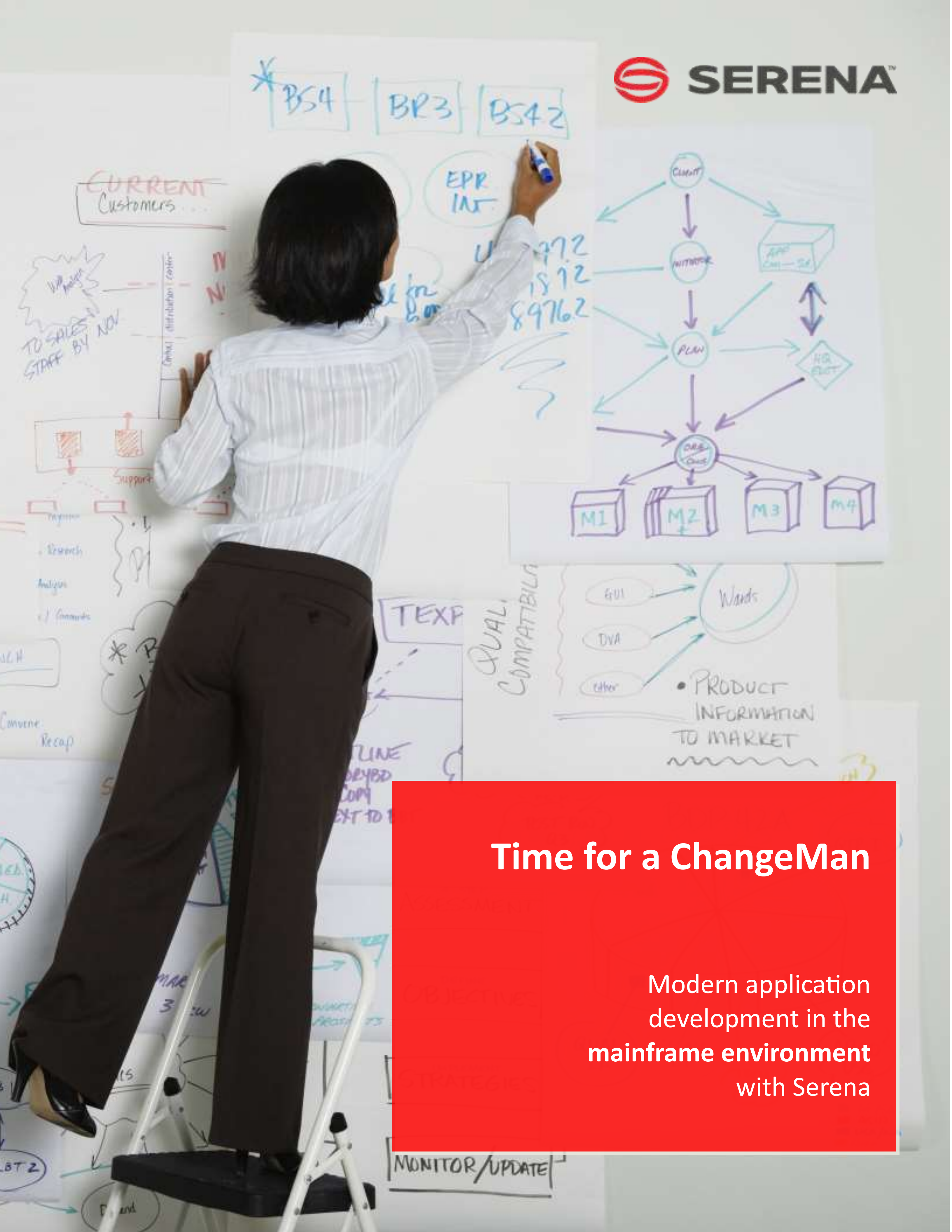


SERENA

Time for a ChangeMan

Modern application development in the **mainframe environment** with Serena

“We had a homegrown system that would not scale. With Serena ChangeMan ZMF, we could move a lot more changes without adding any staff even though we grew by 30%. Most importantly, Serena ChangeMan ZMF allowed us to handle very large clients who, in turn, helped fuel our company’s growth”

Manager, Service Provider

Welcome,

Thank you for taking the time to read the latest in our Technology Blueprint Series. This special edition focuses on the mainframe.

The mainframe has long been at the heart of enterprise computing. This is equally true for Serena. Our first mainframe product was created in 1980 and continues to be a key technology in hundreds of installations every day. From our first product, Comparex®, to the very latest, ChangeMan® ZMF Client Pack, we lead the world in innovation in mainframe solutions.

You drive our success

We achieve this industry leadership entirely through our close relationship with you. It has always been critical to us that we deliver technology that meets your current and your future needs. To make sure we keep track, Serena staff are active meeting with customers, industry analysts, press, partners and other mainframe vendors on a daily basis.

One of the key ingredients to our success is the Customer Advisory Board (CAB) with ten of the top mainframe users in the world guiding and validating our product direction. And there are Virtual User Groups (VUG) and Special Interest Groups (SIG) that meet regularly to share ideas and experiences. Once a year we get together in person at the Serena customer conference, called xChange, and we take a deep dive into the current and future technology. At xChange customers rub shoulders with Serena's mainframe developers, support engineers, professional services and pre-sales specialists in order to "exchange" insights and know how.

Today our mainframe products are installed at more that 2,500 customers worldwide and the list of customers just keeps growing. Where many vendors have moved away from the mainframe, or are treating it as a legacy revenue source, Serena continues to invest and innovate. With the pace of change in the mainframe increasing, vendors have a responsibility to keep up to date.

We have a bright future together

Barely a month goes by, it seems, without there being yet-another-trend: a new methodology, a new technology, a new paradigm, a new standard, a new regulation, a new best practice, a new job function.

Just like you, Serena is determined to exploit the new industry thinking. That's why we are the clear choice for customers who need to address their release management challenges. No other vendor has the same depth of knowledge, understanding and practical experience as we have. And no one else can offer the range of proven, comprehensive solutions from request-to-release across all of your enterprise platforms.

As the development world re-invents itself once again, Serena is here with developer-centric solutions that never compromise the integrity and governance needs of the organization.

Whatever your development paradigm, now or in the future, Serena will be there to ensure you maximize the productivity of the team.

On behalf of the whole mainframe team, I want to say thank you for being a customer .

Sincerely,

Kevin Parker

Kevin Parker, Chief Product Officer, Mainframe Products and Solutions
San Mateo CA, March 2014

“Based on downtime savings alone, some Serena Change-Man ZMF clients experienced payback in as little as two to three months.”

Industry analyst

Contents

Subject **Page**

V2014/03/04

Serena Software Inc.

1850 Gateway Drive, 4th Floor
San Mateo CA, 94404
United States of America

www.serena.com

Serena was founded in 1980 by Doug Troxel here in Silicon Valley. Doug is still an active part of the mainframe development team, working on special technology projects, and serves on our board of directors.

