Insurance On Demand, the IBM Primer
Booklet Three

IBM

Application Transformation

ON DEMAND BUSINESS™

On Demand Business for insurers

"On Demand Business for Insurers, the IBM Primer," is a series of six booklets that describes the on demand world for insurers and provides a roadmap for the journey. This series has been developed with direction from the IBM Financial Services Marketing Leadership Team. Booklets in the series include:

Booklet 1: *"Insurance: Succeeding in the On Demand World" – presents a fundamentally new way for insurers to operate, its benefits, and various paths for achieving those benefits.*

Booklet 2: *"Business Transformation" – develops the Component Business Model and the IBM Insurance Application Architecture (IAA) as means toward transforming a business.*

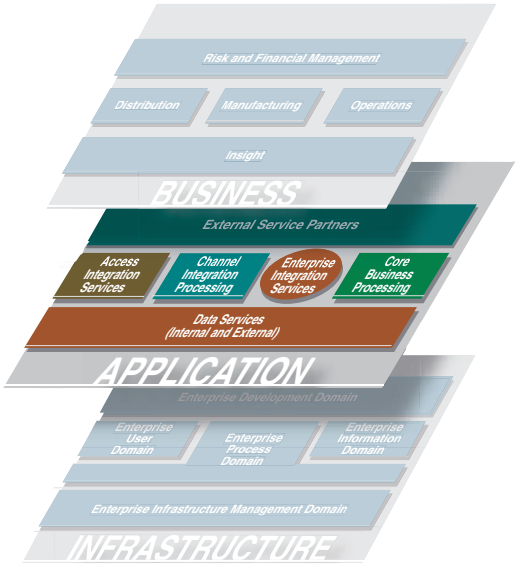
- ***Booklet 3:*** *"Application Transformation" – presents the benefits of a Service-Oriented Architecture and two areas of application transformation: legacy reuse/modernization and model-driven custom development.*

Booklet 4: *"Infrastructure Transformation" – discusses the relevance of an On Demand Operating Environment for insurance, an optimized, resilient and secure infrastructure aligned with business goals.*

Booklet 5: *"Insurance Solutions" – details the broad scope of IBM insurance solutions that include consulting, software, hardware and services.*

Booklet 6: *"Successes along the On Demand Journey" – gives numerous case studies of how successful insurance companies worldwide benefit from IBM insurance solutions, along with descriptions of IBM briefing and competency centers in various countries.*

Contents



Common issues of existing insurance systems

Over the years, insurers have built applications that support their key business processes, but these applications often used proprietary systems that are now outdated, inflexible and inefficient. Exorbitant amounts of money and resources were needed to maintain them, and they limited quick adaptation to growing opportunities, competitive threats or customer behaviors. Unfortunately, efforts to replace or upgrade legacy systems have often resulted in increasing overall IT complexity. Similarly, the promises of insurance vendor application packages to address this have often failed to materialize when required customizations and functional extensions disrupted planned product update or maintenance cycles. Insurers can break free from the grip of ever-growing legacy applications by relying on industry standards such as ACORD and IBM Insurance Application Architecture (IAA) to transform their application portfolio.

As discussed in the first booklet of this series, enterprise reconstruction requires insurers to optimize their business services portfolio by removing silos, duplications and inefficiencies. It also means componentizing the business for on demand operations, while at the same time leveraging existing investments. Business functions tend to duplicate over time because of applications that are specific to a line-of-business (LOB) or to different channels or products. These duplications are compounded by different development technologies and methodologies. Mergers and acquisitions of enterprises with similar portfolio proliferation problems have also complicated business functions. The complexities of these environments and the need for multiple skill sets have resulted in high development, testing, and maintenance costs. They have also led to longer development cycles, which translate into slower time to market for new business capabilities and products. Diversity and duplication in the business function portfolio are major inhibitors to business change and to moving to a Component Business Model (CBM).

