WHY DECISIONS ?

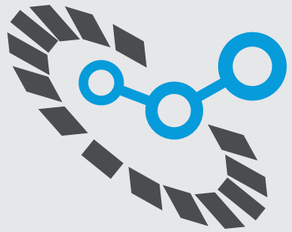
Our Guiding Principles

Why Decisions?

All software products accomplish something or solve some problem: The “What.” What does the product do? What features, limitations does it have? Can I use it to solve my needs…

Equally important is this question: “What is the underlying thinking that caused this product to be built?” Sometimes this question isn’t critical, but with business process tools and logic platforms for moving functionality to the browser and to the cloud – it is of critical importance.

Below you will find the primary reasons why we have built Decisions, Our Guiding Principles. These principles are the reasons that the product contains the tools and features that it has, and will also give you an idea of the types of things that you can expect to come from the platform in future iterations. When setting out to build our platform and portal, we did not gather feature sets from competitors and try to improve on them, instead we contemplated how applications might be constructed in the future.



Business Problems are always changing.

Business Processes are a Moving Target.

Applications must be built in such a way that they anticipate change – wholesale change. Applications and processes should be flexible and easy to change in order to accommodate this. Business have to change to compete, so business rules have to change. New processes need to be added. Existing processes need to be modified. Data needs to be able to be viewed through multiple lenses. New user roles and functions need to be added. Existing roles need to be merged or eliminated. We've been though this just as you have.

Applications that are unable to respond and change as the business needs change are more likely to hinder rather than help a business compete.

Given this: Decisions belongs to a new generation of tools for business optimization. Tools that can change at the speed of business need to be the foundation for business.

01

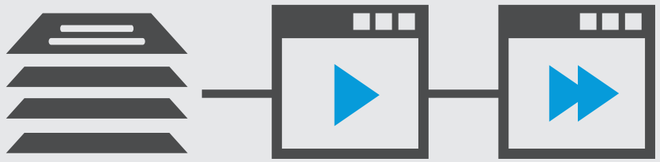
02

Tools for **Business Optimization** should be usable by Business Analysts.

Getting the people who know your business closer to your software processes means that your processes will know your business. Decisions believes that developers and business people can work together in a process optimization tool more effectively. For the business user this means, at very minimum – NO CODE. Yes, a lot of tools claim to be codeless, but often this means that there are scripts, configuration files, structured English, business domain language, and other facades that merely redefine what code is. It's 'code for business people' instead of 'code for software developers'. To be clear: if in order to perform a step in a workflow/execute part of a rule/get data for a report, there is a window where instructions need to be typed (SQL/JavaScript/Structured English, etc) – that is coding. However, a codeless set of tools and designers is only the beginning. These tools must be able to be understood and usable by non software developers. There are many aspects of this, but some of them are:

- Designers are visual and built with a maximum amount of feedback and assistance. This includes attributes like: drag and drop components, the visual assembly of items, immediate visual feedback of what is missing or invalid and more.
- Testing or a preview of the results needs to be fully integrated and a click away. The ability to see how a rule runs, what steps a flow will take given data and interaction, how a form will respond to a user, what a report looks like is critical.
- Deployment needs to be automated or automatic. Its not good enough that I can build something if I then need to involve more technical people to run it.
- Management should be fully integrated. Things like backups, change notes, versions, and comparison tools are important to provide a level of visibility and safety for business analysts designing the process.

Our thinking is inspired by more mature industries. The examples are everywhere, most of us feel very comfortable 'assembling' things that are well designed, even though the individual components may be black boxes themselves. Consider the following example: Installing a gaming system, dvd player or tv is able to be done by almost anyone that does not have a fear of electronics. Components that are a mystery as to how they work (how does that silver disk produce those sounds and sights) can be integrated by normal people. The plugs fit only in the right holes and there is clear feedback when things are and are not working.

