

# Business Rules and Software Requirements



## Applies to:

SAP NetWeaver Composition Environment 7.1.1

## Summary

This document describes how business rules and software requirements are related to each other. It suggests agile methods for managing business rules.

**Author:** Rajagopalan Narayanan

**Company:** SAP Labs India Pvt. Ltd.

**Created on:** 19 February 2008

## Author Bio



Rajagopalan Narayanan is a Product Manager in the SAP NetWeaver Product Management Business Process Management team. His focus area is Business Rules Management. He has many years of experience with Business Rules Management.

## Table of Contents

About this Document .....	3
Software requirements and business rules – Definitions .....	3
How are business rules and software requirements related then? .....	3
Business rules and software requirements – Different Owners .....	4
Business rules and software requirements – Managing Change .....	4
The Paradox - business rules vs. software requirements .....	5
Solving the paradox .....	5
Requirements Gathering Phase .....	6
Application Design Phase .....	6
Application Implementation Phase .....	6
Application Testing Phase .....	6
System Maintenance Phase .....	6
The value proposition of business rules management functionality .....	7
Summary .....	7
Copyright .....	8

## About this Document

One of the most common mistakes of developers and architects of software systems is to assume that business rules residing in applications are a part of the system software requirements. As a result, business rules are handled and managed as software requirements.

Nothing can be farther from the truth. This document aims to convey why this is an incorrect approach and goes ahead with suggesting more agile methods for managing business rules.

[Back to top](#)

## Software requirements and business rules – Definitions

So, what are **software requirements**? According to 'Wikipedia.org' software requirements:

- Describe the behavior of the software system to be developed
- Include sets of use cases that describe all user interactions with the software system
- Include non-functional requirements like quality, performance, design constraints, etc.

Software developers and architects use these requirements as inputs for design and development activities.

**Business rules** on the other hand describe or represent constraints on the behavior of the business. Business rules comprise the core business logic of each organization, guide and control all basic business processes that form the backbone of any business transaction.

When working with business rules developers have to take into consideration that such rules are inherent in:

- Corporate charters
- Management practices and regulatory forces
- Human resources management
- Marketing strategies
- Pricing policies
- Products and services offerings
- Customer relationship practices

Business rules are the most dynamic component of any application. Therefore their constant and correct identification and externalization improve the organization's adaptability to industry changes and competition.

[Back to top](#)

## How are business rules and software requirements related then?

To understand the relationship, let's examine an 'Applicant Eligibility Service' example in an Insurance industry scenario. This service is used to validate specified applicant's eligibility for a specified product or a set of products, depending on the design.

In this example, the requirements would describe the following:

- Service Signature.
- Deployment Platform.
- Performance Considerations.

And the business rules of the specific unit / organization would describe:

- Criterion by which the applicants are determined to be eligible.
- Which geographical locations are applicants accepted from?
- Credit Score Requirements.
- Minimum asset requirements, etc.

As you can see in this example, the requirements for the service and the business rules that define what this service must deliver are two completely different things.

The correct way for modeling requirements in such scenarios is to reference appropriate business rules from software requirements. To be able to do this effectively, it is important to understand that business rules and software requirements must be thought of and managed independently.

[Back to top](#)

## Business rules and software requirements – Different Owners

Software requirements describe software systems behaviors and are therefore owned by IT personnel – e.g. developers, IT managers and software architects.

Business rules, on the other hand, comprise all policies and guidelines in force, for a certain time period, that describe how each business function needs to be performed.

Going back to the 'Applicant Eligibility Service' example, the IT unit is responsible and accountable for the software requirements. On the other hand, the eligibility criteria for all different insurance products are introduced by the business unit and IT is not held accountable for the correctness of criteria definitions, but only for the correctness of criteria implementation.

Business users are accountable for business rules definition, however the way these rules are realized in IT systems prevents participation of business users into the rule implementation process. Usually business users are not able to take real ownership of the business rules that they are defining and executing. It is obvious that there is a challenge raising, because although IT unit is not responsible for the correct definition of eligibility criterion (business rules), they may however be accountable for the correct implementation of these rules, which they do not necessarily understand in their completeness.

IT and Business units in many organizations have to rely on traditional and in most cases ineffective methods of collaboration in order to handle ownership issues, which usually leads to incorrect rule implementations, time and resources losses. Now how can IT and Business units work together to resolve this?

[Back to top](#)

## Business rules and software requirements – Managing Change

Everybody knows that software systems once built need continuous maintenance. Change requests come for a variety of reasons and from different sources. It is of vital importance to understand that business rules and software requirements have completely different change cycles and changes occur for completely different reasons.

Changes to software could come in the form of:

- Software bug fixes
- Performance enhancements
- UI improvements
- Upgrades, etc.

