

IBM Podcast

MATHENY: Welcome to this IBM Podcast, Reduce, Reuse, Recycle, Jazz-based Asset Management Made Easy. I'm Angelique Matheny with IBM. The newest release of Rational Asset Manager -- or, RAM -- has been re-factored to the Jazz platform to improve users' collaboration in the governance of software assets.

In this podcast, Chief Architect Grant Larsen explores how organizations are using RAM to harvest, locate and govern any software asset including reusable services, approved open source components and purchased commercial assets.

So, let me properly introduce our presenter. Grant Larsen, Chief Architect, Asset Management for IBM Rational Software.

Grant works with customers to solve software delivery challenges using Rational Asset Manager and other Jazz based products. Grant drives the asset management strategies through process, standards, tooling and assets to leverage software development and delivery investments. Hi Grant, welcome to the podcast. Thanks for joining us.

LARSON: Thank you very much.

MATHENY: Grant, let's start with this. What are some lessons learned from customer experiences with implementing

organization wide reuse efforts?

LARSON: We've seen, typically, there are four major steps to rolling out and adopting asset management. The first of those is storing and organizing. And by that what we mean is, typically organizations will start with getting their arms around what is it they have today. And as they do that, we find they start to discover redundancy and inconsistency, and a lot of them start taking out some of that cost right away in the business. That's basically where they're starting to capture the assets that they have.

The second we've seen is around customizing the workflow and the policies. And here the idea is that once they've gotten some idea of what it is they have that they then begin to put in place some of the automation around workflow and states and lifecycles that the assets will go through. And put automation in place so that the tooling, and so that RAM is actually examining and verifying the nature of that content.

The third thing we've seen is around impact assessment. Again, the idea there is once they've understood what it is they have and then they started to customized workflows and policies, then they're able to start asking questions like, if I were to make a change to this, what would be impacted or who would be impacted?

The fourth step we've seen is around auditing and reporting.

Once they have those first three in place they're able to start asking questions like, what value does this asset or these assets bring to me or to the enterprise? Now, certainly, it's not water flow in the sense that you only go from one to the next, you actually tend to iterate across all of these. But in terms of the flow, it tends to go from one to four and then iterate back again.

Now, in 2009, this year we queried multiple customers to discover, what were their lessons learned? And some of them included the following, and I'll mention five or six here. One of those we learned was never to roll out RAM and asset management as a big bang.

So, the idea here is that you define your information model, the shapes of your assets, the things you're going to work with. You roll it out and you test it on a target audience, and then you iterate it. And that's actually how we rolled it out in IBM. We ended up rolling it out to a certain community, a set of communities and then learned some lessons from that and iterate. The customers that have done that have had more successful rollout and adoptions we've seen.

The second lesson learned, they've told us, is consider

using what we might refer to as an onboarding server. And by that I mean have a RAM deployment that's a sandbox. Give a place for people to come in, you know, play with it, get familiar with it.

Don't worry about making mistakes or hurting anything. And everyone knows that at some point you're going to come back and scrape it clean and start over again. But it's worth its time and effort to have that. The third lesson learned that we heard is, configure RAM always with the perspective of the consumer.

And the point here is that you want to be able to let people discover things from their context. And sometimes we struggle with that in our software organizations. We'll always configure things from the perspective of the developer or the producer or the engineer. And in fact, what we need to do is we need to allow someone to find what they're looking for in two or three clicks.

And so, you always want to put on your head the head of the developer. If I was the developer or the tester or whoever it is I'm trying to represent as the consumer, what is that person thinking at this point in time in the software delivery? And that starts to guide you on how you categorize and tag and classify it and so forth.

The fourth lesson that we've heard which I've seen quite a bit amount is that much of our content in our enterprises is distributed across teams and geographies and servers and such, and part of adopting an asset management strategy is to examine where that content will live now and in the future.

And sometimes we think, well, if I'm going to go do asset management and try to reuse things, I need to collect everything into one place. And that's actually not the case. RAM will allow that content to live where it exists and allow adoptions, strategies over time to sunset existing tools, or homemade repositories, or whatever it is.

Or maybe it's never sunset, you know, you'll always have the content sitting in some other location. and that's okay. So, you're able to make storage decisions and let content either be managed within RAM or managed remotely. So having those discussions early on, but allowing for that flexibility was another key lesson.

And the final lesson learned that I'll mention here was around adoption. And it seems the top-down management support is critical while still allowing for bottom up acceptance and roll out. And each company's culture varies, of course, as well as each community within a company's culture varies, obviously.

