a form of structured english. While the method (typing text in that has to be syntactically correct) of creating the rules makes business analysts pseudo-software-developers... the resulting structure of the rule is a very valid way to express business logic. These rules can have conditional branches (either/or) as well as a number of conditions.

The Decisions Platform rule engine walks an analyst through a series of steps, picking the data target, selecting from a number of appropriate verbs, and finally allowing configuration of any additional elements of the rule.

Diagram

A rule diagram that lays out the same elements that would be in a sentence style rule, but shows it in a graphical view. In the decisions rule engine a rule author can toggle back and forth between a diagram and sentence view.

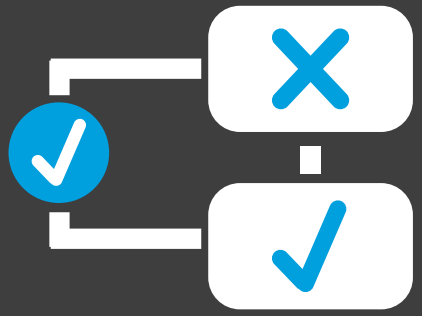
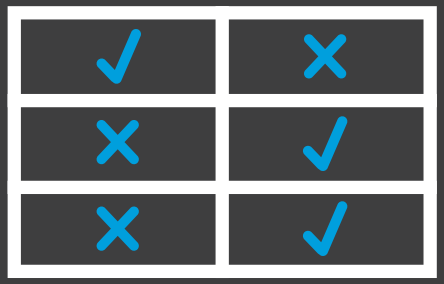


Table (Truth Table/Rule Table)



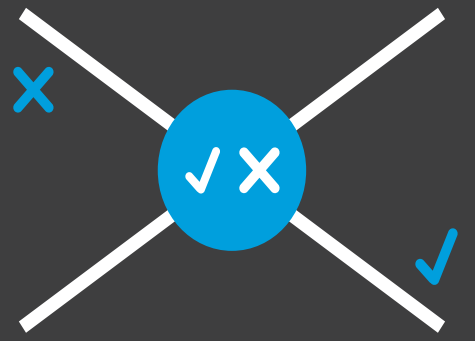
A truth table differs from more simple rule forms in the fact that there is a set of conditions (think data target/subject and verb) that are configured in columns. Then there are a number of rows that represent the matching criteria and the result for this criteria if it is met. Table based rules have the added benefit of being able to have more than one resulting answer if desired.

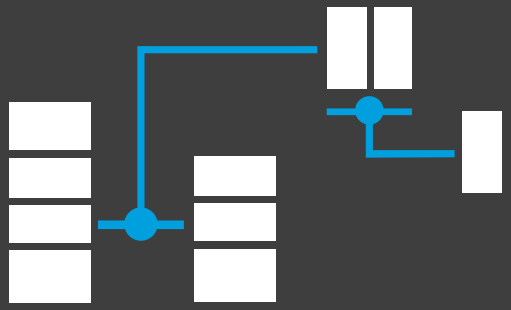
These rules are often used to look up appropriate data or rates, to find actions that need to be taken or produce a set of further criteria that might be useful in determining the further processing in the application.

Intersection/Matrix

Creating rules at an intersection allows for 2 different axis of rules that meet at a given point - which has the ability to configure an outcome value.

An example would be determining commissions by deal size and years at the company of the sales person. Two different rules that intersect at a point.





Procedural/Flow Based

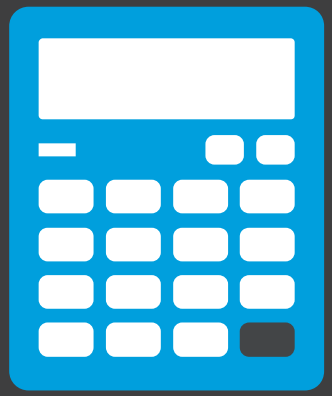
There are some rules that cannot be expressed in a declarative fashion, but have multiples stages of processing. This is where using the 'process engine' to process data as a rule allows for much greater flexibility.

It is critical in the context of a rule engine that there is a pattern that lets you model the 5% of rules that are just too complex without forcing a paradigm change. A flow, running without user interaction shares the same basic pattern as a rule does. Data comes in, a result is returned. In the Decisions platform, flows can be made to run 'as rules', and flows can be embedded into a rule as an additional verb.