Highlights

This booklet focuses on how an environment can be transformed to integrate applications across the insurance enterprise and make legacy systems more flexible, efficient, and dynamic.

Application transformation refocuses activities from specific line-of-business systems to business processes that span the enterprise. Far-reaching and comprehensive, the value proposition of transforming applications includes:

- Reduced cost of maintenance and operations
- Product and service innovation
- Implementation of a shared-services business strategy
- Improved product distribution
- Avoidance of technology obsolescence
- Facilitation of mergers and acquisitions
- Easier regulatory compliance
- Improved customer centricity
- Reduced time to market for products

This booklet discusses the application aspects of the on demand journey. It focuses on how an environment can be transformed to integrate applications across the insurance enterprise and make legacy systems more flexible, efficient, dynamic and adaptable to constantly changing needs.

Approaching application transformation with Service-Oriented Architectures

The difficulties of transforming core application systems are not new, but the business impact of failures to transform applications is growing – the more transformed, flexible and responsive the business itself needs to be, the greater the impact of an inflexible application portfolio.

Highlights

A Service-Oriented Architecture is the key to overcoming the inhibitors to application transformation.

The inhibitors to application transformation are well known:

- Business processes are a mixture of people practices, application code and interactions between people and systems, or systems and systems.
- There is no single data /business /process model across (or beyond) the enterprise.
- Changes to one system tend to ripple at many levels to other systems.
- No single, fully functional integration solution will talk to them all.
- Deployment of any single, proprietary integration solution across the enterprise is complex, costly and time-consuming.
- A multitude of technologies and platforms is needed to support business systems.
- An integration solution is unable to talk to present and future partners.
- Not all integration technologies work as well across a wide area network or the Internet as they do across a local area network.

The key to overcoming these inhibitors is implementing a Service-Oriented Architecture (SOA), that is, employing the architecture as a collection of services that communicate with each other. An SOA enables more rapid and pragmatic response to business transformation. IT can rationalize, simplify and enable new capabilities within the application portfolio while reducing duplication, complexity and cost. A Service-Oriented Architecture is to application transformation what the CBM is to business transformation. For the insurance industry, the transition between these levels is supported by the traceability features of IAA. These traceability features define all of the services required for supporting each business component, thus providing a seamless path from business transformation to application transformation.

Highlights

The SOA vision is a flexible portfolio of functionality provided from an appropriate mixture of existing systems, new build components, and purchased components that can be choreographed to whatever business processes are required.

Without SOA, the application portfolio becomes an inhibitor to both business and infrastructure transformation since evolving the business functionality would typically have to be reflected across many different systems. An application transformation with a Service-Oriented Architecture enables the IT and network infrastructure by allowing existing application functionality to leverage, and be integrated into, an on-demand Operating Environment. The SOA vision is a flexible portfolio of functionality provided from an appropriate mixture of existing systems, new build components, and purchased components that can be choreographed to whatever business processes are required.

A Service-Oriented Architecture overcomes the inhibitors to application transformation in the following ways:

- SOA enables processes and applications to be decoupled. This means business processes can be managed and rationalized and that applications can be transformed.
- SOA drives the development of a set of modular services that can be reused across multiple processes. This facilitates the adoption of enterprisewide and even industry process models, such as that contained within the IBM Insurance Application Architecture.
- SOA encapsulates the business functionality in a set of discrete services that are decoupled from the business process through the service interface. This allows processes to be changed without necessarily impacting the business functionality, and it bounds, localizes and changes the functionality supporting a service. This discipline eliminates the undesirable ripple effect so common in today's complex web of processes, applications and integration.
- SOA utilizes Web services to provide the common access and invocation mechanism for all the services, thereby insulating the process from the application integration issues.
- Behind the service interface, a variety of traditional integration techniques can be used to leverage existing application functionality.

Transforming the application portfolio to a Service-Oriented Architecture delivers modular business functionality that can be rapidly combined in a plug-and-play fashion into new business processes to shorten time to market, while still enabling simplification and transformation of the application portfolio.

