

What Do IBM i Developers Want Out of Their ALM Software?



A Technical White Paper by Marty Acks - IBM i Product Manager for MKS

MKS

Executive Summary

This white paper from MKS will look at frequent activities of a developer and how many of these activities can be faster and more accurate by using the ALM software, MKS Integrity for IBM i.

Application Lifecycle Management (ALM) refers to the discipline of managing software changes from concept to delivery. Poorly developed or implemented, these tools can be a developer's nightmare. Properly built and implemented these tools can make developers even more productive and leave them to do what they do best; build applications for their users or customers. This white paper will focus on the developer experience and explain how MKS Integrity for IBM i can simplify many common development activities.

In this white paper we'll review some of the common activities of a software developer:

- > Fielding questions from users
- > Reviewing assignments
- > Host-based development
- > RDI-based development
- > Checking in work
- > Closing the loop with users
- > Rolling back a change

Fielding Questions from Users

In many companies various ad hoc methods are used by end users to communicate with development teams including phone, e-mail, instant messaging, and camping out at your desk. While these may result in immediate gratification for end users, these methods can be rather disruptive of development activities.

MKS Integrity for IBM i offers easy-to-use web based entry IT Help desk capabilities to allow end users to log problems with the ability to notify various parties (developers, business analysts, team leads) based on the nature and severity of the ticket being raised. Should this ticket actually result in logging an issue that requires a program change, the same issue can be used by the developer to track their software changes as will be shown later. Additionally, knowledgebase articles can be created to allow common problems and questions to be resolved more quickly. These IT helpdesk features benefit the developer by allowing them to review problems tickets in orderly fashion and avoid the hit-or-miss results of ad hoc approaches.

