IBM Software
Thought Leadership White Paper

October 2011

Delivering value through best practices

Process support in systems and software development

IBM

Executive summary

Our planet gets smarter every day. Worldwide mobile phone subscriptions reached 3,300,000,000 in 2007. An estimated 35,000,000,000 devices were connected to the Internet at the end of 2010, and across the globe there exist approximately 1,000,000,000 transistors per person—with the cost of each transistor averaging one ten-millionth of a cent[1].

Smarter products are everywhere: Smart phones, smart cars, smart medical devices, smart planes and smart power grids. Furthermore, these products are often interconnected and interdependent, forming "systems of systems" that provide higher value than any of the individual products could provide on their own. For example, cars can self-diagnose problems, download updated software and let the driver know when a visit to the service center is required. Smart power grids can negotiate with appliances to deliver the comfort and convenience we require while reducing energy consumption and costs. And every day we rely on our smart phones to deliver news, entertainment, social connectivity, location-based services and more. We truly live on a smarter planet where intelligent, instrumented and interconnected products are used by everyone, everyday.

Challenges for smarter product development

But to bring these smarter products to market presents many challenges. Engineering teams must collaborate across different disciplines, different organizations, different time zones and different languages. They must manage complex design information efficiently and respond to rapidly changing market and customer demands, evolving interfaces and product variants. These teams must ensure quality and compliance with standards and regulatory mandates. And they must do all of this while delivering on-time and within budget in a highly competitive world.

To meet these challenges, the leaders of engineering organizations must capitalize upon smarter development approaches. This paper examines the opportunities presented by the deployment of proven practices to systems and software delivery processes.

Objectives for your development environment

The challenges of smarter product development reveal clear business objectives for your development environment. These objectives include:

- Reducing the cost of your development activity and your delivered product
- Improving the quality of your delivered product
- Reducing the development time for your delivered product
- Reducing the risk of failure to meet the cost, schedule and quality requirements for your delivered product
- Increasing the consistency and predictability of development activities

The relative weighting of these generic factors should be considered in the context of your development environment. For example, time-to-market may be the critical driver for a mass-consumer technology product, whereas quality and standards compliance might be the key driver for a safety-critical avionics system. It is therefore important that your development environment has the flexibility and adaptability to address the scope of products and systems that the organization will develop. To ensure sustainable competitiveness, your development environment must also support continuous process improvement.

Defining your development environment

A development environment is a combination of people, process and technology. The development environment is composed of four main elements:

- The development process—What to do and when to do it
- The development methods—How to do it
- The roles that execute the development methods—Who does it
- The tools that automate aspects of the development methods

Tools add value to the process by providing automation, information management, workflow support and reporting. Tools can support and enforce the development process, and provide

governance and "safety nets" to keep people on the right track, but tools are only one component of the optimal development environment. A complete solution provides the linking of tooling and roles through process and methods.

Using practices to optimize your development environment

Changing the development environment represents a risk in any development program. Also, a development organization will typically have existing processes with which any changes should be integrated. Incremental adoption can minimize the risk of changes, because your team can assess the value and stability of each change before making further optimizations.

Incremental adoption requires a modular approach to the deployment of tooling, processes and methods. It also requires a holistic approach, which helps you to avoid unnecessary local optimizations that cannot be reused (and therefore provide less business value). Practices are a mechanism to help you achieve the required modularity and reuse.

A practice is a unit of process and method to support one or more roles in using tooling to achieve a particular objective.

A best practice should:

- **Be self-contained—It should provide you with support for achieving a particular engineering objective without being a monolithic "super process" that requires an "all-or-nothing" approach to implementation.**
- **Define inputs, outputs and interfaces in a clear manner to other practices and disciplines.**
- **Define and provide workflow support.**
- **Integrate with appropriate tooling solutions to automate workflows—through templates, profiles and tool mentors.**
- **Provide implementation guidance for practitioners—to aid training and to support daily workflows.**
- **Support project management and ongoing optimization—through automated reporting and metrics measurement.**

The benefits of defined practices

Proven practices can provide a number of benefits at both the operational and business level. These benefits include:

- **Reusability**. Practices can act as a "corporate memory," enabling successful processes to be redeployed for specific activities within multiple projects.
- **Repeatability**. Practices can provide a known baseline of performance which can be used as the basis for iterative process improvement.
- **Improved auditability** and simpler regulatory compliance processes, strengthened through defined deliverables and reporting.
- **Better delivered quality** resulting from defined handoffs between roles and activities enabling automation of information flows.
- **Improved collaboration** resulting from clearly defined roles, work and information flows.