In summary, there are three key features of a Service-Oriented Architecture:

1. The functionality to support a step in a process is implemented once, and once only, and is encapsulated by a service interface. This reduces complexity and makes changes easier and less costly to implement and maintain. For example, the functionality of retrieving a customer address is implemented only once across the enterprise.
2. Processes and business functionality are decoupled through the service interface allowing either to be changed without necessarily impacting the other. This provides business flexibility and drives out cost by portfolio rationalization and simplification. For example, the order in which a claim is validated could be modified without impacting back-end systems. Conversely, billing functionality could be re-implemented in back-end systems to improve efficiency from an IT perspective without impacting business processes.
3. Services can be reused in a plug-and-play fashion across multiple processes. This provides business flexibility and reduces implementation time, making IT more responsive to the business.

Highlights

A legacy diagnostic identifies gaps between the desired business capabilities as expressed in the Component Business Model and business process analysis engagements and the actual functional capabilities of the legacy systems.

Portfolio assessment enables an insurer to assess the amount of duplication of functionality across several legacy systems.

Diagnostics: Defining opportunities for application transformation

Component Business Modeling (CBM) can help define investment priorities at the business level. Before embarking on a costly path in terms of rationalizing or evolving an application portfolio, it is critical to clearly define the priorities in an enterprisewide way. This can be done through several diagnostic activities:

Legacy (functional) diagnostic – This activity identifies gaps between the desired business capabilities as expressed in CBM and business process analysis engagements and the actual functional capabilities of the legacy systems. For example, by defining the "to be" business processes using the IAA Critical Business Processes as a basis, an insurer might realize that the current legacy system cannot support a particular business functionality. Examples of functionality shortages include the incapacity to provide a real-time view of the claim processing to a customer, the inability to provide combined statements for multiple policies, the lack of overall risk measurement capabilities, and the absence of a customer overall needs profile. In many cases, the functionality cannot be provided due to the lack of an enterprisewide vision of the applications.

The result of such an activity is typically a list of desired business functionalities, which can form the basis for a business case supporting various aspects of application transformation.

Portfolio assessment – This activity describes the existing applications across the entire enterprise according to a framework of services. It enables an insurer to effectively assess the amount of duplication of functionality across several legacy systems. Common examples of duplication include management of customer-related information, execution of business rules, and handling of funds, rating, and policy management across lines of business.

Highlights

Breaking the grip of legacy solutions and transforming the applications portfolio involves three options: transform what you have, buy new, or build for yourself. IBM Insurance Business Solutions accommodate all three approaches.

To be able to perform a portfolio analysis, an enterprisewide list of service definitions is required such as the IAA Enterprise Component Blueprint (ECB), which defines a normalized set of functional services to support an insurance business.

The ECB contains about 300 unique service definitions, such as "request risk assessment." In practice, the functionality described by this service will be implemented in multiple legacy systems, often in conflicting ways. This is a strong inhibitor to business agility. Evolving this capability requires updating several systems, which creates additional work and increases the risk of errors and inconsistencies.

Performing such a portfolio assessment is the only way to really understand how to reconcile the business and IT priorities. The result is a duplication map that describes all the functional overlaps between the existing legacy applications.

Delivery: The choices of transform, buy, build

Breaking the grip of ever-growing legacy solutions and transforming the applications portfolio typically involves three choices: (1) limited forms of legacy repurposing ("transform or reuse"), (2) package integration ("buy"), or (3) custom development ("build"). However, insurers gain the most benefit when they evaluate all three options. Complex relationships between applications like distribution, underwriting, policy administration and financial management also demand deployment of a common framework such as the IAA Enterprise Component Blueprint. As individual applications are transformed or updated, the relationships with other applications must be fully considered. This enables the transition to be smoother with incremental steps leading to a unified and consistent strategic environment.

Highlights

This booklet focuses on two areas of application transformation: legacy transformation (or reuse) and custom development (build).

