Exceptions are a painful reality for most companies. Missing customer information, improperly coded order fields, incorrectly allocated inventory, and data mismatches between customer and billing systems are just a few examples of exceptions that can happen in the typical course of business. When these exceptions divert business transactions from the normal process path, the results are often labor-intensive offline resolution, long delays, inaccuracies and customer dissatisfaction. Everything from order fulfillment to compliance can be impacted when an exception slows or halts a mission critical process. Exceptions are also a costly reality, and can easily account for over 50% of process-related costs. Worldwide, spending on exceptions is estimated to exceed $50 billion per year. Resolving exceptions faster and more accurately can significantly benefit both the revenue and cost sides of a business.

Most companies today are ill-prepared to manage and resolve exceptions. Exception management is an afterthought to most IT projects, with little attention and few resources being applied to the problem during design and implementation. The problem typically surfaces shortly after the IT project goes into production. A standard solution to the exception management problem is to “throw people” at the problem, often with little training and with little system support other than a bare-bones case management system. Nonetheless, the applied knowledge workers are expected to learn how to identify and resolve exceptions over time. While such an ad hoc, labor intensive approach may be a pragmatic approach to exception resolution in the early stage of an IT initiative, it is not a cost-effective long term solution. Moreover, as the business grows, both the frequency of exceptions and the cost of their resolution tend to grow even faster.

Recently, a new breed of software application called a Business Exception Management System (“BEMS”) has appeared to help conquer the exception management problem. In contrast to today’s ad hoc approaches, a BEMS brings a more structured approach to exception management, allowing best practices to be applied in a consistent fashion. A BEMS supports an incremental and evolutionary approach to automating exception management steps, from exception identification to exception resolution. In addition to producing dramatic cost savings, a BEMS can also bring significant additional benefits through better work force management and improved customer satisfaction and retention.

The remainder of this article will cover in more detail the state of exception management today within most companies, and the resulting pitfalls. It will describe the properties and capabilities of a best-in-class Business Exception Management System. The article concludes with the business benefits of using a best-in-class BEMS.

**State of Exception Management Today**

**Labor intensive**

Often exception resolution relies on what is often referred to in the industry as “the heroic efforts of knowledgeable and experienced workers.” By their very nature, these heroic efforts are labor intensive, resulting in costs and delays.
Ad-hoc exception classification and resolution
Unfortunately, these heroic efforts tend to be ad-hoc across workers and situations. Different workers, even when presented with the same exception under similar circumstances, will tend to approach the resolution differently based on their own individual history of dealing with such exceptions.

Unpredictable results
The ad-hoc nature of exception resolution means that results and performance are inherently unpredictable. Different workers take different exploratory paths toward resolution, resulting in varying times of resolution and perhaps even different outcomes based on the path taken. Even the classification of the exception type and severity will tend to be ad-hoc and unpredictable among different workers.

Little automated system support
Often, no automated support is available for the problem worker. At best, a case management system or workflow system may be used to track the status of the problem.

No systematic knowledge capture
As workers learn to resolve exceptions, the knowledge is captured in their heads and stays there. It becomes undocumented “tricks of the trade”. There is no way to capture the knowledge formally in order to pass it on to others or, better yet, support automated resolution. This undocumented resolution knowledge is always at risk of being lost if a worker leaves or transfers.

Contextual information is often missing
Contextual information is often needed to resolve an exception, and this information is often missing or hard to retrieve. For example, in an order processing exception, information about where the order is in its process flow and what data has been accessed is not passed along to the worker. Even the original order itself may not be made available to the worker. Consequently, the worker will typically need to access different systems to manually retrieve the relevant contextual information.

Lack of visibility into status and process
When exception classification and resolution is ad-hoc, there can be little visibility into what is happening at that time other than rudimentary status information. This is true even if a state-of-the-art workflow system is used.

Difficulty of prioritizing and balancing the exception workload
For example, in the order fulfillment process, one would like to prioritize order problems based on type of exception, order deadline, customer class, and other contextual information internal to the order process or in external systems. However, the overall lack of visibility within exception processing combined with ad hoc classification schemes and lack of contextual information makes it difficult to prioritize the exception workload in a meaningful way.

Lack of information for process improvement
Without a systematic way of classifying errors and then capturing both realtime and historical information on problem resolution, there can be no informational basis for improving the resolution process. Moreover, without a systematic method for capturing resolution knowledge, the knowledge gained by individual workers will be lost when workers transition to new roles.