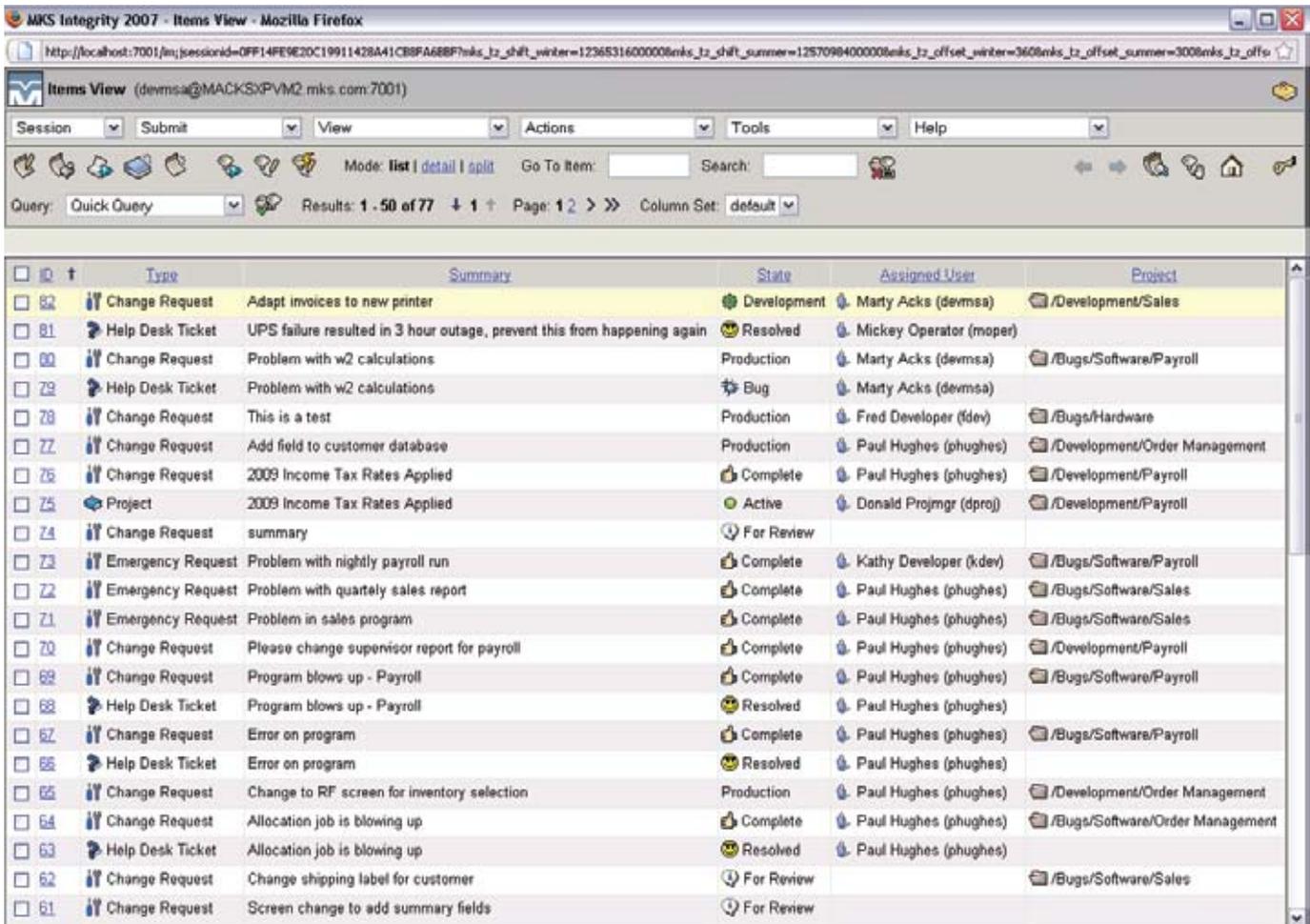


Figure 1 - End users can easily submit tickets for developer review

Reviewing Assignments

In many companies, meetings are called frequently to review the same set of open issues and review their status. E-mail chains that may get too few or the wrong people involved can add to the confusion. Informal requirements and design meetings with results that are not tracked in a central location can add the miscommunication. While it may be comforting to just write off communication errors as “not my fault,” in the end, no one is happy.

MKS Integrity for IBM i allows you to review your assignments from your desktop in the way you find most useful. As mentioned earlier, a browser interface is available. Better yet, you can also review new or reprioritized assignments directly from the MKS Integrity for IBM i Workbench, WDSC or RDi plug-ins, or within other development environments including Microsoft Visual Studio, other Eclipse-based environments, and CA 2E. As new assignments are made or priorities change you can opt to be notified by e-mail or by a simple press or click of a refresh option to see an up-to-date list.

As work gets assigned you now have one place to go to determine what really needs to be done. For larger changes you can utilize electronic signatures to insure everyone is in agreement on what is to be changed. Supporting materials can be attached to the change request. You can also configure MKS Integrity with rich text information to track detailed requirements and design decisions in one unified location.

Using these capabilities of MKS Integrity for IBM i you can reduce reworking of changes due to unclear requirements and more closely build changes to the exact needs of your users.

Repository-Based Application Development

Many IBM i developers work in either PDM or RDi. Both of these environments provide strong source editing, and the ability to easily work with lists of source members and objects. However, they leave many common development activities to be done manually.

At the heart of MKS Integrity for IBM i is a repository that makes us more aware of your application development practices and procedures. This information about your applications is entered one time when enrolling an application in MKS Integrity for IBM i. By defining these various characteristics of your applications to MKS Integrity for IBM i, you will have fewer repetitive decisions to make less data to key resulting in fewer mistakes.

For example, at checkout we can bring up in a single list all members in that application, we can drill through multiple libraries and multiple source files to find the current source code, allow choice from a list of source members when performing parallel development, and enforce naming standards on new items. During development, correct compiling order is automatically determined along with retention of existing characteristics. The relationships between objects is understood allowing all the appropriate objects to be easily selected to be worked on. When checking in a change, MKS Integrity for IBM i insures the source and objects are returned to their rightful location while optionally entering any special instructions and commands that are needed at the same time.