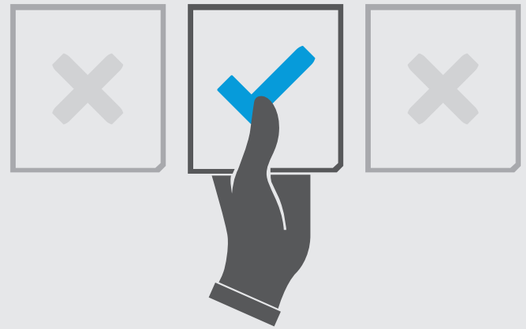


All data should be actionable

Traditionally there has been a separation between viewing data and being able to take action on it. Even environments that hook the 'analysis' and 'transactional' functions often require opening a complex transaction screen where the only action that can be taken is from the report context.

We believe that any report/dashboard/chart/data element should have a baseline set of capabilities that allow a drilling into the data that comprises the results – and allowing the user to take actions on that data (based on what they are allowed to do, of course). While large transaction screens are sometimes appropriate (example: open a contact in a CRM system) often they provide a degree of overkill to what is actually needed (ex. send an email about this contact or mark contact as inactive).

Beyond dashboards, when data is searched for or viewed in any context – if its identity is known – the actions can be present. Workflow and Reporting/Dashboarding are complementary. Workflow provides ability to initiate actions based on context. Reports provide a workflow process its context – data/entities/transactions.

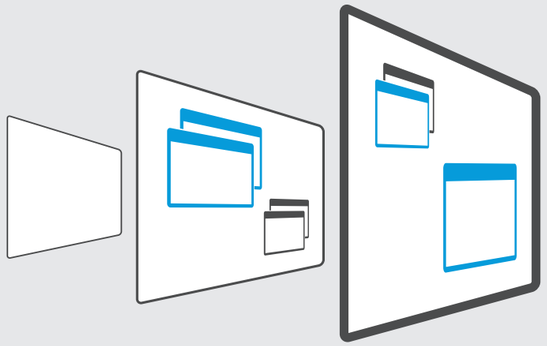


Functionality should be **a la carte** – use only what you need

To enable applications to be built with fundamental flexibility – there are a number of potential areas that could be addressed within a given application and problem domain. Often there is not flexibility to start building an application from scratch – an augmentation of a specific function is all that is desired or within reach.

The tools and designers in the Decisions Platform are architected in such a way that they can be both a foundation of an application and a contributor to an application or process. All elements, need to be able to be used independently of the other elements (forms outside the portal, rules outside flows, dashboards embedded in other portal technologies, flows without user interactions) as well as be able to be combined as appropriate.

While all of the elements are useful in some business applications and problem domains – they might not be useful in your problem domain or business. The underlying structure of these tools needs to account for this.



Our UI, Your UI or NO UI

In some cases – especially in workflow process automation – allowing the analyst to create user interactions is important. Additionally, some organizations or applications require overall management of basic elements of a user interface like user authentication, hosting of dashboards, providing a service catalog for user requests, etc. These elements while useful are supporting elements to the underlying designers.

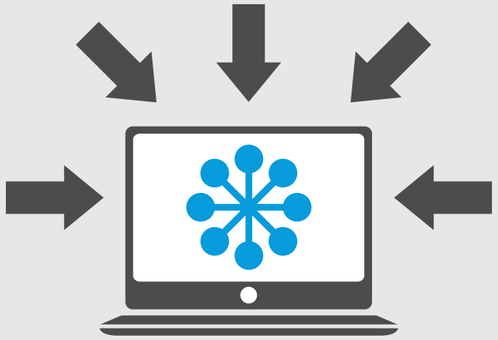
The presentation of the user experience is often dictated by a number of factors including: how do the users want to interact with the application, what other software is involved, the different types of users and the technology they have available and more.

The decision to use or not use user interface components provided by the optimization tool should not force a compromise in terms of the value that the tooling provides.

Users should have a good experience

User experience is very important. Whether the user experience is provided by the design tools or written externally, the user experience elements need to be well thought out. User experience is no longer confined to things with a screen and/or a keyboard. Applications are now routinely using alternate paradigms to provide users functionality 'where they are'. These include but are not limited to: Telephony, Texting (SMS) notifications and response, email response, monitoring other systems, etc.