IBM Insurance Business Solutions were created to accommodate all three approaches to portfolio management, providing a full suite of integrated solutions based on a common framework. The portfolio of solutions relies on open industry and technology standards and deploys a layered architecture that isolates users, data and applications, while componentizing overall IT solutions. This approach allows an insurer to phase out legacy applications without the typical disruption that occurs with the deployment of new solutions. This is because a layered architecture controls retirement of older solutions and the deployment of new, highly configurable applications. This limits the customizing that, historically, has added complexity to the IT environment.

The IBM Insurance Business Solutions are described in detail in booklet five of this series. The focus here in booklet three is on legacy transformation (or reuse) and custom development (build).

Legacy transformation

Inflexible and costly back-office systems impede product strategies, customer centricity, distribution and services. However, while these systems present obstacles, they also represent a significant investment in sound business logic and rules that is worth leveraging going forward. Not surprising, then, is the growing focus on legacy transformation by both insurers and vendors.

Legacy transformation takes many forms, and insurers need to determine which options best align with their strategic imperatives. At the highest level, it is possible to distinguish between functional (impacting the business functionality of the systems) and non-functional (IT-focused) transformations.

In addition, IBM can help insurers who face complex or specialized data migration issues with offerings targeted at the following three areas: mergers and acquisitions, system rationalization and application replacement, and the building of management information systems. These areas represent trends and directions in the industry, and each can result in significant cost and pain if not addressed properly.

Highlights

Insurers with systems that adequately support their business requirements but are characterized by inefficiencies, such as substandard performance, high resource requirements or cumbersome management and maintenance methods, should consider offerings that address non-functional transformation.

Non-functional transformation (IT-focused)
Offerings addressing non-functional transformation should be considered by insurers who have functionally adequate but inefficient systems and supporting processes. In other words, the systems support their business requirements but they have substandard performance, higher than anticipated resource requirements, or require cumbersome methods of managing and maintaining the systems environment. Through the right combination of application management and IT services, insurers can streamline and optimize their technical infrastructure, consolidate core systems, and implement more cost-effective IT processes, organization and technology.

IBM offerings address the following three areas:

Systems optimization – to help insurers drive marginal improvement with their existing core capabilities aimed at achieving cost reduction and extending the useful life of legacy applications.

Consolidation and migration – to achieve reduced complexity and more cost-effective application operations, more accessible and flexible data schema, improved corporate data accuracy, increased manageability of business rules and logic, less IT staff workload for legacy maintenance, a less complex application portfolio better positioned for transformation, and elimination of unsupported domains and duplicate applications.

Network optimization and convergence – optimizes the network infrastructure to combine voice, data, and video streams into a single converged IP network, and, in turn, reduces operating costs. Reinvesting a portion of the realized savings makes it possible for new applications, such as voice over IP (VoIP), "click to chat," video conferencing, and existing insurance applications such as contact centers to be extended and enhanced to take advantage of the standards-based functionality in the converged IP network. This enables new levels of customer, partner and employee service. Constituents are able to access products and services from a wider range of venues – hotel rooms, airport lounges, WiFi cafes – without incurring greater communication costs in the process.

Highlights

Functional transformation (business-focused)
Offerings addressing the business functionality of legacy transformation include the following:

Application integration – provides near real-time, enterprisewide access to critical business data, improves data sharing and collaboration among employees, helps ensure data accuracy for better business process management and strategic decision making, leverages investments in existing legacy applications as part of the ROI equation, and improves application robustness and operational reliability.

In the case of functional evolution, the best approach is to define services at the middleware level to act as an interface to the legacy.

Web enablement – optimizes Internet-based interactions with partners, suppliers, and customers.