Similar rules apply to new and deleted code. New code should be placed in the correct location. Deleted code should be taken out of your normal production view of the application but still kept available through the archives. You spend more time programming while preventing deployment issues created by accidentally misplacing your source code and objects.

As you modernize applications you may have directory-based files that reside in the IFS or other network servers. These files are managed side by side with native source and objects in MKS Integrity for IBM i. This eliminates the need to learn an entire new command set for move, copy, deleting, and viewing these types of stream files.

These are just a few of the dozens of developer-oriented feature that replace manually typing commands, searching multiple locations, and the manual copy and moving of source and objects around. All the activities are done by simple gestures directly within MKS Integrity for IBM i. And when you are done with a change, let MKS Integrity for IBM i do the rest of the work, which is of boring, repetitive, error-prone mechanical tasks. You'll find it's a great improvement to be able to apply the time you currently spend sweating deployment details on developing code.

Host-Based Development

Many developers still develop using host-based Programming Development Manager (PDM) to manage source and objects and edit their source with SEU. PDM's object and member lists make it easy to work with multiple items at a time. User-defined options and repeat function key add to developer productivity. But beyond that, PDM is poorly equipped to really manage development.

MKS Integrity for IBM i incorporates the Developer Workbench that provides a PDM-like experience to start but with much richer functionality due the application awareness that comes from our e its repository. Key features of the Developer Workbench include:

- PDM-like user interface as a starting point – familiar option numbers and functions keys are used
- Single level application view – all source/objects for an application on a single view with advanced filters available
- Viewing and filtering work by task – no exiting of the Workbench is required to view your assignments
- Impact aware – relationships between objects can easily be drilled into
- In process aware – clear indication of who is working and what and why across shared and personal development libraries
- Parallel development – You can select the appropriate copy of the source to initiate well tracked concurrent changes
- Archives – Unlimited number of source archives can be viewed; visual differencing and annotated views highlight what was changed and why
- Smart compile – Mass compiling is done in the correct order and based on existing values (if not prompted) eliminating manual overrides or creation of build scripts
- One step check out and check in
- Easily move from one set of member and objects to another via named filters
- User defined function keys in addition to enhanced PDM-style user defined options allow further customization of the user interface by developers