Beginnings

From the very beginning, the solutions we developed had a common theme, “change”. We detect change, replicate it, prevent it, log it, control it, merge it, approve it, version it, package it, we synchronize it, we track it, we audit it and report on it, set alerts for it, make dashboards of it, we deploy it and, if needed, back it out.

Our first product, Comparex, was designed to compare two data sources and highlight the differences. This is critical in all mainframe production environments for checking that changes have been applied correctly. From flat files to VSAM, DB2 and IMS to now, z/FS and HFS and even XML data-streams and AES encrypted files, Comparex does it all. It is a clear demonstration of Serena’s commitment to continuously improving and expanding our products to meet the latest technology challenges.

Our second product is now the company’s flagship solution, ChangeMan ZMF. Recognized by industry analysts at Gartner, Forrester, Ovum and others, as the leading solution for Software Change and Configuration Management (SCCM) on the mainframe. ZMF (as we usually call it) is the heart of the development infrastructure for more than 90 of the Fortune 100. Without ZMF many household name businesses would not be able to meet their compliance and governance audits.

Today

Serena is still headquartered in Silicon Valley in a very high-tech office complex in San Mateo CA. Here, and at facilities all over the world, our development and support teams create advanced technology solutions to meet the vast array of development paradigms in use in enterprises today. Our award winning support team based centers in the US, UK and Australia are always on call to answer your questions. Typically our mainframe developers and support engineers have an average of ten years of experience working on and working with the Serena solutions they represent.

In our more than three decades as the leader in managing change our portfolio has grown into the most comprehensive solution set for supporting the Software Development Lifecycle (SDLC) from Request-to-Release.

Our solutions are all grown on decades of experience and always in close cooperation with you, our customers.

As enterprise customers have expanded to open systems, personal computers, web, cloud, mobile, embedded we were with you at each step of the way.

And as the paradigms shift in the very way we develop software, from Waterfall to Agile, our process-centric solutions, will continue to future-proof you so that you can meet the needs of your organization.

The industry-wide focus on the importance of quality means that our solutions are designed to support CMMI, COBIT, Lean, Six Sigma and ISO, on your behalf so that you can reap those benefits. The importance of DevOps, ITSM and ITIL initiatives have brought critical attention to the role of Ops and, once again, Serena has ensured our solutions enable you to exploit these new sets of best-practices and standards.

Future

We have worked hard to get you the solutions you need today. We invest continuously in our products to make them even better and to maintain leadership in the industry. We are frequently the first vendor to offer support for the latest thinking in IT. In our development labs we are already at work on the new technologies we know you will need a year, or two, from now.

As a proud part of the Silver Lake portfolio of companies we have unmatched access to expertise and resources for whatever the next big-thing is in our industry.

For more information about Serena please visit us at:

www.serena.com

And if you find yourself in Silicon Valley please stop in and say hello, or at any one of our offices world wide. We’d love to hear about how your business handles “change”.

Modern mainframe application development

When you develop applications for your organization you strive to create the richest user interface experience you can. As a result applications are now multi-platform hybrids with massive high-performance data marts on the mainframe with chic interfaces on our smart phones and tablets all with a dizzying array of technologies between the z/OS data and the iOS (or Android) glass. But every technology that comes along brings with it a new set of challenges, none more so than on the mainframe. The pressure from the business units to deliver more, faster and better is accelerating that rate of change.

Changing face of development

And those are not the only pressures on how we develop applications. Along with the complexity of the platform, there is the complexity of the application, the massive scale of cross-system interdependence and integration, more governance and compliance mandates, time-to-market, technology obsolescence, technology innovation, heightened security, out-sourcing, open source, and a myriad of others that emerge on a seemingly daily basis all within the context of flat to declining investment.

These pressures have resulted in two main trends in the mainframe world:

- Moving development from the mainframe
- Moving execution from the mainframe

These two are often done together but it is now much more common to see development moving off the mainframe.

There is a trend in the other direction too, which is to move development and execution payloads from open systems (UNIX/ Linux) hardware on to z/Linux instances running on the mainframe.

Development and execution platforms

We are going to look at the modern mainframe development paradigms in four main development categories:

- Development in TSO/ISPF for execution on z/OS
- Development in a proprietary IDE on the mainframe for execution on z/OS
- Development in an Eclipse-based IDE
- Development in an non-Eclipsed-based IDE

In addition we are going to look at the options from the perspective of the execution environments:

- Execution on z/OS
- Execution on Websphere on z/OS
- Execution on z/Linux
- Execution on USS

Integrated development environment

One might claim that TSO/ISPF was the first integrated development environment. The Integrated System Productivity Facility (ISPF) provides all the capabilities needed for editing, compiling and testing code. Of course it doesn't provide version control, impact analysis, change and release management.

One great strength of ChangeMan ZMF is that it is seamlessly integrated into ISPF and follows faithfully the long established behaviors of panel navigation and PF-key usage that have become second nature to us all.

Client Pack

In the new world of development more modern IDE's exist and they are often now used for mainframe development. Amongst these IDE's are a class of tools based on the exceptional, and now market leading IDE, Eclipse. In Eclipse there are "perspectives" which are personalities for the IDE that configure it for optimal use for different kinds of programmers. There is, for example, a COBOL perspective. Serena has created a ChangeMan ZMF perspective that gives full access to ZMF controlled datasets on the mainframe from Eclipse on another platform.

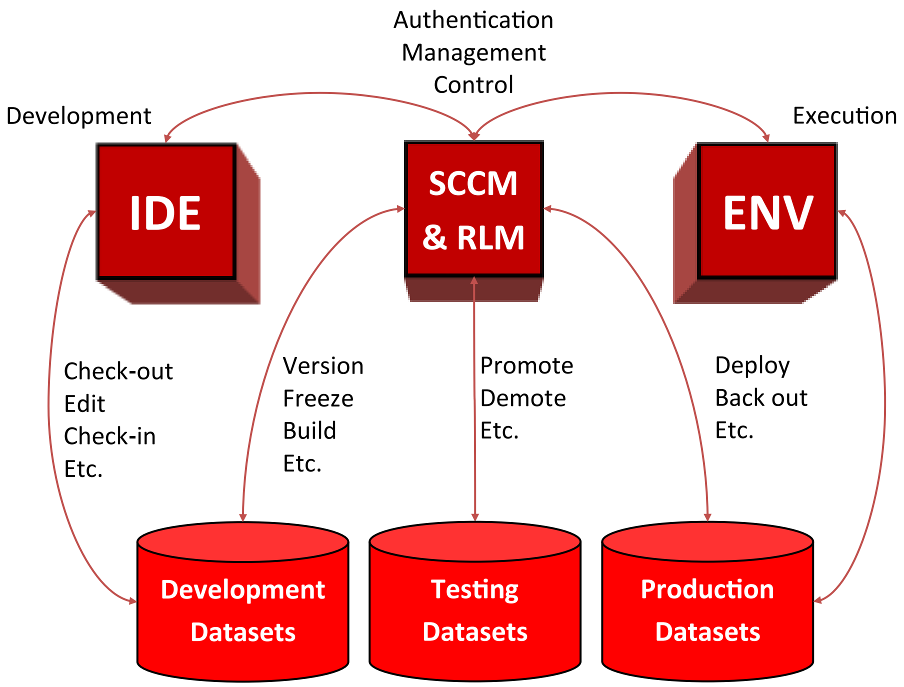
There is an IDE called Rational Developer for z/Series, more commonly known as RDz, which reuses the Eclipse code base and Serena has created a plug-in for this IDE too.