Application renovation – unlocks the management of business logic from legacy applications, increases application development responsiveness to business needs, leverages reuse of existing application functions in new business processes, generates increased revenues by enabling the business to adapt more dynamically and capture new market opportunities, and moves from costly custom development processes to rapid assembly of applications using Web services and salvaged legacy components

"Wrapping" the legacy behind services
In the case of functional evolution, the best approach is to define services at the middleware level to act as an interface to the legacy. In other words, the legacy systems offer an implementation of the exposed service definitions. There are multiple technologies that can support integration – messaging and integration hubs, collaborations, services choreography – but, regardless of the technology, one has to define the business content for this integration layer.

In SOA terms, that means defining all the enterprisewide services that can be invoked to support the business processes. Regardless of how the legacy applications are structured, their functionality is exposed through this consistent layer of services, which can be deployed within a run-time environment.

Highlights

IAA defines the services at this level as well and as consistently with the diagnostic-level services discussed earlier. By customizing this framework of services for specific aspects of their applications, insurers are able to more easily define the appropriate set of services required in their enterprise. In addition to the service definitions, IAA defines how these services relate to each other, or the "service choreography."

When legacy systems are "wrapped" behind a layer of services, it is possible to gradually evolve the back-end applications with managed impact. As long as the service definitions are maintained, the actual implementation can evolve over time from legacy systems to custom-built or acquired components, once they become available on the market, with no impact on the consumer.

The state of the art in terms of application development is to use an approach that is both model-driven and component-based.

Model-driven custom application development

The state of the art in terms of application development is to use an approach that is both model-driven and component-based. As described earlier, these components can then be deployed to implement a set of enterprisewide services.

Model-driven development

Models can be used in several ways. For example, to rationalize an application portfolio, to understand business processes and to define a layer of services to support a SOA. When building applications, the role of models is even more important as they are the only way to make sure that the end result will be in line with the business requirements and will be maintainable over time as the business evolves.

Highlights

Insurers who pursue a staged, model-based approach to building custom applications guarantee the consistency of the resulting code and the traceability of the resulting solution components back to the analysis-level business concepts from which they are derived.

The key factor underpinning a successful custom-built solution is a common enterprisewide description of the business concepts that define the business data entities manipulated by the application components. Without this common language, any attempt to support a consistent and flexible architecture will be more difficult. The IAA model supports a complete and unambiguous description of the business concepts, business activities, and business rules that must be supported within an insurance company.

Insurers who pursue a staged, model-based approach to building custom applications guarantee the consistency of the resulting code and the traceability of the resulting solution components back to the analysis-level business concepts from which they are derived. The model-based approach to custom application development assists greatly in the successful deployment of component architecture across the enterprise, and enables the efficient and accurate gathering of requirements. Refinement of these requirements within a technology-neutral interface clearly identifies business responsibilities. Additionally, the direct generation of technology-specific code from this design model guarantees the consistency of definitions with a single integration effort or across multiple projects.

Models can be defined at any level, from a very business-oriented description of an insurer's processes down to the level of objects or databases. The promise of service orientation can only be fully realized through a good connection between the business (as expressed for example in a set of business process models in a tool like the WBI Modeler) and the IT description of the services.

This has important organizational consequences. There must be good cooperation between the group in charge of defining the business processes and the architecture group in charge of the service definitions. Any change in the business processes needs to be properly managed in terms of its impact on the service layer. Formalizing the mappings between processes and defining clear transformations between the business and IT levels are important success factors for the deployment of service-oriented solutions in an enterprise.

Highlights

A component-based approach provides major functional components for building a fully customized solution or evaluating packaged solutions against an organization's system architecture.

Component-based development

Monolithic applications or package solutions deliver, in general, a set of functions to meet a particular purpose that is based on a generic set of requirements. During the implementation of these packaged solutions, companies tailor them by adding extra functions as required and by deleting or modifying functions that do not match the business requirements. Inherent to this approach is the fact that most packages usually set rigid limits on the scope of additions and modifications.