So, allowing a variety of governance strategies from informal to highly structured tends to work best. Watch and see how people use it and then allow for modifications. So, those are some of the key lessons that customers shared with us this year.

MATHENY: Grant, can you describe how RAM compares or contrasts with other collaboration technologies?

LARSON: Yes. There's several key things here that we've seen with customers. Collaboration tools like Lotus Connections or in some cases we've seen SharePoint or other such product software. Those certainly provide project level collaboration. And if you look at that with RAM you'll see that RAM provides community and asset level collaboration.

These things work well together in the sense that when you do collaborate on an asset or upon a community of assets that collaboration in RAM stays close to the asset and becomes searchable, discoverable material that you can end up searching on all the content and discussions that take place.

So, typically we'll see organizations that will use those collaboration tools like Connections or SharePoint that are

dealing with projects, that are dealing with organization levels that are targeting a certain kind of user. So, you might have managers or executives or perhaps even business users that may not find themselves going in to looking at an asset in RAM, if you will, but they would see them working in the portals of those collaboration tools for their particular project and teams.

So, what we've done with certain customers is provide the ability for bidirectional linking. And the answer to this depends on the background of the company involved. So, in some cases we've seen organizations where they've been using collaboration tools with little governance.

And what's that done is it's created an environment where there's some content that's very valuable to the organization and those collaboration tools. But it's also intermixed with a lot of content that may not be as valuable.

So, companies will build links from the assets in RAM to those pages and those collaboration tools that are highly valuable. And that ends up saving the people time so that they go from the asset over to that specific content in something like Lotus Connections or SharePoint.

Going the other way, we've also worked with organizations

that also support the following scenario where they'll have the business user or the manager come in through Lotus Connections, perform the collaborations there.

But as they're in the pages then it executes searches on RAM. And what comes into their page then in Connections or in SharePoint is the list of assets that have been governed and signed off and approved and are relevant to the content of the page that they're currently sitting in in Connections or SharePoint.

But both those scenarios we've seen take place where they've enabled integration between the collaboration tool and RAM.

And then the other sort of key point I started out with was that RAM also, of course, provides its collaboration at the asset level.

So, in short, RAM's got the collaboration on assets and communities for those assets which often exists beyond the scope of a project or an organization which users are built in. That's probably the final point here, is that you'll see that RAM, in those assets, that they'll often survive the scope and the timing of any given organizational team or any particular project which is often the boundaries that those collaboration tools are built on.

MATHENY: So, Grant, how are organizations measuring and

improving reuse?

LARSON: So, we've seen organizations measure at different levels and at different times in their software delivery activity, but here's a couple examples of what we've been seeing companies do.

Some measure the effect of using assets on their project. So, a moment ago we were talking about how some of the collaboration tools get focused on a project perspective which is a very necessary thing to do.

But as we talked about, the lifetime of the assets might actually go well beyond -- and oftentimes you want it to so you get value out of it -- might live beyond multiple projects. So, in this case we've seen organizations, and I'll refer to one, this is actually a community within IBM that did this in 2009.

What they've done is when every project comes along they look at and examine those assets that have been used on that project and they determine the value of that to the project.

And the way they do is they found out that in the first six months of using RAM in 2009 that they have saved \$5.7 million in development cost avoidance...

...which of course we think is significant because it goes

right to the bottom line and it's on their organization to focus on competitive and strategic activity instead of just re-doing some of the same things again and again across their project.

The way they did this, though, is in RAM there is a cost and usage metric that you can capture on each asset. And so, when they submit their assets, part of their governance activities are that that information gets filled in. So, the people that develop the assets declare things like, what was the level of effort that they put into that version of the asset as well as what's the anticipated usage effort.

And they use those numbers in each project and they then extract what the cost and value was that was realized for that project. That's how they've arrived at this number that I mentioned a moment ago.

So that one is very much of a per asset basis, what's the effort that went into it, what's the effort that comes out of it? And the project manager then stops and examines, you know, what was used during that project and comes up with that calculation.

Another example that we've seen that I'll mention, this is another customer who is working with legacy services in the financial industry. They had the problem of many cryptic,

hard to understand and really nearly impossible to discover services.

And they had created in RAM a set of service assets by providing a set of searchable, understandable descriptions and categorizations that still provided the mapping to the cryptic names that the developer is ultimately going to use.