For other IDE vendors, such as Micro Focus, Serena provides a very powerful cross-platform connectivity solution that extends the mainframe to the desktop. All datasets, job queues and particularly the ChangeMan ZMF repository, become drives and directories. This means that, without a single integration necessary, any Windows application can now safely manage mainframe assets without the need to replicate them in another environment.

All of these features come as a standard part of the ChangeMan ZMF Client Pack comprising of:

- ChangeMan ZDD cross-platform connector
- ChangeMan ZMF Eclipse plug-in perspective
- ChangeMan ZMF RDz plug-in perspective

Classic Development Model



Now and future infrastructure

Replacing your infrastructure every 18 to 24-months as new technologies emerge and new ways of developing become fashionable is disruptive, impractical and expensive.

Multiple infrastructures, one for each development paradigm, is impossible to manage and does not allow for cross-team coordination and collaboration.

You need a technology that is architected from the beginning to embrace the changes in both platform and methodology as they occur. A technology with decades of development and deployment experience built in that proves itself every day in some of the world's toughest IT environments.

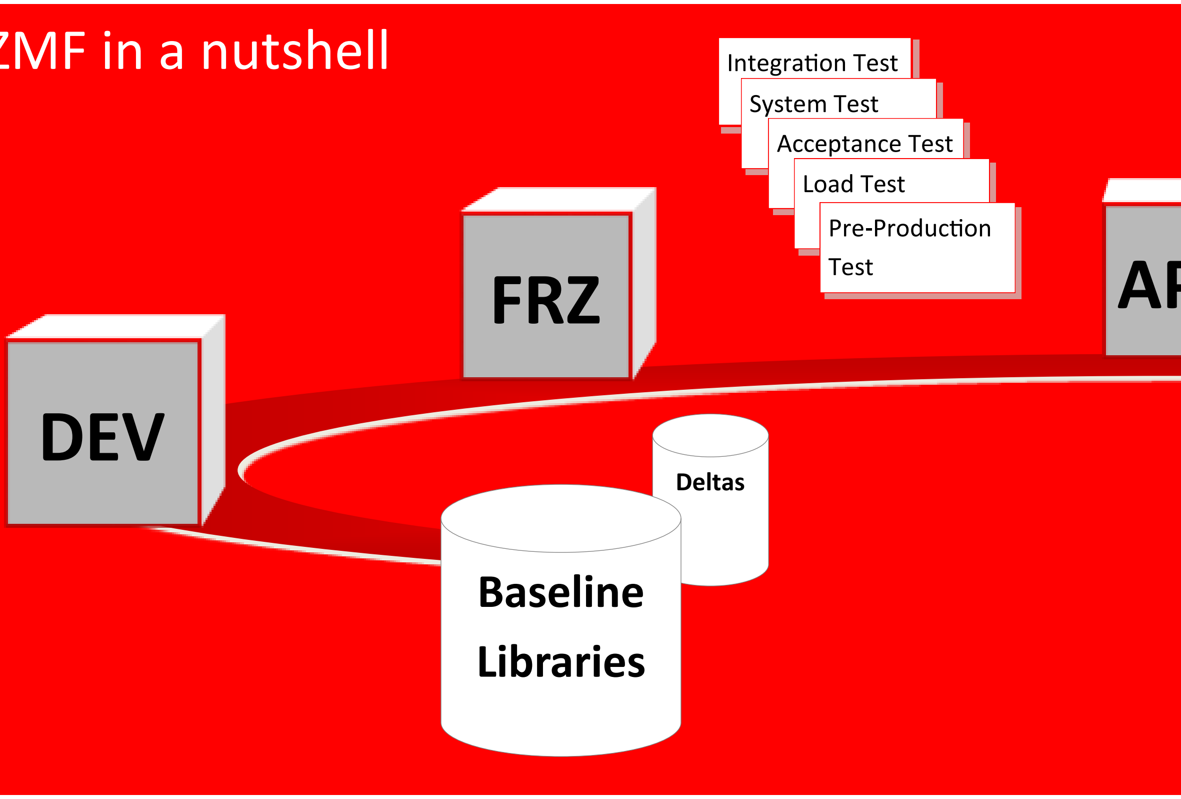
As you will see in the next few pages ChangeMan ZMF is uniquely able to do that.

Capability	ChangeMan ZMF	CA Endevor	IBM RTC	ISPW	IBM SCLM	CA Librarian CA Panvalet	Home grown
Versioning	Yes	Yes	Yes	Yes	Yes	Yes	Sometimes
On mainframe development	Yes	Yes	Limited	Yes	Yes	Yes	Usually
Off mainframe development	Yes	Yes	Yes	Yes			
Deployment to multiple environments	Yes	Limited	Yes	Yes			
Collaboration tools	Yes		Yes	Yes			
Unauthorized change detection	Yes						
Platform optimization	Yes						
Full parallel development support	Yes	Limited	Limited				
Mainframe release management	Yes	Limited					
Enterprise release management	Yes						
Mainframe subsystem automation	Yes						
Predictive audit	Yes						

Crowded landscape

In a crowded landscape choosing the right software change, configuration and release management solution is difficult. There are many competing choices from a variety of well known and well respected vendors. Price is always a consideration, of course, and so is capability. Domain expertise is critical in the selection of any tooling. You want the best technology to meet our current and future needs. With 30 years of experience specializing in solving these problems for the world's most advanced organizations there is only one choice … so choose Serena.

ChangeMan® ZMF in a nutshell



Introduction

ChangeMan ZMF is the easiest to use Software Change, Configuration and Release Management solution on the market.

Right from its inception it was architected to fit naturally into the lives of software developers giving them rich tools to make their work easier. It also provides capabilities that ensure programmers don't overlay each other's code by offering the world's best mechanism for managing the complex task of parallel and concurrent development.

It can even check-out code on behalf of the developer and recompile it automatically when it detects dependencies that are affected by a change.

Whether you are using Waterfall or Agile (or a hybrid we call "Wagile") ZMF meets your needs. If you code in Streams or Changes-Sets, ZMF supports what you need to do. Branching and Merging are a breeze for ZMF with a tool that actually suggests how best to merge code.

Whatever your development methodology, ChangeMan ZMF is already designed to meet your needs.

The Package principle

At the heart of ZMF is the concept of a "Package". This is the basic unit of work that moves through the lifecycle. A Package is the place where a developer checks-in code, edits, compiles and tests it before sending it along to the next step of the lifecycle.

ZMF does not define how a Package should be used. A Package might represent a single change tied to a single change request. It might contain the changes from be several change requests, even a whole new application or even a release. ZMF has no upper limit on the number of components that can be managed in a package.