Lack of scalability
Manually intensive resolution approaches are both expensive and difficult to scale. A major barrier to scaling is that the resolution work requires highly specialized and largely undocumented knowledge about systems and processes.
Best Practices and Best Features in Business Exception Management

Business Exception Management Systems are an emerging class of software applications specifically designed to manage and resolve business and process exceptions. The best-in-class products support the following practices and features.

Move the classification and resolution of Business Exceptions outside of the main flow of the business process

A business process typically has only a few “happy” paths, that is, normal and correct execution paths, but may have many error-prone paths that result in exceptions. The ratio of exception paths to happy paths can easily exceed 10 to 1. If exception management logic is encoded as part of the normal flow, it can overwhelm the normal logic, turning the resulting process flow models into a “spaghetti” wiring diagram.

Entangling the exception logic with the happy path logic results in process definitions that are large and hard to comprehend. The exception logic will tend to dominate the so-called “happy path” logic. Such entanglement also makes it very difficult to change the happy path logic.

Exception resolution flows will tend to change more rapidly and frequently than “happy path” flows, as ever more types of exceptions are identified and standardized resolution processes are defined. If exception resolution flows are part of the “happy path” flows, then each process change to assist in exception resolution can jeopardize the “happy path” flow. Minimally, the “happy path” flows will need to be subject to regression testing. Worse, new exception logic can introduce errors into the “happy path”.

Separation of “happy path” process logic from exception logic is a classic example of the design principle of “separation of concerns” leading to better overall comprehensibility, modularity, isolation of design and runtime errors, and support for evolutionary change. This separation provides the following benefits:

- It provides increased comprehensibility, evolutionary change, and error isolation.
- It makes it easier to modularize and re-use exception resolution solutions across multiple business processes

Provide explicit Lifecycle Management of the Exception Resolution

The processing of an exception typically follows a general pattern. First the exception must be detected, either by being explicitly raised by an application or system, or being implicitly detected through monitoring application behavior or logs. Once detected, the exception needs to be identified and classified by type so that an appropriate set of investigation and resolution actions can be taken. Once all resolution steps have been taken, the process or application that raised the exception must be notified as to the resolution outcome, whether successful, unsuccessful, inconclusive or escalated. This general pattern is called the Exception Lifecycle and it needs to be explicitly managed.

Lifecycle management provides the following benefits:

- By following a standardized exception processing pattern, it allows for a more structured overall approach and more predictable outcomes.
- It makes it easier to build automated tools for exception processing.
- It provides standard processing milestones where information can be gathered for reporting and analysis.
- It facilitates the crafting of methodologies for exception processing, the training of analysts to design resolution processes, the design of the resolution processes themselves, and the testing of the resolution processes.

Model-driven approach

A model-driven approach allows the knowledge about exceptions and their resolutions to be captured in tools, such as process models and business rules, that are both human readable and machine readable. Exception knowledge,
when captured as models, is easier to comprehend by analysts than when it is coded in a programming language. A model-driven approach allows for rapid development and, more importantly, for rapid evolutionary change.

A variety of model types can be used to capture the intricate knowledge of exception management, including,

- The sequence of steps to investigate and repair an exception can be captured in process models which depict both automated system interactions and human-oriented workflows.
- The classification of exceptions can be described using Business Rules.
- Exception definitions, error codes, and associated document definitions can be described in error dictionaries and data models.
- When incomplete or inaccurate document data is the root cause of an exception, data validation rules can be used as a basis to repair the document and hence resolve the exception.

Use a Repository-based approach

A key element of a model-driven approach is a Model Repository (also known as a metadata repository) for storing and managing the various modeling artifacts critical to exception processing, including data models, process models, and rules. The Model Repository should organize and relate the various models around the various exception types – associating with each exception type the definitions, models, and rules that are used to classify and resolve the exception.

The model repository should support both design time and runtime governance. During design time, it should support a rapid model development methodology with incremental revision, preferably through an explicit version tracking and management scheme. This governance should extend to runtime and support, both for planned promotion of models from design time to runtime, as well as for unplanned runtime model changes.

Leverage Business Process Management (BPM) and Business Rule Management (BRM) systems

The model-driven approach to exception management relies on a strong BPM capability. The resolution of an exception is typically a complex, multi-step process. Many exceptions will still require human intervention, and this is especially true in the early phase of an exception management project. As experience and knowledge of exception management increases, so should the level of automation within resolution processes, trending toward the overall objective of fully automated exception processing. Thus, the underlying BPM system needs to support not only both automated and manual process steps, but also the incremental and graceful evolution of processes from mostly manual to mostly automated. A BPM system with comprehensive and completely integrated workflow management is therefore required.