Changes to business rules are always initiated by business operational needs. Examples of change requests to business rules include:

- New product introduction (Eligibility rules changes)
- New Tax regulations (changes in Government regulations)
- New pricing rules (competitive pressure)
- New interest rate sheets (in mortgage industry, from capital market)

So, how are changes to business rules different from changes to software requirements? Here are some more details:

- Business rules may change much faster than software in many different cases, e.g. in the mortgage industry, sometimes interest rate sheets change up to 3-4 times in a day.
- Business users involvement is required for changing business rules, as business users are owning business rules.

[Back to top](#)

## The Paradox - business rules vs. software requirements

Here is a short summary of what we have covered so far:

- Business rules are owned by business users, while requirements are owned by IT units.
- Business rules change lifecycle differs considerable from requirements change lifecycles.
- Requirements must always reference rules, never embed them.
- Business rules cannot be managed using conventional requirements management techniques.

We have just outlined what is called the “Rules vs. Requirements Paradox”. This is a paradox because:

- Business rules are owned by business users, but their implementation is controlled by IT.
- Business users are responsible for the correct definition and implementation of business rules, but they cannot enforce this in software systems.

[Back to top](#)

## Solving the paradox

Any good solution for this paradox should take into consideration all challenges identified so far in this analysis. Business rules management components offer a clean, scalable and business-friendly approach towards solving these issues.

Here is a more detailed description of a solution employing business rules management functionality:

## Requirements Gathering Phase

1. Identify all critical and volatile business rules in application(s).
2. Identify all those requirements/decision points in application(s) that must reference business rules. Per example, if the requirement is to build a 'claim settlement service', then business rules should be 'claim settlement rules'.
3. Involve business users in identifying and defining correctly all those business rules.

## Application Design Phase

1. Once the integration touch points have been identified, business rules design and application design can be initiated in parallel tracks.
2. This works well, because the application will no longer implement business rules in code or stored procedures, but will defer to the business rules management functionality for rule correctness and availability validation.

## Application Implementation Phase

1. Implement your business rules using the business rules management functionality - Many rules like 'pricing', 'underwriting', can be even written directly by business users, because these are always in the form of tabular rules called *decision tables*. Business Analysts can be trained to write down the rest of the rules.
2. Application development and rule development are truly concurrent, because clear integration points have been previously defined.
3. The application will call a *rule engine* which is responsible for applying rules on business data and deliver rule based decisions or results.

## Application Testing Phase

1. Different test cases typically can be sourced by business users or analysts.
2. Business rules can be tested rigorously, independent of the application, in the best case using test cases captured and provided by business users.
3. Use same patterns to perform regression testing for changes.
4. The test cases can be created and captured independently by different business teams.

Rules testing performed independently of the application testing eliminates the need to go through the whole application testing cycle for business policy/rule changes. Per example: rate sheets in a mortgage company can be introduced into the business rules management functionality directly and tested for correctness.

## System Maintenance Phase

1. Because rules can be captured, implemented and tested independently of the application, now business rules lifecycle can follow its naturally different path from the application lifecycle.
2. Business rules changes are business driven and not IT driven. Rules repositories allow regular versioning and also enable specialized use cases like 'historical rules invocation'.
3. Once the major rules are set up by the rules development teams, regular rules changes / updates can be performed by business users. IT professionals can handle more complex 'If – Then' rules that might be difficult for business teams to implement.

[Back to top](#)

## The value proposition of business rules management functionality

Based on the details outlined before, the value proposition of implementing business rules management functionality can be highlighted as follows:

- A clear separation between business rules (business requirements) and software requirements is achieved.
- Translation errors and complexity is reduced by enabling Business Analysts to participate in business rules implementation and change management stages.
- Representing rules in user-friendly formats like 'If – Then' and decision tables enables business users and analysts to work with rules directly. That helps to eliminate errors and collaboration delays, delivering increased alignment on top.
- Clear responsibilities are set streamlining effective collaborative processes - business users take real ownership of business rules definitions and application developers are no longer held accountable for definitions correctness.
- Concurrent development and testing is enabled leading to considerable turnaround times reduction.

[Back to top](#)

## Summary

Business rules are different than software requirements. Business rules represent different business requirements on how business decisions need to be delivered and are owned by business users. At the other hand, software requirements, in general describe software system behavior and the way users interact with the system. Software requirements are usually created and used by technical users.

Employing business rules management functionality enables organizations to clearly separate and handle individually issues arising out of the outlined differences. Therefore, leveraging business rules management organizations can step forward in building agile business systems which clearly and productively involve business users in establishing, managing, owning and executing informed decisions.

[Back to top](#)

## Copyright

© 2008 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.