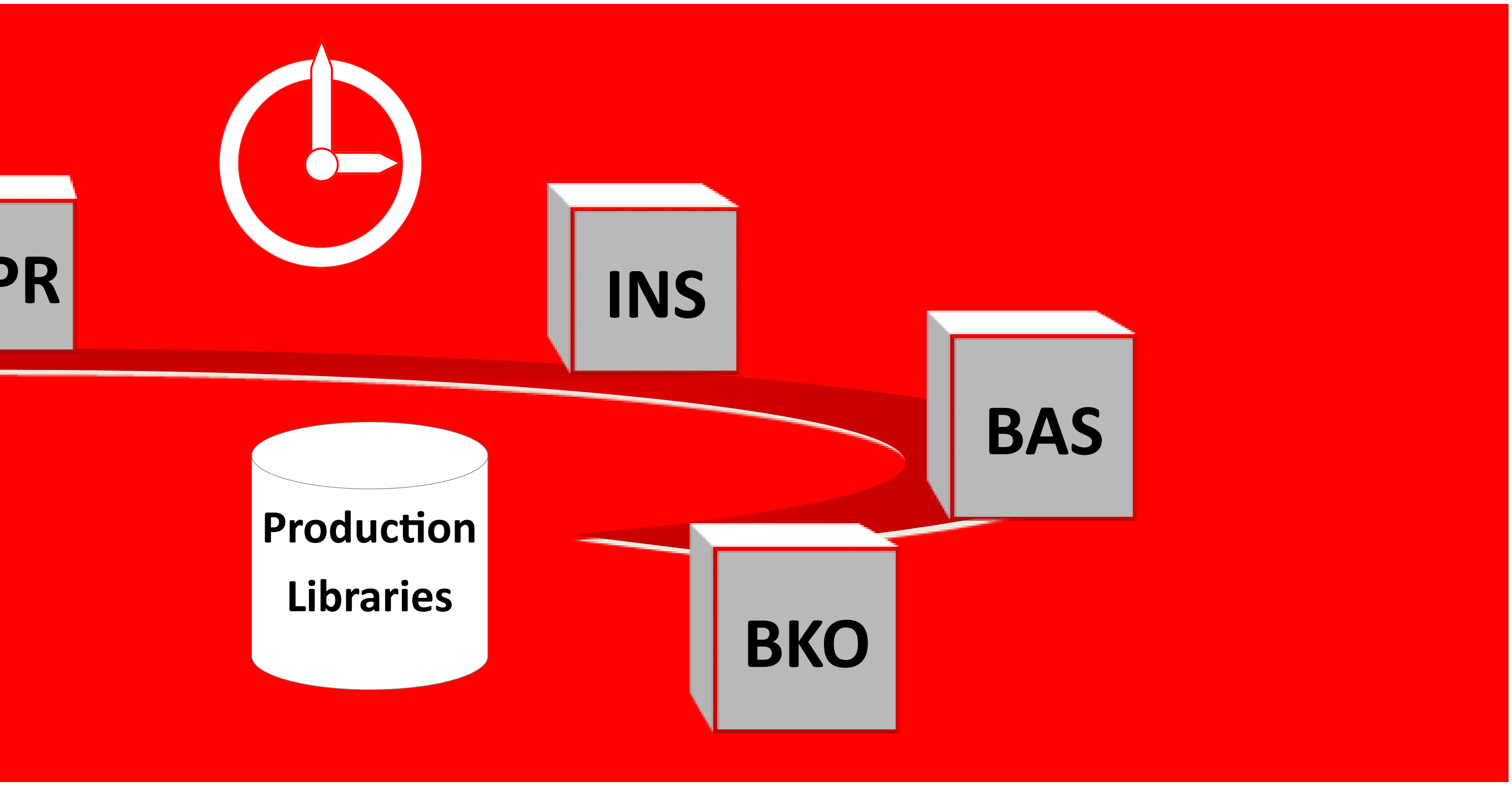
Nor a lower limit. So a package might just be the changes from part of a single change request. ZMF has maximum flexibility and is designed to support your way of working.

The Package lifecycle

When a developer wants to start a new development or new maintenance task they Create a Package. ZMF makes this easy to do and developers can clone, or model, their package on one they have used before. The Package contains information about the purpose of the Package including things like the task number requesting he changes and the date the Package is due to be installed to production. Some team leaders like to do this for their team members. Again, ZMF provides the flexibility and controls you need to make this match your current process.

DEV: When the package is created it is said to be in "development", or more simple DEV. Developers can check in code from the baseline, where the latest version of the code currently running is production is found, or they can check-in from another Package, creating a branch, or they can check-in from a dataset outside of ZMF's control.

The developer now begins the edit-compile-test iterations necessary to complete the task (or tasks that the package repre-



sents. Developers can check-in to the package at anytime the package is in DEV status.

FRZ: When all the development is completed the developer, or team leader, or QA engineer even, as your process dictates, will now lock the changes in the Package by issuing the Freeze Package command. This puts the Package into “frozen status”, or FRZ.

When a Package is frozen none of the components in the package can be changed. The Package can be unfrozen and returned to DEV status if needed and who can do this can also be controlled by ZMF.

PROMO: Once a Package is frozen is can be promoted to the testing areas. There can be any number of test areas, “promotion areas” in ZMF terms, or PROMO areas. Promotion of the code to a test area can be set up to require that the code be promoted in sequence through these areas or, as needed, to suit your process rules.

At each stage of the testing cycle approvers can be set up. These approvers can be named individuals or people in a specific role. You might require several approval levels and you could have, say, a rule that requires any two individuals from a list of 5 people. Once again all designed to meet your process control needs.

APR: When, according to your process rules, the final approval is received the Package automatically shifts to “approved status”, or APR. Once again, if needed, the package can be reverted to frozen (FRZ) or even to development (DEV) status if needed. Controls are in place to determine who can do this of course.

Scheduler: Built-in to ChangeMan ZMF is a deployment scheduler. ZMF also works with all the common scheduling tools there are from third parties. The scheduler looks for the installation date that was set on the Package. If the Package is in APR state and the installation date and time has arrived, ZMF will automatically move the executable parts of the Package into the production environment.

INS: At this point we have now installed the changes and new code that has been developed so we change the Package to “installed status”, or INS.

BAS: Once the installation is completed successfully, ZMF then updates the baseline with the new code that is now in the production environment and it versions the old code in the repository. We call this the “baseline ripple”. When this completes and all the components have their versions rippled one version back the new code is copied, from the package, into the baseline libraries. All the versions are kept in “stacked-reverse delta” format in a separate delta-version library. Finally we change the Package to “baselined status”, BAS.

BKO: If, subsequently, we need to back the code out if, say, a production error has been found, ChangeMan ZMF can do that very easily and automatically. The number of people who can do that is restricted and when it happens the package is marked as being in “backed out status”, or BKO. We also restore the baseline libraries so that they match the production code.

ChangeMan ZMF vocabulary

Here is a list of the common terms you will here used by the ChangeMan ZMF community. The usual terms for the process of Software Change, Configuration and Release Management are still used as well.

