

# MKS Integrity Suite 2005

Reviewed by  
Michael Sayko

“The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirements...Therefore the most important function that software builders do for their clients is the iterative extraction and refinement of the product requirements.”

– Fred Brooks in *No Silver Bullet*  
– *Essence and Accident in Software Engineering*

Experienced Java developers recognize that capturing and refining the requirements is one of the most important, yet difficult, parts of building a software system. An iterative software development process that incorporates the right tools can facilitate effective requirements management.

## Product Background

MKS added requirements management to their software configuration management (SCM) product offerings with the recent release of Integrity Suite 2005. Although the enhancements to Integrity Suite encompass more than requirements management, this review focuses on the extensions to support requirements management since these extensions are the distinguishing features of the 2005 product.

Rather than developing a requirements management tool from scratch, or acquiring an existing product, MKS incorporated new features into their change management tool, Integrity Manager. These features allow Integrity Manager to collect and manage requirements. In addition, Integrity Manager can be used with Source Integrity to link requirements with source code files that are placed under version control.

## Product Architecture

MKS Integrity Suite 2005 is a J2EE application comprised of Integrity Server and Integrity Client.

Integrity Server manages the process items and source code files that reside on the server. It runs under an application server packaged and installed with the product. It uses FLEXlm as the license manager.

Integrity Client consists of three logical pieces: Source Integrity, Integrity Manager, and the Administration Client. Source Integrity is the version control interface, while Integrity Manager facilitates process and workflow management activities. Both Source Integrity and Integrity Manager share the same GUI. In a separate GUI, the Administration Client allows an administrator to perform common administrative tasks on the Integrity Server.

## MKS

410 Albert Street  
Waterloo, ON  
N2L 3V3 Canada  
**Phone:** 800 265-2797  
**Fax:** 519 884-8861  
**Web:** [www.mks.com](http://www.mks.com)

## Test Platform

Dell Inspiron 8100, 1.2GHz Intel Pentium III mobile, 512MB RAM, 40GB disk, Windows XP Professional Service Pack 2

## Specifications Platforms

Server and Client: Windows NT/2000/XP, Unix, Linux

**Pricing:** Pricing for 10 users of the MKS Integrity Suite (including MKS Source Integrity Enterprise, MKS Integrity Manager, Integrity Server and MKS Requirements) is \$37,000

## NEED HEAD SHOT

**Michael Sayko** is a software configuration management consultant based in Austin, Texas. He is experienced with the practice of software configuration management from having served as a configuration manager on large, fast-paced software projects.

[mss@acm.org](mailto:mss@acm.org)