A Business Rules Management system is required to support a flexible exception classification system. It is also helpful in many other areas within the exception lifecycle, including data validation, skills-based workflow assignment, and guided manual exception investigation utilizing decision-trees. A general-purpose rules system that is fully integrated into the Exception Management System provides the maximum flexibility and ease of use.

Produce a complete Audit Trail of the Exception Lifecycle

Logging information at each step of an exception lifecycle, from identification and classification through resolution and on to final disposition, provides a wealth of information for both real-time and historical reporting and analysis. The realtime information can assist in load-balancing the current workload, while the historical information is useful for cost analysis and resource allocation. Capturing this information at consistent milestones in the exception lifecycle facilitates a comparison of different resolution approaches for the same exception. Historical information can be used to derive cost and resource profiles based on exception type, and these profiles can be used as a basis for prioritizing the current workload of exceptions and predicting their future resource requirements and cost. Also, historical information is useful for trend analysis on exception occurrence and can help identify exception patterns that may occur on a daily or monthly basis. Historical information, together with explicit knowledge acquired by workers, forms the basis for improving the resolution processes.
Benefits of a Structured Approach to Business Exception Management

Reduced Labor Costs
A reduction in labor costs is often the most immediately visible benefit of Business Exception Management. Experience from our customers demonstrates that a 60% reduction in the labor costs is achievable during the initial implementation of an Exception Management System. These cost reductions are expected to continue to grow during the lifetime of an Exception Management System as resolution processes are incrementally improved and fine-tuned to specific exception types.

Consistent classification and resolution of exceptions
The enforcement of consistency throughout the lifecycle of exceptions leads to more repeatable results and lays the foundation for analyzing and improving resolution processes. A consistent classification scheme is a prerequisite for prioritizing workloads.

Predictable results
With consistency in classification and resolution process come predictable result outcomes. Even exceptions that are resolved manually become more predictable because a consistent classification scheme leads to common vocabulary among knowledge workers, enabling them to share and exchange best practices. Common resolution metrics, including total resolution time and resources costs, also become more predictable, as these values converge toward quantifiable averages with narrower standard deviations.

Automated system support for all exceptions
Even for the remaining manual-intensive exceptions, a Business Exception Management System can offer significant support to workers responsible for resolving the exception. This includes automated classification, audit logging, automated retrieval of necessary contextual information, and guided manual resolution.

Systematic and incremental knowledge capture
One of the greatest benefits of model-driven resolution is that knowledge can be captured formally in models that are both human-readable and machine-readable. Exception knowledge now becomes generally available and easily shared. Moreover, a model-driven approach supported by a model repository allows knowledge to be captured incrementally and in pace with the experiential learning curve of the associated knowledge workers. Acquired knowledge can readily be transferred to automated processes via models.

Availability of contextual information to speed exception processing
Contextual information residing in applications and IT systems can often be automatically retrieved pre- and post-classification to speed resolution. In some cases, the additional contextual information may be sufficient to resolve the exception in a completely automated fashion. Otherwise, the additional contextual information can be presented to knowledge workers, thereby easing their tasks.

Visibility into status and process
A Business Exception Management System, supported by a full-featured BPM system, provides visibility into exception resolution throughout the lifecycle of an exception, with detailed visibility into the resolution process itself. This visibility is useful in managing the workload and the utilization of systems resources. Also, status information can be communicated to customers and other stakeholders affected by the exception.

Prioritized and balanced exception workloads
With the visibility provided by a BEMS, exceptions can be prioritized by type, deadline, customer, and similar contextual information. Prioritization routines can also utilize profile information collected by exception type and statistic information obtained from audit logs.
Incremental process improvement
A model-driven approach based on repositories provides the foundation for incremental process improvement. Business process models and business rules can be evolved as necessary and more rapidly than software code. Realtime and historical information can provide insight into the effectiveness of resolution processing and highlight opportunities for process improvement.

Scalability
Automated processes can be scaled faster and cheaper than manual workflows. Workers can also be brought up-to-speed more quickly since the exception knowledge is stored explicitly in models, which are shareable, rather than primarily in the heads of experience workers. Even the most labor-intensive exception work is facilitated with a BEMS.

Summary
Exceptions are a painful and costly reality for companies. However, a new breed of software application, called a Business Exception Management System (“BEMS”), is helping companies to conquer the exception management problem. A BEMS brings a more structured approach to exception management, allowing best practices to be applied in a consistent fashion. It supports an incremental and evolutionary approach to managing the entire exception lifecycle, from exception identification to classification and on to resolution. In addition to producing dramatic cost savings, a BEMS can bring significant additional benefits through better work force management and improved customer satisfaction and retention.