Application: The libraries and the components within them are organized into Applications. Each application represents one or more programs that are normally managed together. For example the payroll application. An application is also where you can set specific rules that apply to development in this application thus allowing different development processes for different application areas.

Approval: The act, by an authorized user, of agreeing that the criteria they have for the code to move to the next stage has been achieved. No action occurs until all approvals have been received.

APR: The "Approved" status of a package that has received its final approval from the list of approvers.

Audit: A ZMF feature that validates the integrity of a package (or packages) and reports on issues that could result in unexpected outages.

Autoresolve: A feature of Audit that resolves conflicts automatically by checking missing components into the package and recompiling modules as needed.

BAS: The "Baselined" status of a Package that has been successfully installed and the baseline ripple updating the baseline libraries with the latest code versions has completed successfully too.

Baseline: Also "Baseline Libraries": The libraries where the code that matches the current production code resides.

Baseline Ripple:

The process of updating the Baseline Libraries with the latest version of the code after a successful installation to production.

BKO: The "Backed Out" status of a package that has had the recently deployed components removed and the previous version of the system restored.

Check-In: Common name for moving components into a Package. Components will have changed. This does not trigger a compile. See Staging.

Check-Out: Move components into the package without changing them on the way.

Component: A member of a library that is managed by ZMF. Components are some form of source or executable code.

Delta: The previous version of components are stored in the delta-libraries. They are stored in a compact form in reverse-delta format. This is the fastest and most efficient format to enable a component to be restored to its previous version. See also Staging Versions.

DEV: The "Development" status of a package that is currently being developed. Components may be added to the package and changed.

Emergency Package:

See Unplanned Package.

Enterprise Release Option (ERO):

ERO is a feature of ChangeMan ZMF that is designed to support release management teams that operate in a highly dynamic environment. If release content and schedules are constantly changing ERO provides the needed automation to ensure that the new release configurations maintain their integrity.

FRZ: The "Frozen" status of a Package that has completed development and is now ready for testing. Components may not be added to the package and they can no longer be changed.

INS: The "Installed" status of a Package that has been deployed successfully into the production environment.

Library: A dataset that has been designated to hold one type of code, for example source code, or copybooks, or executables.

Merge+Reconcile:

A built-in utility to reconcile parallel development versions of components.

Package: A collection of components that are being developed together as part of a change request or assigned development task.

Permanent Package:

Packages are normally designated Permanent in tat the changes are expected to stay implemented. See Temporary Package.

Planned Package:

This is a normal package as part of a regular project lifecycle. See Unplanned Package.

Project: Term used interchangeably to mean Application for more project centric organizations.

PROMO: The short name for the Promotion Libraries.

Promote: The process of moving frozen code to the Promotion Libraries.

Promotion Libraries:

Test areas that contain the components from one or more Packages that are to be tested. Often just called PROMO.

REVERT: Move a package to its previous status. For example if errors are found during the testing of a frozen (FRZ) package it can be reverted to development status (DEV).

Scheduler: The built-in (or third party) clock that tracks when code needs to be deployed.

Stage: The process of checking a component into the package. This can be done from Baseline, from another Package or from a regular dataset. In some cases the Stage include then compiling the component. See Check-In.

Staging: Also Staging Libraries. The collection of datasets that make up the Libraries containing the components that are in the packages.

Staging versions:

In the package ZMF is able to maintain versions of the code in development so that developers can experiment with different approaches and revert back to earlier versions of the code they had developed. This is different to the Delta versions which record different versions that went into production.

Temporary Package:

A special kind of Package that is designed to back itself out after some defined period of time. These are used to allow temporary changes that will be replaced later by planned code. See Permanent Package.

Unplanned Package:

A package with special privileges that allow it to fast-track through the process and have fewer than normal approvers. Used for emergency fixes. Also known as an Unplanned Change.

Serena® ChangeMan® ZMF

Unmatched Mainframe Release Management Out of the Box

Release with confidence by including all the key stakeholders in the release process, from approval to receiving immediate notification of changes. Whether you are migrating one component as the result of an emergency fix or 50,000 components for a major development project, ChangeMan ZMF provides comprehensive release management capabilities.

You can use ChangeMan ZMF with Serena Release Control, our enterprise release management planning and control solution for both mainframe and distributed systems.

Maintain High Volumes of Changes

If you have complex and massive release management processes, the Enterprise Release Option (ERO) for ChangeMan ZMF simplifies the maintenance of multiple simultaneous release processes, including:

- Consolidating migration paths into an integrated environment
- Providing a release audit with automatic resolution of out-of-sync conditions

Automatically determining copybook and load module concatenations for build processes.

Work Faster with Change Packages

A change package is a secure development environment using rules stored in your security system. A change package consists of descriptive information, control parameters, component metadata, history information, and a set of libraries containing the software components being changed.

Bringing together all affected artifacts simplifies mainframe development. Change packages, invented by Serena, let you manage a project as a unit throughout your defined lifecycle while software changes are in motion. For example, if you need a special process for emergency fixes, ChangeMan ZMF supports that. Perhaps you need a different lifecycle for the online programs versus the batch programs. Or maybe the development task is vast and you want to organize the changes into groups based on geography, functionality, or technology. ChangeMan ZMF supports all these scenarios.

Built by Developers for Developers

Whether you are developing natively on z/OS in TSO/ISPF, or developing in Eclipse on Unix or Windows, ChangeMan ZMF has the solution for you. You can collaborate with your team irrespective of their development platform and irrespective of their execution target.

ChangeMan ZMF supports the development methodology of your choice too. If you are developing in a high speed agile environment ZMF is built for your way of working. With its flexible approach to defining work units you and your team can move code through the lifecycle at great speed all the while knowing that ZMF is ensuring the integrity of your work automatically.

Track Every Change

ChangeMan ZMF tracks every change you make. You can assess dependencies, perform impact analysis and, when necessary, completely back out a change on your mainframe.

Unlike other tools that only allow audits at the end of a lifecycle, ChangeMan ZMF lets you perform audits at any stage of the lifecycle. By auditing throughout the cycle, you can prevent problems much earlier, when they are easier and cheaper to remediate.

The audit function ensures the synchronization and integrity of all components that are being developed. If you need to back out, ChangeMan ZMF does that for you. The fully automated process backs out copybooks, source, DB2 components, and more, allowing you to get your production environment back to a stable state so you can begin correcting the problem immediately.

British Airways

COMPANY PROFILE
British Airways (BA) is a full service global airline, offering year-round low fares with an extensive global route network flying to and from centrally-located airports.

The company is part of International Airlines Group, and is responsible for more than 30 million passenger journeys per year. The airline has a fleet of more than 250 planes covering both short-haul and long-haul destinations.

THE CHALLENGE
As part of its support services, BA has an extensive system of applications for managing its flight operations. These are based on a variety of platforms, from IBM mainframes through UNIX and Linux to Windows. The IT team is responsible for the development and management of these applications across each platform.