A component-based approach does not generally provide a ready-to-use set of business functions like those provided by a package. Instead, it provides major functional components, from which a fully customized solution can be built or against which packaged solutions can be evaluated for fit within an organization's system architecture. With a consistent set of interoperable components, it is possible to provide a solution that is better tailored to the needs of the business without having to build from scratch.

Recently developed packaged applications increasingly define their interfaces in a component style, which makes them much more suited to component-based and service-based architectures.

Components to implement the services

Although there are two approaches to SOA enterprisewide services – integration (to the legacy systems or package solutions) and components – in almost every case, the best solution is a mixture of both approaches.

When opting for a component approach to support the implementation of services, an enterprise component architecture must be defined: How do the components relate to each other, are there dependencies, how do they collaborate, and can they be deployed independently?

IAA proposes an enterprisewide set of components for the insurance business. These components are defined to support an enterprisewide set of SOA services. These two layers are kept completely synchronized and traceability is enforced.

Highlights

IBM offers many valuable technologies to help organizations architect, build, deploy and manage service-oriented solutions.

In technical terms, IAA proposes one platform-independent model (PIM) and from there, consistently targets multiple technologies such as Web services to support an SOA, or J2EE® beans to support component-based development.

The advantages of the component approach are full consistency between the levels (from business and process definitions to services and components), an enterprisewide view that reduces duplication of functionality and maintenance costs, and an external validation element that reduces the risk of overlooking business requirements or embarking on a dubious design.

IAA provides insurance content on top of the IBM Software Development Platform, a set of tools and capabilities that helps realize the SOA vision.

The IBM Software Development Platform

IBM offers many valuable technologies to help organizations architect, build, deploy and manage service-oriented solutions. While individual product capabilities are important, the real value is the combination of these capabilities in a robust software development platform for creating this new generation of applications. Moreover, the vision expressed by organizations today is for a set of capabilities to execute IT projects with a level of coordination, accuracy, and clarity that is rarely achieved today. In fact, in the post dot-com era, IT is seen as a core "utility," one in which investment in IT resources provides a predictable, risk-managed impact on the goals and mission of the business. Consequently, organizations are beginning to view software development as a "business process" in and of itself that must be measurable, predictable and manageable.

This is a compelling vision, and one that can only be delivered through the deep integration of tool and run-time capabilities across all different aspects of the business in support of a services-oriented view of their solutions. The IBM Software Development Platform offers the tooling and technology infrastructure to realize that vision. With respect to SOA, the IBM Software Development Platform addresses four critical needs:

Highlights

IBM Software Development Platform addresses four critical needs with respect to a Service-Oriented Architecture:

- ***Bridging the business-to-IT gap***
- ***Supporting the changing roles in the IT organization***
- ***Focusing on assets and reuse***
- ***Increasing levels of collaboration***

Bridging the business-to-IT gap. It is essential to align the business view of activities and processes with the technology that is used to realize (parts of) these activities. This alignment includes the ability for business models to drive downstream development, and to evolve the business models and IT solutions in combination. The service concept is critical to this alignment. Services and service-based thinking form the common ground that ties business analysts, IT architects, integrators, and developers together. Common design practices are essential to this to ensure that the concepts, artifacts, and activities are synchronized across these different perspectives. Finally, having tools that can efficiently transform models representing the business intent into efficient designs and implementations is critical for bridging the business-to-IT gap.

Supporting the changing roles in the IT organization. The move to services thinking changes the skills and composition of teams in an organization. The focus of development is on finding, defining, managing, and assembling services, with architectural descriptions highlighting service level agreements (SLAs) and inter-service protocols. The traditional breakdown of tool functions into today's lineup of products is not appropriate to this approach. There will be a different blend of capabilities required by the different members of an IT organization. For example, the skills required by existing roles such as "software architect" are changing to include greater emphasis on assembly and management of services across a diverse set of service providers. Similarly, new roles such as "integration specialists" are emerging, whose focus is on assembling a services-based value chain in support of an organization's key business goals.