In addition to these primary benefits, the reuse of practices can also lead to more efficient transfer of personnel and skills between projects.

Areas for the application of practices in development projects

The principle of practices can be applied to realize benefits throughout your development activity. In complex product and systems development, systems engineering and software development are areas which are increasingly critical to project success.

Systems engineering is the process of deriving an optimum systems architecture and allocation of functionality from initial stakeholder requirements. Some of the key systems-engineering activities include:

- The capture, analysis and elaboration of requirements.
- The derivation of candidate system architectures through approaches such as operational analysis, functional analysis and use-case analysis.

- The selection of an optimal architecture using techniques such as trade studies.
- The "hand-off" of systems engineering output information to detailed design groups.

Software engineering covers the development lifecycle for the implementation of software components, based upon the specifications that are "handed off" from your systems engineering process. Some key software engineering activities include:

- Software requirements analysis and architectural design.
- Software architectural modeling.
- Real-time interaction modeling.
- Detailed software design and implementation.
- Safety and reliability analysis.
- Testing and integration.

Each of these activities is a candidate for implementation through a practice.

Implement practices intelligently

A practice must be defined before it can be implemented, but implementation does not end with definition. The medium in which the practice is defined and distributed is also crucial. Simply writing down the process and methods that make up a practice is likely to lead to "shelfware"—practices which are either not followed or are applied inconsistently. It is critical that practices become the basis of workflows for the development team rather than being seen as an "add-on" to the development effort or as the cause of additional cost or complexity.

Appropriate development tooling can be used to facilitate the adoption of practices. Elements of tooling which are useful in this context include:

- A means to author, adapt, maintain and deploy practices.
- A means to automate workflows and information flows—especially throughout large and distributed teams.
- Automation tools for engineering processes.
- Reporting and measurement tools.

To work effectively, these categories of tooling must be integrated to permit the transfer of information in a way that is efficient and error-free. For example, the published form of the practices should be able to interact with your workflow and engineering automation tooling, and your reporting and measurement tools should be able to extract information from workflow and engineering tools.

Interaction may take different forms over different tooling interfaces. Of particular value for practice implementation are:

- Context-specific linkage between automation and workflow tooling and the published processes and methods.
  - This linkage helps users to move efficiently between process documentation and the enactment tools.
- Tool mentors—which can provide in-context, on-demand training and guidance to facilitate rapid skills acquisition for new team members.

The IBM Rational solution for systems and software engineering

The IBM® Rational® solution for systems and software engineering provides collaborative, integrated systems engineering and embedded software development capabilities. These capabilities help your organization to build the systems, technology-enabled products and services that encourage success in today's competitive marketplace. This IBM solution comprises a combination of world-class practices and tools on an open integration platform that help unite mechanical, electronic and software disciplines, assisting you in your efforts to accelerate delivery and to improve quality.

Platform architecture

Figure 1 shows the core capabilities of the IBM Rational solution for systems and software engineering. This solution is built upon IBM Jazz™, the IBM initiative to transform systems and software delivery by making it more collaborative, productive and transparent.

Jazz is an open platform that is designed to improve the delivery lifecycle by breaking down the walls between different development activities and the tools that support them. The Jazz Integration Architecture incorporates specifications defined by the Open Services for Lifecycle Collaboration (OSLC) project, an independent, multivendor effort to define a set of protocols for sharing information between tools and vendors. Tools may be either built upon the Jazz platform or integrated with Jazz using OSLC specifications.

Solution components

The IBM Rational solution for systems and software engineering offers a core set of capabilities which span key activities of the systems and software engineering lifecycle. These capabilities include:

Requirements management to help you to elicit, engineer, document and trace requirements throughout the lifecycle. This helps your stakeholders to specify their needs and helps the project performance to deliver against those needs. The through-lifecycle traceability provided by IBM Rational requirements management helps your teams to remain focused as change occurs during development.