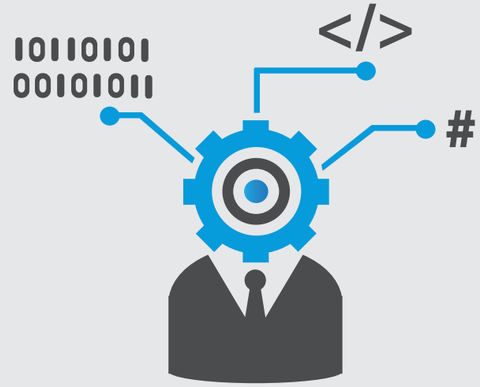
While these users interactions might need to be built outside of the designers, the users interactions produced by the designers need to provide a full featured and user friendly result – and support the different mediums that might be the preferred mechanism for a user to access the system. For instance, a workflow task might be responded to in a screen or by sending back a response to an email. A notification might go by text or email – or a popup if a user is logged in.



Integration is critical and it should not be hard

Workflow is glue. It ties people together by business process. It also must tie systems together. Integration to systems needs to be at the heart of business process optimization tooling - and not require a huge effort to do common integration. To as great extent as possible, the ability to integrate should be encapsulated and provided as elements that are prebuilt and pretested with the tool. When integration needs to be done with services/databases/ specific system API's tooling should provide the same level of help, validation and testing that are present in the designer tools.

When there is no out of the box implementation and integration to a specific system, then software developers are needed. There is full SDK and API that allow the basic elements (steps, rules, datasources, etc) to be extended in the platform. Developers can build and package up additional integrations that are 'native' to the tool and become additional reusable components.

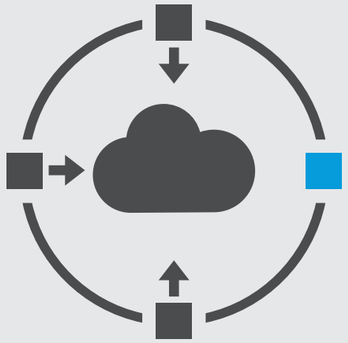


Developers are Welcome

Putting tools in the hands of business analysts is not an 'anti developer' move. It is allowing them to control their own logic and processes without involving software developers to make changes that could be done by the business. Developers are needed to both extend the capability of the platform and often build or integrate the designer capabilities with other software packages and applications. The SDK/API for the the design tools needs to be central to the technical architecture. All interesting elements need to be able to be extended or replaced using this SDK.
In the case of the Decisions Platform, developers can:

- Access all services via API's. (REST, WCF, Webservices)
- Add 3rd Party Libraries
- Extend Designers
- Flow Steps
- Rule Steps
- Data Sources (reporting)
- Filters (reporting)
- Form elements
- Dashboard Elements
- Data Structures
- Create New Business Services
- Add Actions to Existing Entities
- Leverage Decisions Framework
- ORM
- Validation Engine
- Aspect Oriented Programming Engine
- Configuration Storage

By providing tooling to the business analysts to make appropriate changes within the application, not only will those rules and process changes happen more quickly, development resources will be freed up to focus on other issues rather than spending so much effort making business logic changes.



Many applications/processes are migrating to the cloud, but some are not

Many applications are moving to the cloud, maybe even most applications are moving to the cloud – however, for a variety of reasons on premise software is often required or unavoidable. While the benefits of centralizing the management and administration of applications are clear – not all organizations or applications are there. Decisions was built for the cloud and it was built for on premise environments. The exact same software that runs in the cloud can be hosted on premise or in your own managed servers at other public cloud providers. The choice of where you run the workflow/rule engine is up to you.

Additionally, we recognize that even if you a run in the cloud, a safe and robust mechanism is needed to allow secured access to on premises resources. Very few organizations have all systems running in the cloud and accessible – even if this is their strategic direction.

Business Optimization is a journey

Business optimization tools are just that – tools. Businesses are complex and an ever changing landscape in which to operate – not to mention evolving systems and the cast of people that are involved means that workflow processes are ever changing.

These changes come both from the fact that the business is changing – but also from the observation and analysis of the results of prior changes. By observing the effect of rules and workflow adjustments additional adjustments become evident. The need for strong data analyses tools to facilitate these interactions is critical.

DECISIONS