As part of its technology strategy, the IT management team at BA decided to move away from an internally developed software change management system for its critical mainframe applications. This would allow the application development team to concentrate on support and development work that directly supported business operations, rather than on support tools Jim Dixon, Infrastructure Engineer at BA explained: “The decision was taken to migrate over to a commercial product for managing and tracking development. As part of this, the IT team at BA referenced IT analyst firm Gartner to identify appropriate solutions that would meet the firm’s needs around tracking activity, auditing changes and automating a repeatable process to install into Production.”

This project included managing development of applications that were critical to BA’s operations across the finance, engineering, flight operations and back-office functions in the most efficient and cost effective manner. These programs included the registered take-off and weight system, which checks the weight of all planes that are flying out. Information from this is then automatically fed into the applications that manage allocation of fuel to each plane, fuel costing and charging.

The software programs involved are therefore responsible for helping BA’s operations, finance and flight teams be as cost effective as possible.

THE SOLUTION
Based on the airline’s criteria, BA chose Serena ChangeMan ZMF as its platform for managing application development on its mainframe software. The original implementation project was completed and responsibility for tracking the key applications development was shifted over from the existing internal tool over to ChangeMan ZMF.

“ChangeMan ZMF acts as our repository for production applications, as well as maintaining a log of all events that take place around development. It provides us with the control required over who does what for each program,” commented Dixon. “As part of the whole process for updating some of our core software and applications, ChangeMan ZMF plays a central role.”

Since deployment, the ChangeMan ZMF system has taken over responsibility for managing application delivery across 38 applications. It provides a full process for all work carried out around development. This begins with product developments and code changes that are initiated in response to business requirements. Once the development work is completed, testing is carried out.

This is split into two sections. The first is testing to ensure that there are no problems or bugs introduced within the new software update. This is followed by User Acceptance Testing to check that the software continues to meet employee requirements around usability. Once these two stages are completed, the update is moved through to audit.

“This last stage before production lets us check the overall package that has been put together with no inconsistencies to other programs in Production and the team lead can ensure that everything has been carried out properly. If the team lead has been involved within prior coding and testing processes and their own work would require approval, then ChangeMan ZMF automatically flags this and a second approval contact must be used.” explained Dixon.

BA’s mainframe applications connect into a variety of other software, from other mainframe applications through to software that runs on UNIX, Linux and Windows. As the information output from the mainframe apps is related to business-critical processes within BA, this makes the software update and tracking procedure essential to keeping systems across the IT software infrastructure running smoothly.

“The applications that we run to measure, apportion and track fuel usage assists our business planning and performance data, so making sure these systems are fully tested ahead of new updates being rolled into production is essential. ChangeMan ZMF helps us to achieve this,” said Dixon

KEEPING PACE
ChangeMan ZMF has kept pace with BA’s own development of applications. BA has migrated to newer versions of ChangeMan as they have become available, as well as moving more of its software change management projects over to the solution.

“Our development teams covering the mainframe applications is about 80 people, though it can scale up to 120 when larger projects come through. As we need to add more resources into the development teams, ChangeMan ZMF ensures that we can continue to keep track of assets and changes made,” commented Dixon. “Over the past 15 years since we made the decision to go with Serena, we have continued to rely on ChangeMan ZMF for our software change management requirements as it meets the needs we have as a business and as an IT function.” he added.

Featured Customer

Impact analysis:

When you check code into a Package you have the option to run ChangeMan ZMF's Impact Analysis tool. During the compilation process ZMF is monitoring the code and makes a note of any dependencies there that refer to other modules and copybooks.

When you run Impact Analysis it determines which "upstream" and "downstream" dependencies the component has and gives the developer the option to check out those components directly into the development Package too.

Upstream dependencies are when the component is called by, or copied from, another component.

Downstream dependencies are when the component calls, or copies in another component.

Impact Analysis is a key component in improving developer productivity and in eliminating careless errors. Even the most experienced developer can miss recompiling a module with a shared copybook.

This is a built-in feature of ZMF and included with the product.

Audit and Autoresolve:

Every customer who has ever installed ChangeMan ZMF always says the same thing: "we could not live without Audit" and "we don't know how managed before without it."

Audit is one of the most sophisticated and advanced technologies in the world of software development. It's purpose is to verify the integrity of the components stored in the repository.

Audit monitors the changes being made to code, it looks for branching of code and the subsequent merging, it cares about shared components and copybooks and ensures the correct version is being compiled into the component in the development package. Audit looks for "rogue" changes to components that have occurred outside the control of ChangeMan and it tracks to make sure that the executable matches the source code to guarantee source-to-load integrity.

In short Audit applies a level of detailed study and analysis that would tax even very skilled and experienced developers and would take them many hours for just one project's worth of components. Audit does this in minutes for the entire inventory of components and it produces a report highlighting where all the errors and inconsistencies are. We call this the Predict-and-Prevent ^{SM} capability of ChangeMan ZMF and it is unique in the market.

Audit can be run at any time, it can be set up to run on a scheduled, regular basis. It can be limited in scope to just the current development Package (or Packages) or to a specific application.

Audit is frequently used at the beginning of the development lifecycle to make sure all the required components have been checked into the development Package. It has become mandatory to run Audit for most customers prior to freezing the Package so as to ensure the code is in a good state before it goes to QA. And most organizations run Audit prior to deployment as one final check to make sure everything is as it should be and nothing unexpected has been introduced.

But perhaps the most powerful and useful feature of Audit is Autoresolve. When Autoresolve is turned on and Audit finds a situation that it can rectify, it does so. It checks out the code necessary, recompiles the program and adds it to the package of deliverables. Not only eliminating costly errors but also saving valuable developer time.

These are built-in features of ZMF and included with the product.

Security and Access Control:

ChangeMan ZMF makes use of the built-in security services of the mainframe environment, namely SAF. It is compatible with RACF, ACF2 and Top-Secret.

All artifacts in a ZMF system, baseline libraries and staging (that is development) libraries are under ZMF's control exclusively. This means that changes to those components are only possible with the permission from ZMF and ZMF only gives that permission based on the user's rights assigned to them through their security system.

Special code is included in ZMF to detect unauthorized changes that may have occurred. Though this is a very unlikely scenario, ZMF is ever vigilant in its protection and control of your source code assets.

This level of security extends to developers who are accessing ZMF from off-host platforms. ZMF uses the same mechanisms to authenticate and grant access to Packages and components.

This is built-in features of ZMF and included with the product.

Merge+Reconcile:

One of the most difficult challenges in modern application development is parallel and concurrent development. It is not uncommon to have the same component being worked on by several developers simultaneously, one bug fixing an overnight issue, one adding new features for the big spring release, someone working on the release after that and, perhaps, some outsourcer adding code to support mobile phone integration.

Keeping track of those changes, making sure that code is not overlaid, or regressed, is more complicated now we are a global development team, in the office, off-shore and in our homes.

ChangeMan ZMF keeps all these versions straight and warns developers if they try to make a production change without merging the code and reconciling the potential conflicts that might result. Once again it is the audit feature that detects these situations.