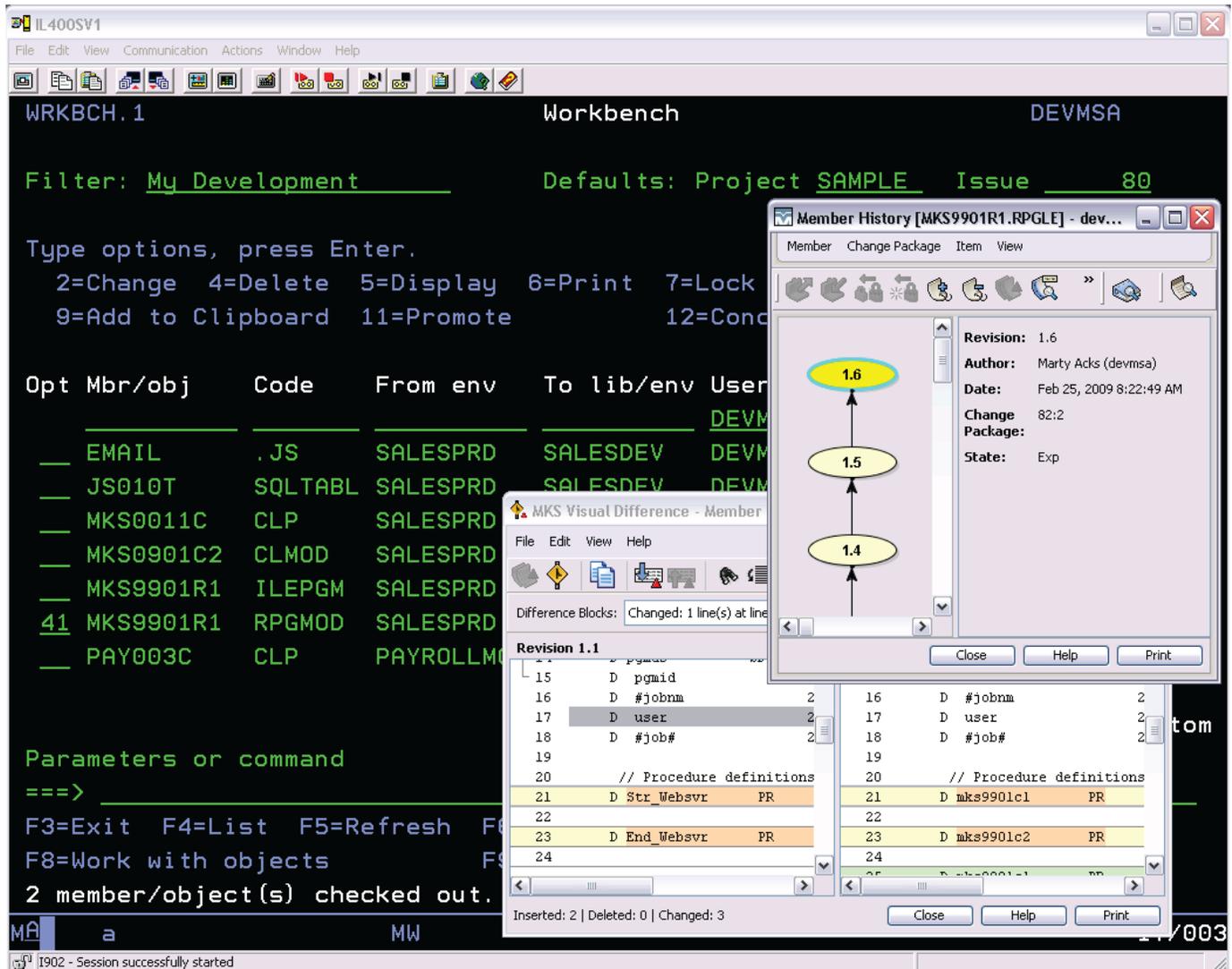


Figure 2 - MKS Integrity for IBM i Workbench includes built-in visual differencing

RD*i*-Based Development

We also see many developers utilizing Rational Developer for i (RD*i*) and also continue to use its predecessor WebSphere Development Studio Client for System i (WDSC). Both products are built on the Eclipse platform. One of the many features of RD*i* is to build host-based application using a GUI-based development environment. IBM i developers using this platform have similar challenges to those of the PDM developer. While RD*i* adds a greatly enhanced editing experience, source comprehension features, and a visual debugger, it faces many of the same shortcomings as PDM for managing software changes.

Much like our approach on the host with our Developer Workbench, we have integrated directly into the Remote System Explore (RSE) perspective of RD*i*. Rather than relying on disconnected projects or totally different perspective, we have placed a subsystem inside RSE giving a more nature and familiar way of working to those already using RD*i*. MKS Integrity