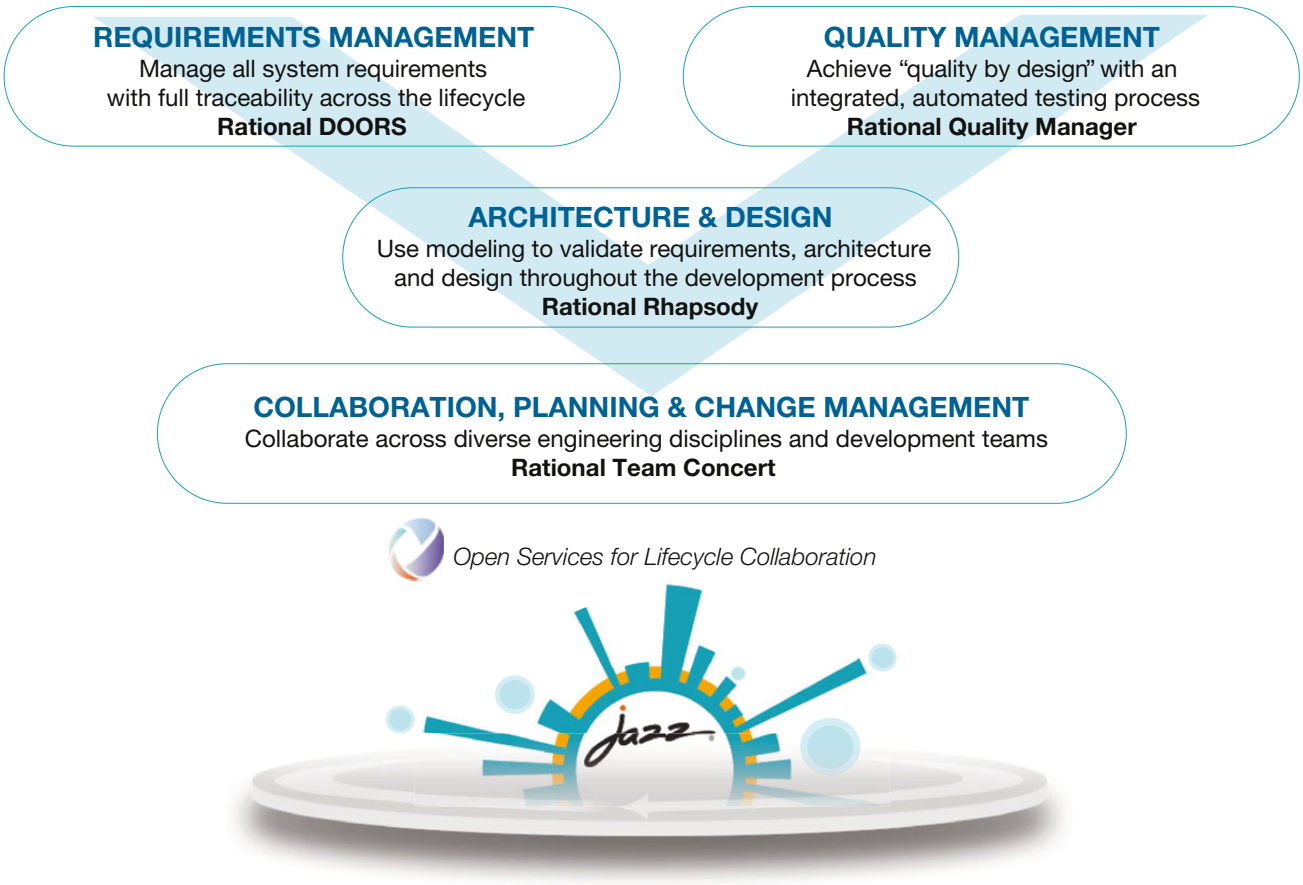

Figure 1: The IBM Rational solution for systems and software engineering

Architecture modeling and model-driven development to enable teams to validate requirements, to visually derive and define architectures and to create robust real-time and embedded software designs. Execution of models enables the early verification of architectures and software designs, improving quality and reducing time and costs by avoiding late-stage rework.

Quality management capabilities to help your teams to achieve "quality by design" through a collaborative approach to quality planning, the ability to test traceability to requirements, automation of tests and a smart, integrated defect management process.

Collaboration, workflow and change management functions of this IBM solution to help diverse and distributed development teams to work together efficiently and effectively. To help improve your ability to integrate planning and execution, automate workflows and manage change throughout the lifecycle.

In addition to these core capabilities, the IBM Rational solution for systems and software engineering is extensible to meet the requirements of different industries, development approaches, regulatory and compliance standards. Extensibility may be realized through additional IBM offerings and through third party offerings that use OSLC interfaces.