And what happened then is they exposed to the development community this set of service descriptions, and then that's what they did is searching off of when they were in Eclipse, or in RAD, or RID, doing their development activity, they're hitting RAM and doing the search and discovery on human consumable material that's helping them understand and find it, and then with a mapping to the actual names and locations where they're finding their existing mainframe or other services that were created.

One of the things that they discovered after doing this was that they reduced their service reuse cost by tenfold, and that meant that whenever any new services were being developed that they put in governance then that that new service also then, you know, needed and required RAM service asset that described it.

And they found that the cost of reuse went down 10 times, that people found their stuff and got applications developed

10 times more quickly by having those service descriptions and that governance lifecycle in RAM. And one of the things, of course, one of the factors for them was that they had teams spread across the planet, so, you know, it was hard for them without having an enterprise solution like RAM to help solve this problem so they need a way to easily discover.

Another sort of key characteristic that came from this was, once they had done this they found that their enterprise architect could then more easily examine redundancy of coverage in their existing legacy services. Not that they couldn't do before, but it sure got a lot simpler. And that's, every time you do that of course, simple means taking cost out of the business so that was a real benefit to them.

And then the third thing I'll mention here are that the experience was, because they did this, because they lowered their service reuse cost by getting those service assets in RAM, it actually allowed their management team to bring on new people and also to outsource more of the application development.

So, and the reason why was because the learning curve for understanding and reusing services dropped so dramatically they actually were able to significantly impact their

business model and reduce costs significantly. So, those are a couple of examples that we've seen organizations use for measuring their reuse and for reducing cost in their business.

MATHENY: Grant, what's the role of the Definitive Software Library in managing asset reuse?

LARSON: So, the Definitive Software Library -- or, sometimes we refer to it as the DSL -- it serves as the place for the golden bits to be managed. And that term golden bits might be a little confusing. What we mean by that, and oftentimes we might think that golden bits means just the software, you know, just the running executable. And it certainly does mean that, but it means more, too.

We find that for any given release, whether the release is an internal product that's being delivered, or internal tool, or something external that's being used in the commercial marketplace or what have you, for every delivery there are golden bits that are required, they get baked into that release.

And so, a release and a DSL means that there are many different assets that represent the version golden bits for that release. So, for example, you might have the golden bits of a business case for that particular release, and

that might be in the form of a PowerPoint presentation. Or you might have the golden bits of an enterprise architecture that's in the form of one or more System Architect models.

And it has relationships to that business case aspect that I mentioned just a minute ago. So, then that says, for this particular release, here's the business case and here's the enterprise architecture strategy that it's compliant with. And moving further on, you might then have the golden bits of your software design, which is an RSA model. And it would have relationships then to the enterprise architecture asset as well as to the business case.

And then finally, the things that of course we often think about is we would also then have the golden bits of the software components and the build that were comprised of that. And again, they would have all the appropriate asset relationships.

And by having a DSL with these kinds of relationships of assets across this release, across the entire software delivery lifecycle, it starts to get the business the ability to make decisions about impact. So, if there is going to be a change to one of those assets in the next release, what are the impacted or associated assets?

Who are the owners of those, and who uses those? So you now

start understanding both what gets impacted and who gets impacted, and again, that's about improving the governance and management and reducing some of the pain and delivery aspects around software.

MATHENY: Grant, you've mentioned governance several times already. Talk about the critical role of governance in asset management.

LARSON: So, certainly governance is one critical aspect of asset management with the DSL. The reason why that of course is the enterprise needs to determine what the level of governance it's going to enforce and it needs flexibility in doing that across communities. And so, even within any given enterprise, there are many communities, and across those communities you'll have the spectrum of no governance all the way up to highly formal governance.

And you need to allow the enterprise that flexibility to have those pockets of skills or techniques within the business to do that. So, for example, you might have a community where it's going to follow an open source style where there might be little in terms of the way a formal governance in terms of automation where you're relying on rating systems and the views of the people and what the crowds think. And that's an extremely important style of governance, if you will, for a certain class of software

[INAUDIBLE].

All the way up to communities where the governance is highly formal, where as material moves through it, as assets move through the governance lifecycles in there, certain teams always need to be notified. The legal team needs to cast its vote, and the QA team needs to perform certain things, et cetera. And there needs to be audit trails and traceability. Certain industries require even more of that, where even the slightest change to documentation needs to be traceable, and we need to know who touched it.

So, having the ability to define governance in the DSL along with that spectrum becomes important. Just take a minute here on what we mean by governance. Our experience with it tells us that there are several key elements to governance.

One of those is, which I talked about a moment ago, around the notion of community and defining that community.