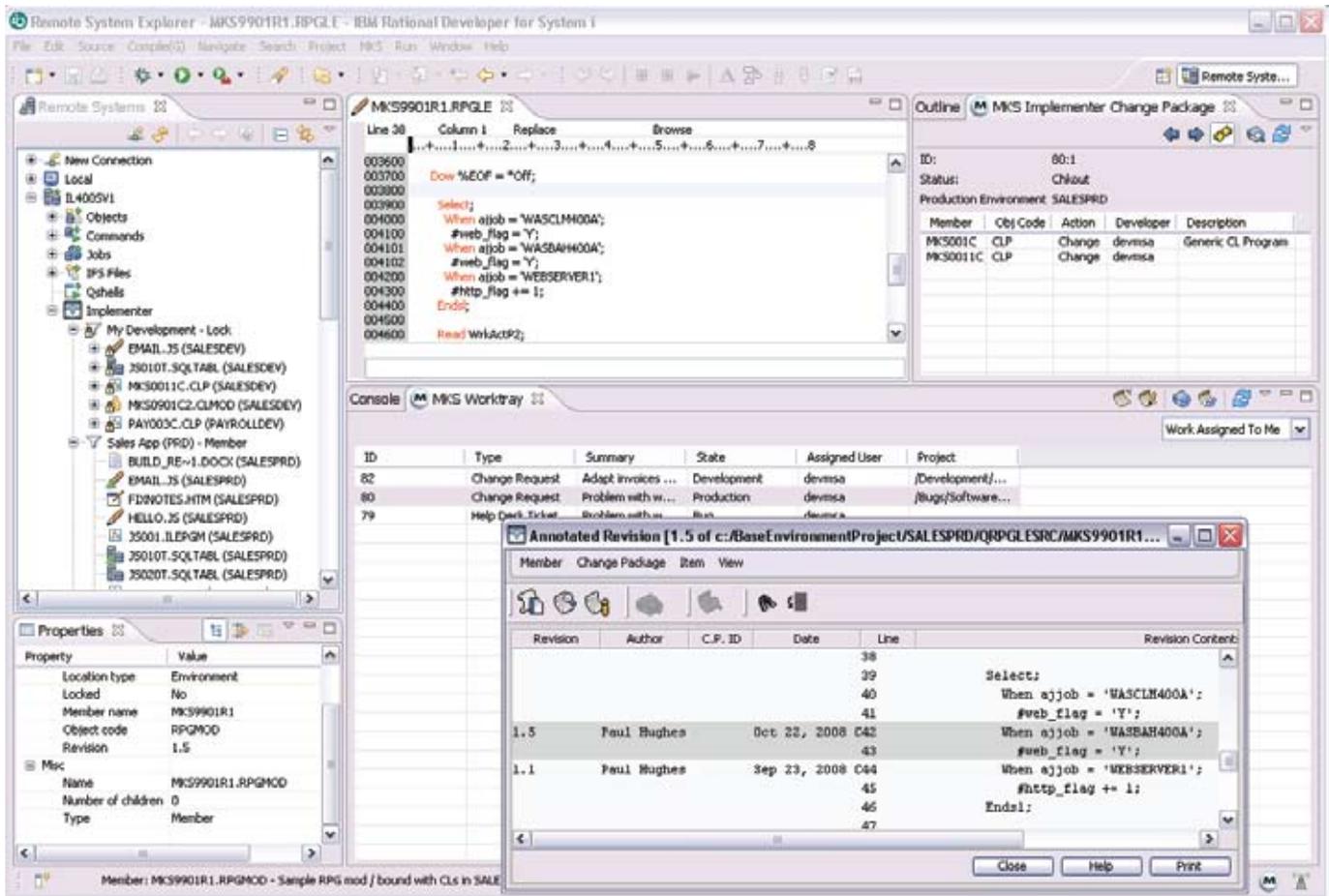


Figure 3 - The MKS RDI plug-in includes advanced features such as an annotated revisions view

form IBM i also uses a filter oriented user interface, which allow you to easily define groups of source and objects into logical sets as you see fit. While the filters are much more extensive than those available in RSE for standard Member and Object filters, the underlying approach we have taken will be familiar to RSE users.

Similar capabilities exist for RDi development as with host-based development. Since we rely on a unified repository, if you are just starting to transition to RDi you can move back and forth between the RDi and host panels with ease. Even the filtered views you define in RDi can be shared with the Developer Workbench.

And if you want to work offline in some instances, we allow you to easily add your source to IBM i projects (also known as iSeries projects in WDSC) for offline development. MKS Integrity for IBM i also provides a separate a Worktray in a separate perspective (see the bottom right pane in Figure 3.) that shows the open work assigned to you and allows you to filter to any view of items you desire. You can review supporting information for a change you are working on and update the issue directly from here. (See the top right pane in Figure 3 above)

In summary, if you choose to use RDi or WDSC we have integrated deep into the RSE perspective to allow you to stay within RSE to perform your common developer activities.