IBM Practice Library

In addition to the tooling components of this solution, IBM offers a practice library which comprises a set of loosely coupled, ready-to-deploy practices. These practices are defined using a standard language that is developed specifically for systems and software process descriptions. The solution provides process authoring, adaption, deployment and management capabilities to support the customization of practices to your organization's needs. The architecture of the library is designed to simplify the composition of the overall development process, using practices and existing process elements to meet your business and operational objectives.

IBM practices address a particular aspect of the development process and can be adapted independently of other practices, which permits incremental adoption and process improvement.

The IBM practices define not only what to do but also how to do it and who should do it. Once configured, the development process can be deployed to your project team in the form of a richly linked website. With IBM Rational tooling, each practice includes tool configuration assets to align tool behavior with the practice methods. Together with practice guidance that can be accessed in-context from the practitioner tools, this IBM solution provides on-demand assistance when and where it is required.

The IBM Practice Library covers both systems engineering and real-time and embedded software engineering. Each practice provides a proven starting point which can be adopted as it exists or configured further to meet the requirements of a particular development context. As practices are adapted to meet the requirements of your development organization, the practices can be added to the library for reuse on future projects.

The systems engineering practices currently available in the IBM Practice Library include:

- Elaborate draft systems requirements specification (SRS)
- Detailed use-case requirements analysis
- Build and validate use-cases
- Architectural analysis—key system functions
- Architectural analysis—operation based
- Architectural trade study
- Architecture design—use case based
- Architectural design—operation based
- Joint realization

The currently available real-time and embedded software engineering practices include:

- High-fidelity modeling
- Real-time architectural design
- Real-time collaborative ("mechanistic") design
- Real-time detailed design
- Model-based testing
- Safety and reliability analysis

In addition to the core practices, the IBM Practice Library also includes industry-specific practices to cover safety-critical and other standards such as DO-178B for the aerospace industry and ISO 26262 for the automotive industry.

Metrics, measurement and reporting

As each practice is deployed, its performance should be assessed; this assessment helps to ensure that process improvement objectives are met. Each practice defines the relevant metrics that enable such measurements. Measurement capabilities within the Rational solution for systems and software engineering use the Jazz Integration Architecture to automatically extract metric data from various solution components to facilitate this assessment.

These measurement capabilities can support continuous improvement of practices. Figure 2 illustrates the "assess, act, steer" approach that forms the foundation of the IBM incremental and measured improvement philosophy.

Automated measurement and reporting can also deliver other benefits within the lifecycle, improving project management by providing real-time "dashboards" of crucial project parameters and helping to meet compliance and governance requirements by automating many document production processes. By making measurement and documentation an automated by-product of the process, both efficiency and quality can be improved through the elimination of time consuming and error-prone manual data-gathering and reporting activities.

Figure 2: The IBM approach to incremental, measured process improvement

Professional services

It is true that proven practices and tools are crucial to improved delivery, yet each project has its own unique requirements and constraints. Realizing improvements at minimum risk therefore requires efficient assessment, adaption and deployment. IBM provides packaged professional services to help you to assess your needs based upon your organization's business and operational objectives. The IBM team helps define a solution roadmap for incremental, measured improvement of your systems and software delivery processes. IBM professional services can then work with you to adapt, deploy and validate the solution.

Conclusion

This paper has examined the opportunities for deploying proven practices to systems and software delivery processes. As delivery organizations face a "perfect storm" of rising complexity, increasing cost pressures, compressed development timescales and increased expectations of quality, businesses can no longer rely on "piecemeal" changes to delivery processes.

Practices provide a robust and flexible solution to these challenges which form the basis of a modular approach to deploying tooling and process solutions. By linking customizable practices, tooling and measures of effectiveness it is possible to derive a highly adaptable solution that can be deployed at minimum risk through an incremental, measured approach.

The IBM Rational solution for systems and software engineering combines world-class tooling capabilities with a broad practice library and deployment services to deliver solutions that can be tailored to meet the business and operational needs of complex systems development organizations.

For more information

To learn more about the IBM Rational solution for systems and software engineering please contact your IBM marketing representative or IBM Business Partner, or visit the following page: **ibm.com**/software/rational/workbench/systems

© Copyright IBM Corporation 2011
Route 100
Somers, NY 10589 U.S.A.

Produced in the United States of America
October 2011
All Rights Reserved

IBM, the IBM logo, ibm.com, Jazz, and Rational are 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.

[1] See Cisco technology predictions, http://www.cisco.com/web/about/ac79/docs/Top_25_Predictions_121409rev.pdf

Please Recycle