And so first you say, well, here I'm going to have a community of people and the assets that are relevant to them. They have sufficient affinity in terms of the problem set that they're focused on, the types of assets that they're going to be using.

The second sort of key part around government is determining the shape of the assets and their categorizations in that

community. So there the idea is, for our community, here is what we mean by a component. Or, here's what we mean by a service interface, or by a business case, and so forth -- meaning it's got a certain set of attributes and has a certain set of content and requires certain relationships to exist and must pass certain policies and so forth.

Which leads into the third of the six things I'll mention here. The third is around the notion of the asset lifecycle definition. This often is the crux of the governance aspect, meaning, it's where you're going to dictate, what are the steps that the assets will go through in this community? What states will they be in, what policies will be fired at certain states? Who gets notified, and who signs off on it?

And you always proceed with caution at this point. It's sometimes easy to maybe over engineer some aspects of this where you might choke some of the life out of a given community if it's overly governed, or you might end out spending more time doing reviews and sign offs and getting other things done. So, you do need to proceed carefully and find the right balance.

The next key around governance we've seen is around to finding access rights and role definitions. So that's just things like, well, in this community users that are of the

role [of] developer have full access to assets of type service, but they have read only access to assets of type business case perhaps. And that's where you make those kinds of decisions.

And then collaboration and notification, keeping people aware of what's next on their task list for them to do without requiring them to deviate far from the normal day-to-day activities is a key part of a DSL, is a key part of making sure people are aware of what is next on their plate of things to do. And they know who they are working with while they're doing that, and so governance has collaboration as a key part there.

And then the final piece I'll mention by what we mean with governance is all about some of the feedback mechanisms, which is all the usage metrics and the measurements, you know, who's touching it, who's using it, who searched on it, who actually retrieve it, what are the comments that have been made on it and so forth.

All of that becomes critical because when you go to create Version 2 of the asset, those metrics and all the activity logs and everything captured represents your target consumer, your community that you're going to reach out to.

And you need to understand the impact to them certainly as you move forward. So yes, those six things tend to be

included in what we at least mean by governance.

MATHENY: There's a new release, RAM 7.2. What's new?

LARSON: So, there were quite a few things that we put into this release of RAM. One of course that was mentioned, we talked about here, was putting RAM more and more on to the Jazz platform. And I've mentioned what some of those were, such as RAM using the Jazz foundation server, and the RAM collaboration now has begun on Jazz.net.

So, on there you will see there is an asset based collaborative application lifecycle management scenario. And there's also this work on the OSLC Rest APIs which has started, and you can begin to see all that happening on Jazz.net.

In terms of the Jazz capabilities as I've mentioned, there's the ability to federate across the Jazz repositories and link to resources using this OSLC interface, so that's certainly new in RAM 7.2.

As well as the deeper Build Forge integration, so there is the ability to trace the RAM to Build Forge and vice versa, and deeper API support for Build Forge [INAUDIBLE].

There's also the ability that I've mentioned with the Jazz

platform to customize these lifecycles and to pull those in.

And you end up configuring those per community and so, again, that gives you the flexibility to enforce or to loosen up some of the governance activities we've talked about.

A couple other things here, is we actually provided some of the lifecycles out of the box. So, it's not like you're starting with a blank sheet, but seven or eight of these that you can start with and begin to configure, customize for your particular RAM deployment.

A couple other things here is that there is some usability improvement. RAM has always had some ease of use capabilities, and this just continues to improve here. There is the ability on an asset now to declare thumbnails, and so there you can highlight certain content in the asset.

So, if it's a PowerPoint file, or if it's a PDF document from a redbook, or whatever it might be you can highlight these so that when someone comes to the asset summaries and overviews, thumbnails of those with summaries appear and it aids some of the browsing and selection activity.

There's also improved asset publishing. This is one of the key feedbacks that we got from customers. In the past, you would walk through sort of a wizard style of submitting

assets, and now it's a sort of a simple one-step publishing.

You don't ever leave the page, but you're collecting all the material that you need.

And while you're in the course of doing that, you might decide to use some existing asset as a template. And so, there's a list of assets, template suggestions sitting off to the right that if you click one of those RAM will create a brand-new asset for you, fill in everything, and then you're just doing modification and cleanup for the current asset that you're working on. So, asset publishing got a lot easier.