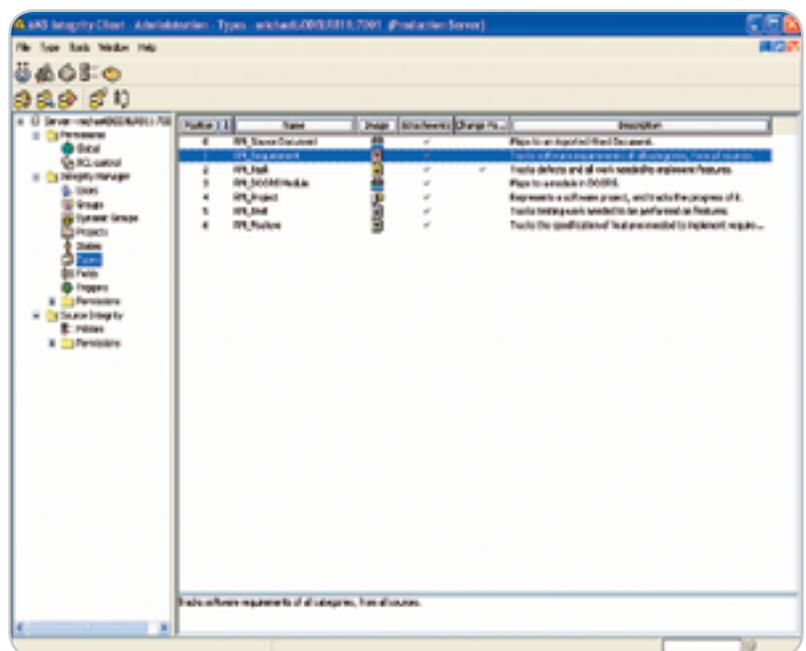


Figure 1 Requirements types in the Administration Client

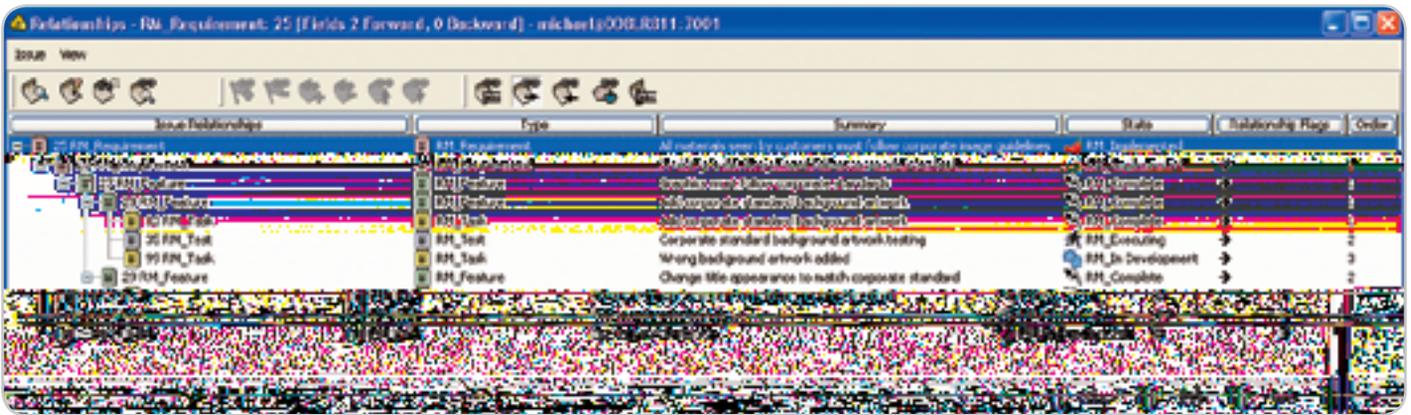


Figure 2 Relationship view

## Installation

The Integrity Suite is distributed on two CDs. Installing the software is straightforward. For this evaluation, I installed both the Integrity Server and the Integrity Client on a 1.2GHz Wintel Notebook computer with 512MB RAM. I installed the Integrity Server using the embedded PointBase database. I configured the Integrity Server to use the flat file authentication scheme.

In addition to the embedded PointBase database, Integrity Server is designed to work with Oracle, MS SQL Server, and DB2 databases. Integrity Server stores process item data, including requirements, in the database. When using one of the supported commercial databases, files checked into Source Integrity can also be stored in the database. During the installation of Integrity Server, the administrator needs to decide where version controlled files will be stored. If the database option is not selected, files placed under version control with Source Integrity are stored on a file system in Revision Control System (RCS) format.

## MKS Requirements 2005

After installing the Integrity Suite, I proceeded to start the requirements management component. I soon realized that the requirements management component is really Integrity Manager. MKS added the following enhancements to Integrity Manager so that it could be used as both a requirements management repository and engine:

- Named issue relationships that link one issue to another
- A relationship view to visualize how issues are related

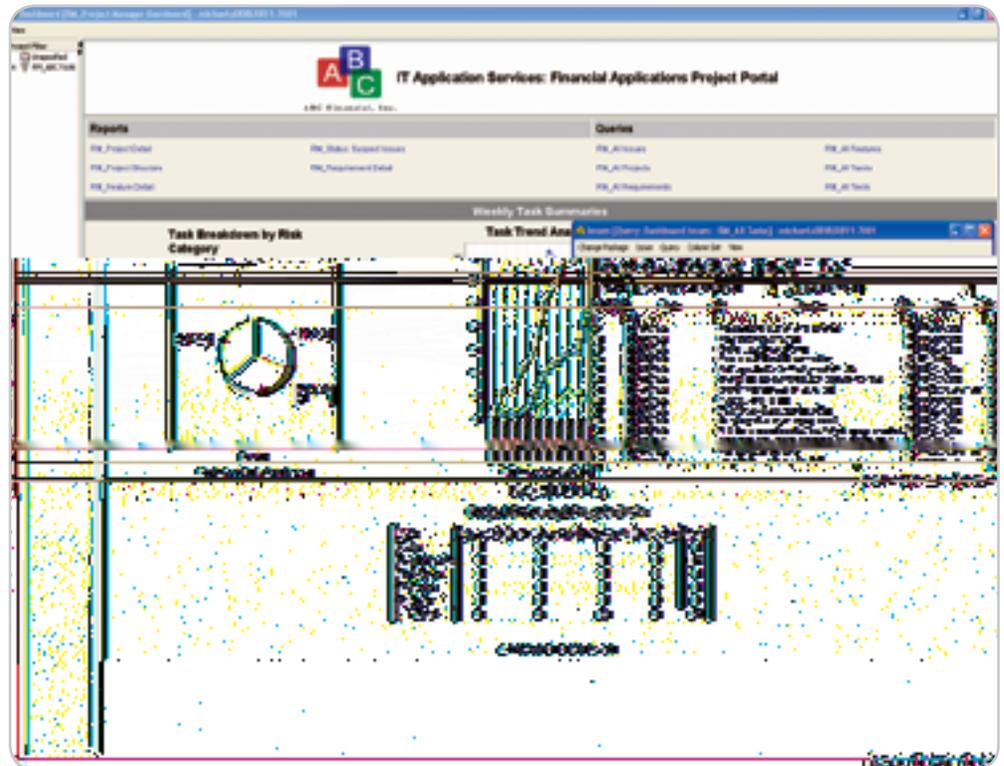


Figure 3 Project management dashboard

- Suspect links that control changes to requirements
- Integration with Microsoft Word to capture requirements defined in a Word document
- Integration with Telelogic DOORS to capture requirements defined in a DOORS module
- A process template consisting of five requirements-related processes

The *process template*, also called a *solution*, is named MKS Requirements 2005. While the process template is not required for requirements management, MKS created this template so their customers could visualize how to use the enhancements in Integrity

Manager to manage requirements. Although MKS describes the process template as “an illustration of MKS Integrity Suite’s requirements management capabilities,” I suspect that customers will want to use the template’s contents to model their requirements management process. Of course, customers can modify the process items defined by the template to address their specific needs.