But ZMF also makes it easy for developers to merge their changes. It does this using the Merge+Reconcile (we usually just call it "M+R") tool. This tool first analyses all the versions available, including the baseline version if required, and determines if the different sets of code changes are compatible, if they have a benign effect on one-another or if they are in conflict. Depending on the outcome it then grades the difficulty of each merge and produces a recommendation on the complexity of the task ahead.

If it is simple merge you can even have M+R to make the changes for you. If it is complex you might assign the job to your most experienced developer and if it is in between you get to make the call.

M+R can merge up to 8 versions, with associated versions of copybooks, simultaneously and it does this using the same ISPF edit commands that everyone already knows. This makes learning M+R an easy task for all developers.

This is built-in features of ZMF and included with the product.

“Cost of downtime due to bad code last year? $511,252!”

Any downtime is a huge issue for most organizations. All outages directly affect the business; unplanned outages can cripple a business. We do everything we can to plan for them and minimize them. However unplanned downtime is a serious drain on resources and is very costly, not just in terms of lost revenue, but, significantly in the damage that can be done to a corporate reputation. Disaster Recovery Journal estimates that 8% of unplanned downtime is due to software errors.

Estimates vary by industry but it has been calculated that unplanned outages average between $300,000 and $1,000,000 per hour.

One large retailing client who had been plagued a number of outages that occurred because of code regressions told us “One big reason we purchased Serena’s ChangeMan ZMF was to stop outages caused by changes. Those outages were lasting several hours apiece. Speedy back-outs of failed code and automated changes means no long outages anymore.”

Using the power of the ChangeMan ZMF Audit feature they are now able to Predict-and-Prevent ^{SM} code regressions. Instead of avoiding parallel development this retailer is now able to use our Merge and Reconcile tool to make this a regular part of the development process.

ZMF’s full back-out capabilities ensure integrity of the code as you return to the previous version while fix the new one.

Mainframe Release Management

"If anything is certain, it is that change is certain. The world we are planning for today will not exist in this form tomorrow."

Phil Crosby creator of the Zero Defects initiative at the Martin Company

Release management is the critical process in IT today. Never has so much emphasis been placed on a single IT discipline. Delivering releases safely to production requires that we balance the competing needs of:

- **Velocity**—time-to-market is the primary driver for organizations today. Delivering releases sooner means taking the coveted "first mover" advantage.
- **Compliance**—the level of scrutiny in business is at an all time high. With internal and external auditors, industry, federal and state regulators, methodology and process certifications, service levels and KPI's: there is an abundance of controls that we have never seen before.
- **Visibility**—from the "C-suite" to the PMO, from the dev-lead to the production-control manager, everyone is asking the same questions, what is in it, when is it coming and where in the lifecycle is it now? And everyone wants this information up to date and online.
- **Dependency**—modern applications stretch from the mainframe to mobile platforms. Systems are complex and highly interdependent. Minor changes can have profound impact. Sequencing of changes and deployments is sophisticated requires great skill.
- **Flexibility**—the business world is no longer predictable or stable. Changing needs, changing priorities, changing landscapes and technologies all require that we are able to adjust our release dependencies and delivery dates. We need to do this quickly, safely and with the minimal impact on resources.

Enterprise Release Option (ERO) solves all of these problems. Originally designed to meet the exacting needs of a large telecommunications company (see case study on opposite page), ERO solves the most difficult issues organizations face when trying to manage the complex world of enterprise release management.

ERO is built on top of ChangeMan ZMF and uses all the same base technologies. This gives organizations the ability to manage the release workloads that are light and simple using the tried and proven Package-based approach that is the hallmark of ChangeMan ZMF and an industry leading first.

There are three issues that are the most difficult to manage in any release management process:

- **Cross system dependency**—some systems don't have any external dependencies on the changes being made to other systems. The existing package-based approach works well for that. Where there are dependencies ERO will track those and ensure that any changes made to dependencies are notified to important stakeholders.
- **Release deployment sequencing**— most modern development organizations have many releases in motion for the same release. This means there are non-production code changes in motion that subsequent releases will depend upon. When the release sequence changes this throws out all the dependencies. ERO manages this by reporting on all of the components that are now out of synchronization as a result. It does this for both upstream and downstream issues.
- **Testing integrity**— when a release is being tested the concatenation of code is usually the code in test to the production code. But, with multiple releases ahead (closer to deployment date), testing teams want to test code against the actual version of the code that ***will*** be in production. Of course this code may not be reliable so ERO provides the ability to choose a concatenation path that meets the reliability expectations of the test team and test phase.

ERO is available for all customers of ChangeMan ZMF and is fully supported through the ChangeMan ZMF Client Pack from both Eclipse and Eclipse-based IDE's such as RDz.

On the following pages we will take a look at the capabilities of ERO and show how the architecture is designed to support you way of working.

For more information:

Please contact info@serena.com

Or contact your local Serena sales team

CASE STUDY

For one **large telecommunications company** release management is an extreme sport. At any time they have as many as 36 planned releases in motion simultaneously and any number of emergency changes to contend with too.

Today's telephony marketplace is dynamic and very competitive. Taking and holding on to a leading position requires great investment, innovation and the ability to get those ideas into the hands of consumers as quickly as possible.

Therefore Release Management has to be an organizational strength and not an impediment to delivery of new products and services. IT has to be able to switch priorities and juggle content delivery in an instant.

Release is our way of life

With an 18-month look-ahead window, "on-cycle releases" represent planned changes to their billing systems that will go live on the last business day of the month. These release can be as short as 4-weeks in the development and testing or up to 2 -years.

In addition there are up to 18 "off-cycle releases" that contain corrections, delayed content and other improvements to previously deployed releases and these too have a variable duration.

If there are outages or serious errors these are fixed in emergency changes.

This means a lot of code in flight, much parallel development and, therefore, a complex system of tracking and versions and ensuring branches are merged and dependencies are handled.

Each development team is responsible for ensuring the integrity of their own code.

Schedules change: priorities shift

If a release needs to be held back, or another release moved ahead in the schedule, the disruption to all of those dependencies, the need to "un-pick" code from one version and migrate it into a later one, the identification of new dependencies, is a mammoth task.

In a highly regulated industry like telecommunications, many changes have a legal requirement to be delivered on a particular date.

One of the most pressing challenges is dealing with more than 50 state and federal agencies who have similar, but often subtly different, requirements. When these schedule changes move around the exposure for non-compliance is a major concern.

Automation is the answer

As a long time customer of ChangeMan ZMF they turned to Serena for help. Working together in a close partnership we developed the Enterprise Release Option more than a decade ago.

Today releases continue to managed and tracked by ERO. Upstream and downstream dependencies are identified in the release definitions and when the sequencing of releases change ERO's Audit function reports on all of the impacts to each component in each affected release.

One critical feature is exclusively designed to support the quality assurance team. When testing a release the QA team can put the releases in sequence ahead in the dependency concatenation. This means they can simulate the test environment as it will be months ahead in production.

