



How you can drive business success by efficiently supporting the development of reusable software components.

Advances in Software Engineering have brought about rapid strides in enabling true reusability. Today, Component-based Development (CBD), Service Oriented Architecture (SOA) and Model-Driven Architectures offer tremendous advancements for developing complex software and systems. However, software reuse and its associated benefits have not scaled systematically. One of the biggest reasons is the inability of software teams to develop, share and manage components through an underlying framework of integrated and optimized process patterns and change management techniques. To gain repeatable benefits from a single asset, you must incorporate reuse effectively in your development organization.

IBM can help. We can show you how you what to consider when choosing a software change and configuration management (SCM) solution. An SCM solution helps provide an innovative approach that enables reuse of software assets across your entire development organization.

Maintain your components in a reusable, distributed fashion

One of the primary motivations behind implementing component-based development is reusability. Software change and configuration management (CM) software should help achieve this by providing for a repository where components are uniquely identifiable and referenceable, with metadata to identify characteristics of components as well as dependencies and relationships, including those with associated non-source artifacts.

Best practices and patterns for component-based development should be made available out-of-the-box, with flexibility for managing variants that you may encounter. The development environment should support reuse engineering project rules—ranging from agile teams to large, distributed programs—by decoupling the process model from the component under development. This allows you to then leverage your design environment for more flexible component hierarchies.

Overcoming software asset management challenges

Any software change and configuration management solution you use should support component consumer discovery, distribution and publishing. The environment should provide for

identification of published versus internal components, component sharing management, differentiation between component types, source component customization and the ability to integrate these changes into source component releases.

A CM system should also closely align with your organization as developers begin creating domain-specific reuse models. With component-based technologies and middleware frameworks becoming more mainstream, the solution should natively support a layered architectural approach to design and development—promoting true business alignment across your organization.

Implementing a good architectural approach, coupled with a promotional model for intra- and inter-component testing can help lead to faster identification of issues and also a reduced ripple effect when modifications are made. The environment you use should contain a mechanism for monitoring feature interaction and testing prior to production build, and also offer separate build areas for insulated, collaborative, integration and system testing with associated process rules.

Coordinate and monitor development while also mitigating design risk

An end-to-end task-based change and configuration management system (TBCM) can provide for higher component interaction, and help you establish a fast, reliable process between component consumers and developers.



Tasks let you monitor and design asset relationships with extensible, customizable attributes, which provides for greater definition, conflict resolution, dependency knowledge, functionality propagation, tailored workflow and change management processes.

The challenges posed by CBD and software reuse are daunting, even to the most skillful practitioners. Organizations can minimize this issue and plan for continuity by switching to a system that lets you manage composed components at any level of granularity for integration and reuse. Conversely, you must be able to also decompose components into constituent parts to enable versioning, restoring, simulation, parallel development and value derivation.

Build user confidence

The transition to software components is often fraught with uncertainty, due to the inherent abstract and business critical nature of this task. Any software change and configuration management solution you choose should make the task of managing software assets, change requests, and application configurations transparent to your developers, so that they can focus on what they do best: solving core architectural and design issues.

IBM Rational Synergy platform

In a broader sense, an asset may be a software component, a design pattern, an architecture layer, a business process, a test algorithm, etc. Its development framework should combine with application lifecycle management (ALM) disciplines such as requirements management or enterprise architecture in order to provide a scalable and systematic approach to development, resulting in higher quality products and faster time to market.

IBM® Rational® Synergy is a task-based change, configuration, and release management platform that brings together global, distributed development teams. Synergy software provides high-quality software configuration management capabilities to support your complex development needs. You can now leverage a collaborative development process across the globe that you manage centrally, with visibility to all levels of your development process, and achieve the benefits of ALM on a global scale.

For more information

To learn more about how you can use IBM Rational Synergy to help you efficiently reuse your software components, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/rational

© Copyright IBM Corporation 2009

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
June 2009
All Rights Reserved

IBM, the IBM logo, ibm.com, and Rational are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

Other company, product, or service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

The information contained in this document is provided for informational purposes only and provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. Without limiting the foregoing, all statements regarding IBM future direction or intent are subject to change or withdrawal without notice and represent goals and objectives only. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.