Orchestrating work, intelligently

CASE MANAGEMENT 103

A PEGA
WHITEPAPER

PEGA

People – whether they're your customers or employees – expect seamless experiences and exceptional outcomes around the clock. One way to manage both expectations and workflows, is through case management: a software-based approach that connects people, processes, and technology to get work done rapidly and correctly, the first time – every time.

To enable the kind of customer journeys that lead to better business outcomes, case management blends centralized artificial intelligence (AI), hybrid robotic process automation (RPA), and low-code software development to fundamentally evolve workflows and engagement.

And when coupled with a Center-out™ approach to business architecture, your enterprise can shine. Business processes are centralized, regardless of customer channel or organizational complexity, so work doesn't get lost in the shuffle.

In **Case management 101** and **Case management 102**, we examined a high-level overview of case and how to assemble a team to ensure success. Now, let's look at how to orchestrate work.

Define case types

Now that you've assembled your team, it's time to get to work. First, you need to define your case types. A case type is essentially a category of work. Case types include the characteristics a set of cases share, including the data collected for them and steps required to get work done.

When users interact with a case management solution, every case they create is an instance of a defined case type. For example, you might define a case type called *Address Change*. Each time you process a new request to change a customer's address, it creates a new case of the case type, *Address Change*.

Sophisticated case systems apply an object-oriented hierarchy, where case types can be children of other case types. *Address Change*, for instance, may be a child of a larger *Customer Service* parent case.

When defining case types, ask:

- How would customers describe the work they want done on their behalf?
- How would the business describe the desired outcomes for these types of work?

You may need to ask several people for answers to the second question, including users of the application, managers, and maybe even senior executives. When you ask managers and executives about the business outcomes they're trying to achieve, their answers should be a pretty good indicator of the reports, metrics, and KPIs you should bake into your solution.

Manage complexity

As we mentioned earlier, you may have parent and child cases. Defining these cases allows you to expand and adapt to customer and agent use cases across the enterprise. The result? Teams can manage work while still delivering a cohesive, friction-free experience.

Creating child cases is one way case management helps insulate your customer from the work silos in your organization. Organizations develop silos for a reason; silos allow a company to manage functional areas and align employees who are working on related tasks. However, this approach becomes a problem when an enterprise with multiple business lines and geographies tries to deliver a personalized, frictionless service everywhere. Employees struggle to navigate silos, often dealing with multiple systems and applications. And customers end up with disconnected, dissatisfying experiences.

Parent and child case relationships also make parallel work easy to manage. A parent case may spin off multiple child cases. Each of those child cases – transaction payments, fraud investigation, and so on – can be processed in parallel, with the parent case managing the relationships among the child cases.

Part of the definition of a case type includes the definition of the child cases it could contain. If a case type *can* contain child cases, it becomes a parent to those cases. Additionally, the case type should define when (and whether) that child case should be attached to the parent. Some child cases are created automatically at the start of the parent case or when the parent case crosses a particular milestone. Other child cases are defined as optional and are added to the parent only under specific conditions, or when requested by a stakeholder or user.

Define stages

After case types are defined and parent/child relationships are slotted into a case hierarchy, you need to define how work in each case type will be completed. Begin by defining the stages that outline the journey – from starting a piece of work to completing it. Stages represent the high-level flow a case goes through.

It's important to remember that stages aren't child cases – they don't represent meaningful business outcomes on their own. Stages aren't tasks either, because they aren't individual steps. If you've ever drawn a series of steps on a whiteboard, you've probably defined a series of stages.

Using stages to define the high-level flow of a case can be a lot easier than defining every detailed step that makes up a business process. It's easier to get stakeholders to agree on stages than low-level steps, which have levels of detail and variation that are challenging to capture. For this reason, capturing case stages is a great way to start defining your case management application.

Define stages so that their meaning is apparent to all stakeholders in the case – including the customer. Stages become the way to communicate about the status of the cases being managed.

Map alternative stages

Cases usually flow linearly from one stage to the next. But what if a customer calls to cancel an order? At this point, the case needs to drop out of the normal flow of stages and into an alternative stage. Alternative stages handle exceptions in case processing. Tasks responsible for cleaning up canceled or restarted processes often live within the alternative stages of a case.

Mark milestones

The markers between stages are called milestones. Milestones are the gates that a case must pass through to move from stage to stage. Defining the milestones at the beginning or end of each stage is essential to tracking and monitoring the progress of work.