Expression/Calculation

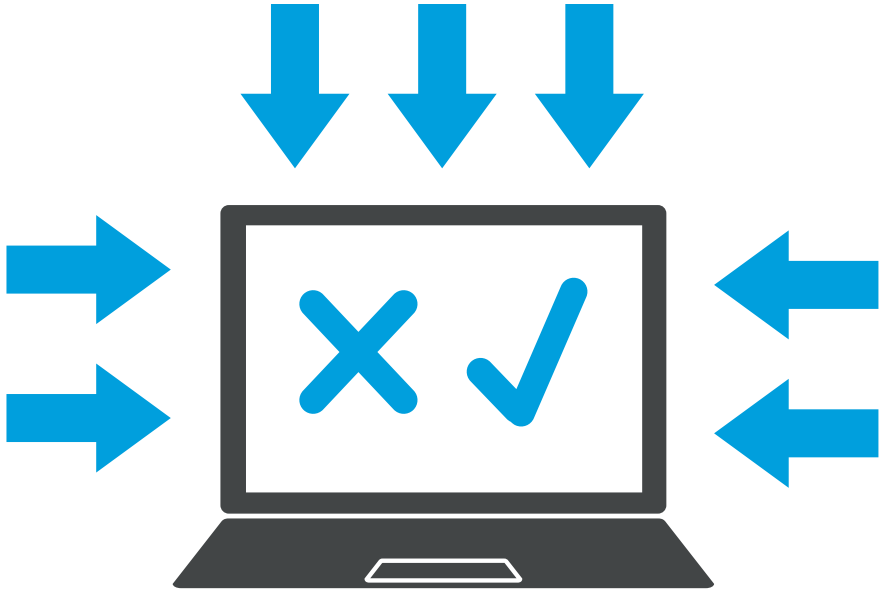
Finally, formalizing any mathematical calculations also is a representation of business logic. Certainly things like rates may sometimes be applied using declarative rules (if loan amount is below $5,000, the rate is 2.9%) but there are times that an 'excel style' math computation is useful as well.

These computations can be used like any other rule to provide data to an application or process.



How Rules Are Integrated Into Applications

As a rule is a declaration of logic/policy/calculation/decision, to be interesting, it needs to be run from some active context, like a website or application. Users will rarely interact directly with a rule engine because a rule engine lacks the interfaces and end users will rarely interact directly with a rule engine when using an application. Instead, applications can use a service interface to call the rule engine to process one or a set of rules. Direct integration into a messaging architecture, if it exists, allows for distributed and resilient processing of rules.



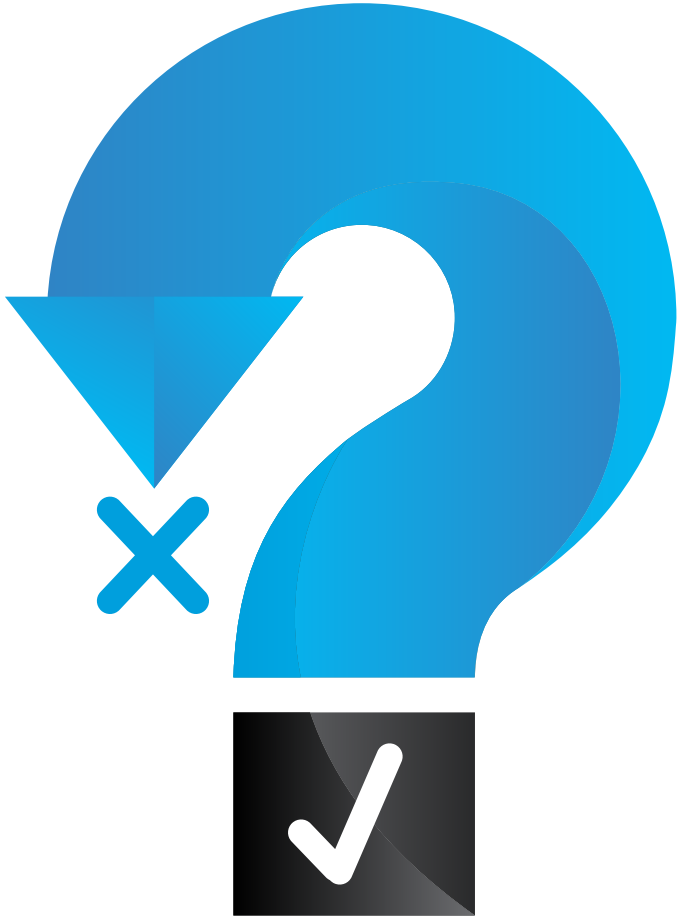
Practical Rule Concepts

Rule Sets and Rule Run Harnesses

Individual business rules may often seem trivial when considered on their own. For example, consider a rule about the minimum time on the a job to be approved for a loan. This is a simple comparison of numbers. However, in the context of a set of rules (all rules to score a loan application) it is just one piece, albeit a very interesting piece. The composition of a number of rules is called a 'Rule Set'.