Checking in Work

Once change is complete, it needs to be installed properly. While the developer is typically not responsible for the deployment to production, they do often check in the source code and deployment to the first testing region(s) for the application. Key features for checking in and installing a change into a test environment include:

- > Compile in logically correct order
- > Staging of source and objects to eliminate partial promotions
- > No manual copying or build scripts required
- > Ownership and authorities automatically set
- > Elimination of library list errors
- > Inclusion of all impacted objects across all applications to prevent level checks and application failures
- > Scheduling of changes
- > Advanced support for ILE, IFS, SQL, and database files
- > Deployment of partial or complete sets of data
- > Deployment to a predictable location or set of locations
- > Dashboard for monitoring progress

When these changes need to be deployed to other servers or partitions MKS Integrity for IBM i provides even further automation. In today's world where physical access to QA and production systems is becoming increasingly restricted these capabilities are even more important. Key features for deploying changes to other server or partitions include:

- > Automatic save changes to save files
- > FTP of changes to target systems/partitions
- > Restore changes (to staging location)
- > Separate distribution and installation steps, if desired
- > Optionally, compile on target system/partition
- > Previous IBM i operating system version support for target system/partition
- > Source and/or objects archived on the target system/partition for fast rollback when needed
- > Adapts to network security policies such as SSL

MKS Integrity for IBM i simplifies check in and deployment to just selecting the issue or selecting a specific list of objects to be deployed. This tends to be one of the highest risk areas when done manually. These features of MKS Integrity for IBM i gives you back the time you currently spend sweating these details so that you can do better development and more of it.

Closing the Loop with Users

Those in your end-user community legitimately need to know when a change is accepted, rejected, underway, being tested, and/or finally implemented. As a programmer, you tend not to be opposed to the users being in the loop (hopefully!), but you would like this to occur with minimal effort in your part.

With MKS Integrity for IBM i you can enter notes targeting your end user when appropriate for you: when starting, when in the midst of, or when completing a change. The users can be notified at the appropriate times as a change progresses through to the live application. This side effect can greatly reduce the ad-hoc calls and walk-bys from your users and other IT staff. This improved communication with the users (which you need to participate in only marginally) can turn visits outside of IT into “attaboy” accolades for the some feature or correction instead of those glances that seem to ask “What do you actually do for the company?”

Rolling Back a Change

In today's world of applications being exposed to the web, and global deployments that demand 24 by 7 availability, any failures in production must be dealt with quickly. Typically, what you need to do right away, is place the previous version of a program back into the live system to buy some breathing room while a full correction is formulated. This used to mean you have to copy the source or objects out of the live system when you initially implement each change. Then, while moving the old items out and moving the news ones in, you have to check for any dependencies. What if the defect was subtle and you really need, for example, to get the great-grandfather version of something back quickly? That can turn into a little research project of its own.

The archival process built into MKS Integrity for IBM i relies on a secure formal archive mechanism that allows easy roll back of a single change or set of specific items in a change without worry, whether recent or not, giving you the ability to reliably go back to the way the program worked before. MKS Integrity for IBM i can turn a potential crisis into a minor occurrence.

In Closing

This white paper has demonstrated a variety of ways in which MKS Integrity for IBM i can simplify development for IBM i software developers. Hopefully, we have dispelled misconceptions about ALM just being a set of controls that get in the way of real programming. If you add up just a few of the benefits mentioned above, you will find using the developer-oriented features in MKS Integrity for IBM i can allow you to work more effectively and accurately within a unified framework that also provides broader overall ALM benefits to the entire IT organization.

MKS Headquarters

North America

1 800 613 7535

UK & Northern Europe

44 (0) 1483 733900

Central & Southern Europe

49 (0) 711 3517750

Denmark

+45 4420 9831

Singapore

65 6732 8768

Japan

+81 3 5422 9503

sales@mks.com

For more information

visit **mks.com**

MKS