Establish service level agreements

Each stage should also have a service level agreement (SLA) associated with it. SLAs define the time frame that stages should be completed in and are often expressed in the form of a goal (when you want to have this stage completed by) or deadline (when you need to have this stage completed by). Some applications define both a *Goal* and a *Deadline* stage for each service, increasing the level of escalation as each goal or deadline is met.

SLAs are usually defined in one of two ways:

- **Explicit date:** In a case type for processing a mortgage application, for example, a closing date is usually defined. Each stage of the mortgage application has a service level that's defined relative to the desired closing date.
- **Number of days from the start of each stage:** A customer service case, for example, may demand that the *Fulfillment* stage be completed within three days of starting that stage.

You should define service levels not only for each stage, but for the overall case type as well. Make sure that the service levels for the stages roll into the overall case-type service levels.

Service levels should also define actions. When the goal or deadline for a service level is crossed, the case and its tasks should be escalated. Escalation could take the form of increasing the urgency of the case or triggering escalation actions, such as sending an email or text, or transferring the case to a supervisor.

Plan for resolution

The last stage of every case type is *Resolution*, which represents the completion of a piece of work. Most cases get to the *Resolution* stage and stay there indefinitely. At this point, the case, its data, and its processing history become vital pieces of information for reporting on and improving your business. Looking at resolved pieces of work is a great way to understand your stages and data. Because resolved work has already passed through all the stages, you can find evidence of the stages in the resolved work and use that evidence to define the stages for your case type.

Think about reuse

Keep reuse in mind as you define case types, parent/child relationships, and case stages. You may eventually expand your solution beyond its initial project to other pieces of work, so you want to find ways to reuse the case types and stages you've already defined. Reuse is important for two reasons:

- **Improved time to market:** If you've already defined a case type that you can reuse in multiple places, reusing that case type means you don't have to rebuild it a second time (or a third or fourth time).
- **Consistency:** Your legal department may have a way of handling fraud investigations that it wants every department and line of business to conform to. Creating a reusable case type called *Fraud* and defining that case type as a child case with many parents *(Account Opening, Claims*, and so on) ensures that all groups in your organization follow the same best practices for investigating fraud.

You can reuse much more than just your case types and stages, of course. In fact, as you think about your solution, you may find many opportunities – especially when you think about the tasks and processes that define how a piece of work is to be completed. The right low-code platform also makes managing reuse easy, allowing you to use a variety of common components across solutions.

Design by doing

While the creation of new ad hoc work is a given with case management, no enterprise wants to reinvent the wheel. The goal is to standardize as much as possible. *Design by doing* describes the case application's ability to define new case types based on completed ad hoc work. It creates the model and structure for something based on what was actually done.

A new piece of ad hoc work is only new once. The next time that situation is encountered, regardless of who encounters it, whatever was learned in the first experience will be applied to the second to ensure consistency and begin building best practices.

Design by doing allows end users with permissions to take ad hoc work they've done and save it as a template that can be reused by anyone across the enterprise. This approach provides a kind of controlled crowdsourcing – allowing you to tap into the expertise of your people without waiting for IT to approve of and deploy the changes.

With case handling practices constantly evolving, and new exceptions constantly cropping up, this kind of hyper-adaptability is necessary to support case management applications.

Start automating intelligently

We've looked at how to assemble your team and organize your work – all to get work done. We also briefly examined the types of work case management can help you track when it's too unpredictable to guide or automate.

But what about the cases that you can automate? By using AI, RPA, and low-code software development, case management can help you empower business users and IT to automate, develop applications, and deliver results – fast.

Automation also plays a huge role in the future of work. Despite worries that automation will reduce or eliminate jobs, it actually creates jobs, frees up resources, and improves worker satisfaction. In fact, 81% of enterprises we surveyed say intelligent automation saves each person four hours per week.[1]

Learn more about case management, the **Center-out™** approach to business architecture, and how intelligent automation can help your enterprise ride the third wave of digital transformation.

Visit us at **pega.com** to learn more.

1. The Future of Work: New perspectives on disruption & transformation (Pega, 2020)

PEGA

Pega delivers innovative software that crushes business complexity. From maximizing customer lifetime value to streamlining service to boosting efficiency, we help the world's leading brands solve problems fast and transform for tomorrow. Pega clients make better decisions and get work done with real-time AI and intelligent automation. And, since 1983, we've built our scalable architecture and low-code platform to stay ahead of rapid change. Our solutions save people time, so our clients' employees and customers can get back to what matters most.

For more information, please visit us at **www.pega.com**