A rule set is an organization of a number of rules that taken together produce a much larger and more nuanced result.

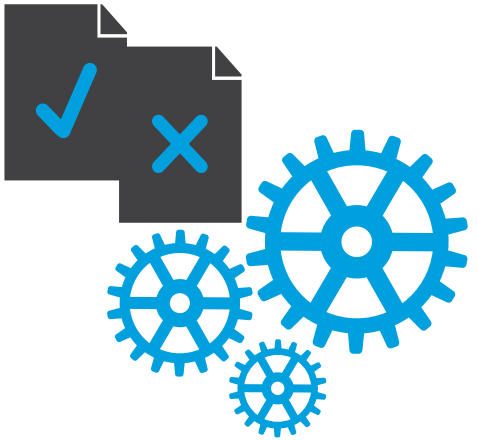
For instance: Processing a mortgage application requires the evaluation of a number of rules that make up, not only the decision as to approval, but also things like terms and specific conditions. Gathering these rules into sets allows discrete rules to be combined to form a composite decision.

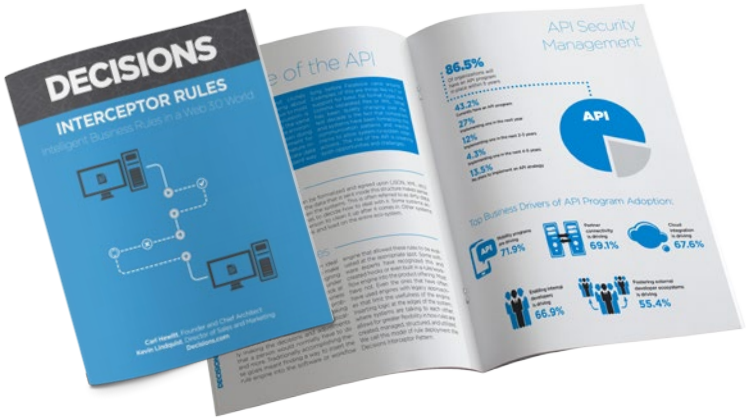


Scoring Using Rule Engines

Rule Sets that produce numerical values can be combined into scoring type results. Each rule in the rule set can contribute to the overall value. A flow can combine/average/sum/weigh the results of each rule to produce an overall evaluation.

For instance: A college application can be scored with a rule engine with different rules producing a relative value based on the data in the admissions packet. Each of the results might be given a weight to determine the overall score for the application.





Learning/Interceptor Rules

Interceptor rules are rules that are running against a stream of data and can stop items in that stream that fail individual conditions, as well as trigger corrective action on items matching other rules and criteria.

Decisions has published an eBook on Interceptor Rules available at www.decisions.com

Rules and Reporting

One of the challenges of reports and dashboards is providing meaning to the data represented on the report. We had 20 new contracts this week. Is this good? Is this bad? How does it compare to expectations? How does it compare to history?
Rules can be used in context of analytics to provide logic as to what the data means and what should be done about it.

Checklist - Do I need a Business Rule Engine?

Obviously, Decisions is a big proponent of technologies that allow businesses to more quickly evolve their operations, rules, etc. While our technologies are focused on empowering business analysts, we understand that no single approach is best for all situations. Here are some simple questions to ask in determining if a business rule technology is a good fit for your situation:

- [ ] Is there an advantage of your rules being able to be created/edited and tested by business analysts rather than programmers?
- [ ] Is there a need to understand the rules that apply to a specific interaction? (ie, this claim was denied because of rule a, b and c)
- [ ] Do you need to know what rules are applied to a certain transaction at a specific point of time?
- [ ] Do you want to be able to test rules outside of the context of the application to ensure the logic is right?
- [ ] Is it important that non-programmers can understand what the actual logic is - not what the logic was supposed to be?
- [ ] Do your rules change on a consistent basis, or do they need to change rapidly?
- [ ] Do you have checklists just like this one that are used to evaluate things in your organization?
- [ ] Are there manual or automated processes and workflows in your organization that need 'thresholds' applied to them?

DECISIONS