Focusing on assets and reuse. Considering services as key assets in the design of systems changes an organization's view of the value of reusing these services. A service assembly viewpoint leads to "software factory" thinking. As a result, technologies and techniques for management and governance of assets, and repeatable ways to capture patterns for combining assets, become much more important. In an asset-based development approach, these assets hold critical value to the organization and must be carefully managed and administrated. The team infrastructure for managing assets takes on a key role in this approach.

Highlights

The IBM Software Development Platform helps insurers to understand, define, build, and deploy collections of services that support the business, and to monitor those services in execution to provide feedback to optimize business operations.

Increasing levels of collaboration within and across practitioner roles. Enterprise application development has always recognized that software development is a "team sport," and focused attention across the life cycle on managing shared assets, artifact traceability, and shared practices and processes. The collaborative nature of software development is increasing with greater geographic distribution of organizations, enhanced real-time communication among individuals in teams, and software being embedded as one part of broader systems development initiatives. Increasingly, the role of software development infrastructures will be seen as a collaborative development environment for software practitioners that encourages sharing and reuse of services across teams.

The IBM Software Development Platform supports a business-driven development life cycle aimed at bringing together the tasks and the varying roles that collaborate in any enterprise-scale project. They work together to understand, define, build, and deploy collections of services that support the business, and monitor those services in execution to provide feedback to optimize business operations.

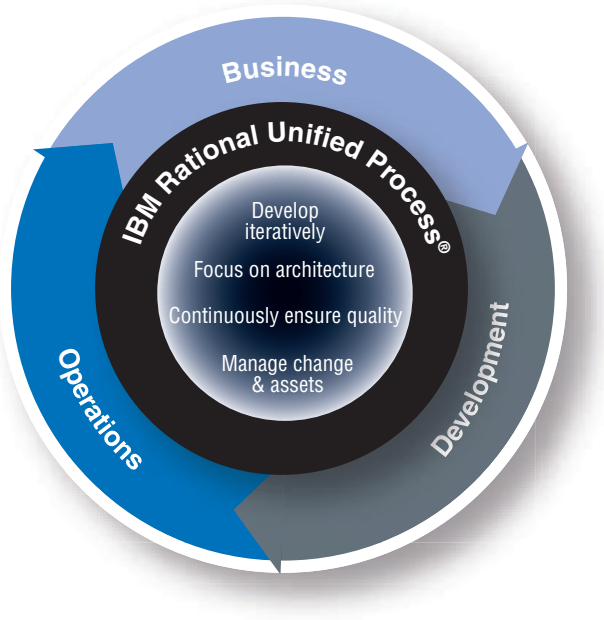


Figure 1. *The business-driven life cycle and the IBM Software Development Platform*

Highlights

IAA together with IBM's Software Development Platform accelerates and decreases the risk of component-based development projects.

For practical efficiency, the creation and delivery of IBM's rich, integrated Software Development Platform has been built upon a common tooling infrastructure based on a set of shared components. The Eclipse infrastructure with its plug-in architecture, meta-model framework, shared meta-models, and libraries of capabilities makes this possible. Through use of common components among IBM development teams built upon this shared infrastructure, the IBM products can be used together more easily, are open to extension by third parties, have greater consistency and quality, and can be evolved more efficiently.

In summary, IAA together with IBM's Software Development Platform accelerates and decreases the risk of component-based development projects in the insurance industry.

Moving forward to infrastructure transformation

This booklet has discussed the benefits of Service-Oriented Architectures and two areas of application transformation (legacy reuse/modernization and model-driven development) to support the business vision and strategy defined in booklet two on business transformation. The third transformation required on the road to on demand insurance involves the infrastructure – designing an infrastructure (or an On Demand Operating Environment) that operationalizes an insurer's application solution portfolio. This subject is discussed in booklet four of this series, "Infrastructure Transformation."

IBM Corporation
1133 Westchester Avenue
White Plains, NY 10604
U.S.A.

Produced in the United States of America
5-05

G299-0617-00