Just a few other things here, one is around the RAM WSRR integration. There's been a lot of improvements on that synchronization, where now there's the ability to find the mappings of service assets in RAM to the service documents in WSRR. And to keep each side up to date, so that if there were a change to a service document in one of the WSRR registries, that would then get synchronized with the service assets of RAM and vice versa. So, bidirectional synchronization between the two has been another key improvement.

We also added quite a bit in the asset reporting area. I had mentioned briefly, I referred to it briefly, which is around asset reporting, the ability to understand trending,

to understand who are all of your asset users.

So, at the asset level I can see that a certain user has retrieved it, download it, and RAM gives the ability for sending e-mails to them now. And that of course helps with the asset versioning scenarios where you want to make users aware that new versions might be coming out.

And there's some graphics for this now where you can look at trendings or you can see that the asset is used in a particular build. You can see those that are downloading it.

Also new in this particular release was community reporting.

What was key here was we needed to add this ability to say, well, let me understand, you know, the summary of the asset usage and asset distribution for this community.

So, there's some graphics in there now that allow you to see the assets by type and by state and those that have the most usage in that community and what are the top viewed assets and so forth. So, now community level reporting is in RAM as well as the ability to integrate using the RAM report URLs. You can integrate real easily with Rational Insight.

And so, the way that works is you'll give it a RAM HTTP request as a URL and you feed that to Rational Insight and

then Insight works its magic creating great trending and great chartware and pulling in from multiple sources to give cross reporting.

I'll mention just a couple others. You can tell we're excited about this release, a lot was put into it. RAM is also participated in this release with the IBM Smart Business Developer crowd. So, here are RAM is being used as a place for managing the images that are used to then provision in the clouds. And so, we're excited about RAM's participate in the cloud work.

Also in this release the rich client -- meaning the Eclipse client -- had a lot of new capabilities added to it. One of the capabilities is if you are working on an asset and you're in an Eclipse environment, again, whether that's RSA or RAD or RID or what have you, WBM...

If you're in one of those environments and you're modifying and making some changes to an asset, and let's say someone else in another part of the planet also make a change to that asset and it gets submitted, now the rich clients pick that up.

So, in the background they're just sort of checking occasionally and saying, oh, there's been a change to this asset you're working on. What do you want to do? Do you

want to pull those changes in and do a compare and merge and so forth. We find some great new capabilities there, plus some new abilities for dragging and dropping artifacts and composing assets has certainly got a lot easier.

And then the final thing I'll mention is the licensing has changed. It's gone through a floating license model, and as you'd expect there's some new license management tools in RAM to support that along with some additional platform support. So, quite a long list as you can tell.

MATHENY: Yes, there's a lot to be excited about, though, Grant, thanks. Let's end with this. With a look to the future and looking ahead, what are the RAM development priorities?

LARSON: Well, there's several here. One is we're going to continue with the Jazzification of RAM. And I don't know if that's a term, but I'm going to use it. The Jazzification of RAM will continue, and part of that means that RAMs going to become a project on Jazz.net, so we're hoping customers and partners will participate and get involved with that.

RAM will be implementing the OSLC Version 1 spec that I had mentioned earlier, so that will be coming out. We're also going to continue improving the ease of configuration and

expansion of RAM. RAM typically is configured for a given domain, a given environment and continuing to simplify that and make that easier to do is a key priority for us.

Also, the ability for architecture asset management and governance to continue to grow will be a key point. And what I mean by that is we need to help our customers realize the benefit of their investment in architecture and management tools.

So, RSA and System Architect and Rhapsody and so forth, deeper integrations with those where models and portions of models are managed as assets is going to continue to see work.

As well as in domain specific asset libraries, and that helps bring up some of the configuration and support some of the ease of configuration pieces I mentioned earlier. So, like [TOGAS] and IIF and other sorts of libraries. So, that's a brief look of some of the things we're going to do as well.

MATHENY: Well, even if Jazzification isn't a word and we're going to make it one.

[LAUGHTER]

Grant, thanks so much for sharing your time today to discuss

Reduce, Reuse, Recycle, Jazz-based Asset Management Made Easy. We really appreciate it.

LARSON: Thank you.

MATHENY: That was Rational's Grant Larsen, Chief Architect for Asset Management. If you're interested in more podcasts like this one, check out the Rational Talks To You Podcast Page at www.ibm.com/rational/podcasts.

We'll also include a link to the recently published white paper, Using a Definitive Software Library as Part of Your Asset Management Strategy. This is been an IBM podcast. I'm Angelique Matheny. Thanks for listening. Keep tuning in as Rational Talks To You.

IBM Podcast

[END OF SEGMENT]