## Using Issues to Manage Requirements

To understand how requirements are managed by MKS Requirements 2005, you need to recognize that Integrity Manager uses *issues* to model process items. Each issue consists of *fields* that store data. An issue

transitions through a series of *states* to model a process workflow. Each issue is defined by its *type*. When a user creates issues from a type, the user can locate these issues with a *query*. A user-defined query simply selects and lists issues that meet certain criteria.

An administrator defines fields, states, and types using the Administration Client. A user then creates an issue (i.e., an instance of a type) and a query using Integrity Manager. Requirements, features, and tasks are examples of issues that can be managed by Integrity Manager. Prior to the release of Integrity Suite 2005, users of Integrity Manager created issues, such as change requests, to track software development activities like fixing a defect or adding an enhancement.

After enhancing Integrity Manager to support requirements management, MKS developed the Requirements 2005 process template to model requirements artifacts. The template consists of seven types (Project, Requirement, Source Document, DOORS Module, Feature, Task, and Test) that take advantage of the enhancements to Integrity Manager. Keep in mind that these types are administrator defined, rather than a base feature of Integrity Manager. For this reason, they can be used as is or modified through the Administration Client. Figure 1 shows how to access these types using the Administration Client.

### Tracking Requirements, Analyzing the Impact of Changes, and Visualizing Project Status

The strength of MKS Requirements 2005 is the way in which it allows users to track requirements, analyze the impact of the inevitable changes to requirements, and visualize the status of a project.

When using the issues in the process template, a *requirement* is defined by *features* that are implemented as *tasks* and then validated through *tests*. Figure 2 depicts a chain of relationships from requirements to features to tasks. Although not shown in this figure, *change packages* link tasks to the source code files that implement them. This is the premise behind *task based development*. Each check-in to the version control system

is associated with a fine-grained development task. One benefit of task based development is that builds of the software application can be described by the features implemented, rather than just the source files modified. This makes the construction of a software application meaningful to a broader audience. Project managers, software testers, and end users can identify the features and fixes incorporated into every build of the software application. MKS Requirements 2005 extends task-based development to *requirements-based development* because it maintains relationships from requirements to features to tasks to the versions of source code files that implement the requirements. Now every build can also be described by the requirements that it implements.

Advocates of agile software development recognize that requirements evolve from their inception to their realization in working software. MKS Requirements 2005 supports requirements changes through *suspect links*. Suspect links are flags on relationship fields that are triggered when a requirement is changed. This allows dependent features, tasks, and tests to be marked as needing to be reviewed for the impact of a requirements change.

Finally, MKS Requirements 2005 allows team members to view the status of project using a *project management dashboard*. The dashboard is a real-time view of project data that provides

for interactive drill down to details. Figure 3 shows a project management dashboard with project status graphs and links to reports and queries.

### Summary: Advantages of an Integrated Requirements Management Solution

The manner in which MKS integrated requirements management capabilities into Integrity Manager demonstrates the flexibility and extensibility of this process modeling and workflow management tool. Requirements, features, and tasks are managed like any other issue (i.e., process item) stored in the Integrity Manager database. One clear benefit of building requirements management artifacts from issues is that the artifacts can be linked to affected source code files using change packages. No integration effort is required to facilitate traceability of requirements and change management from the same repository.

While the requirements management components, workflows, and process rules described in the product documentation may appear to represent a rigid model, a closer look at the process template shows otherwise. The Integrity Manager documentation contains detailed, but easy to follow instructions for extending any Integrity Manager components, including the issues in the process template. By following these instructions, the components in the process template can be tailored, in a straightforward manner, using the Integrity Manager GUI. ☛

### JDJ Product Snapshot

**Target Audience:** All members of the software development team including software configuration managers, developers, business analysts, and project managers.

**Level:** All levels, from beginner to expert.

**Pros:**

- No integration required. MKS Requirements 2005 is a complete and self-contained requirements management solution because it's built with the process modeling and workflow management tool Integrity Manager. Contrast the approach followed by MKS to the model used by other tool vendors. Other requirements management products must be integrated with a software configuration management tool to provide traceability from requirements to source code.
- Facilitates requirements based development. Relationships coupled with change packages link requirements to the source code files that implement them. This also provides the traceability needed for audits, including software baseline audits.

**Cons:**

- First generation product – Even though MKS Requirements 2005 is built with the mature Integrity Manager, potential enhancements will be identified as MKS Requirements 2005 is used in production environments.