QA can even choose to pick up modules that are early in the test phase, or late in the test phase, on a release by release basis. This give great flexibility to the test team.

Release at the speed of business

The control that ERO provides ensures that the development and test teams can adjust to meet the ever changing needs of the business. The release team have the confidence to know that ERO is managing the impact of those changes.

ERO is the most advanced release management solution for mainframe developers available in the world today.

OVERVIEW

The increasing complexity of IT environments has placed new demands on Enterprise Software Change and Release Management systems, as software change must now be managed across geographies, multiple concurrent releases, technologies, methodologies, and platforms. Consistency yields efficiency, yet very few organizations have established consistent processes in each of their disparate computing environments, and fewer still have achieved an integrated enterprise infrastructure that gives developers, managers, quality assurance teams, and auditors the flexibility and control they need.

Serena gives organizations an advantage with the most thoroughly integrated multi-platform Software Change and Release Management solutions available. Furthermore, Serena has developed the Enterprise Release Option (ERO) for Serena ChangeMan ZMF, to extend its software change management solution to support organizations with multiple releases in motion simultaneously. For release-centric organizations, ERO represents a quantum leap in automation, productivity, auditability, and quality assurance.

RELEASE-CENTRIC AND PACKAGE-CENTRIC CHANGE

Solutions that focus on package-centric development approaches address the needs of many organizations. In a package-centric solution environment, all components of a change are grouped together at the beginning of the defined, repeatable, quality-driven development process. Developers and managers then act on the package as a whole, comparing (auditing) it to the production version to ensure quality and that the correct component versions are used when changes are made.

In a release-centric environment, developers and managers must maintain the relationships between the components and packages that make up any given application release. Relationships must be ensured between production versions and all release versions in motion. The comparison (audit) must be made between releases, not packages.

Quality assurance for enterprises with multiple releases in motion. When release schedules change, the Enterprise Release Option of Serena ChangeMan ZMF manages all component relationships, notifies release managers, and ensures that teams work with the right code, every time.

When release schedules shift and features are dropped from or added to a release, the release manager must make sure that all development teams are working with the right components. There is tremendous potential for rework and regression, and managing the process manually is a considerable challenge. The longer developers work with incorrect component versions, the greater the cost of changing release schedules. Furthermore, without an online audit function, inadvertent use of incorrect component versions poses a threat to production environments and regulatory requirements compliance.

THE CHANGEMAN ZMF ENTERPRISE RELEASE OPTION: A QUANTUM LEAP

The Enterprise Release Option for ChangeMan ZMF simplifies complexity in release-centric environments and delivers dramatic productivity improvements. It automates communication and equips release managers with tools for monitoring activity and component movement. For IT organizations driven by strategic objectives that result in several releases being active simultaneously, ERO aligns development practices with business objectives.

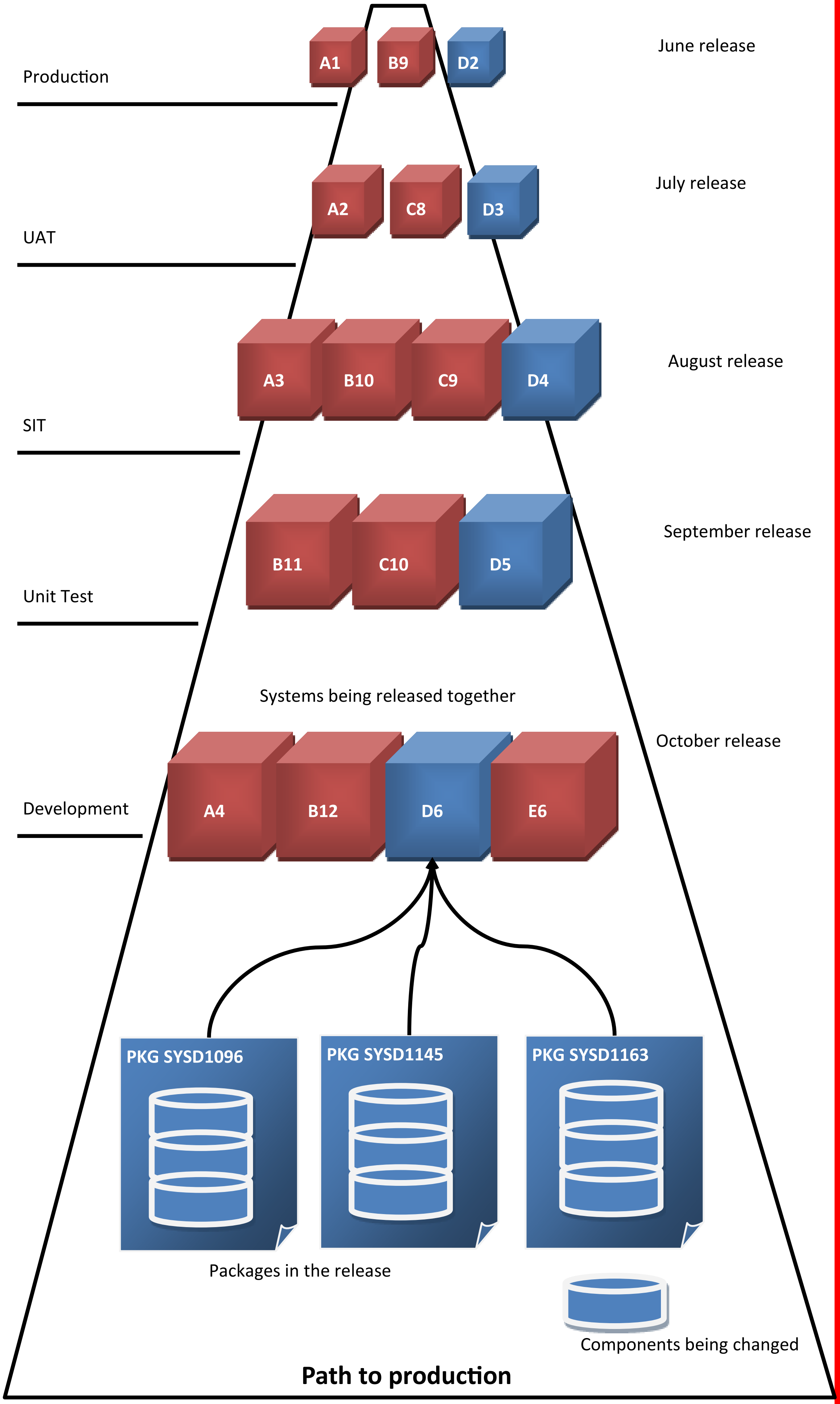
ERO is an elegant solution for release-centric development environments that simplifies the process of supporting multiple release versions. ERO is an optional module that extends Serena ChangeMan ZMF to streamline the release management process, and supports many packages from many applications in a series of dynamic and dependent releases, complete with audit capability. ERO allows all changes in multiple packages, and across multiple applications, to be managed at the enterprise level.

In complex environments, releases may comprise hundreds of change packages. The same programs may be changed in each release, and multiple releases might be simultaneous works in progress. A change anywhere in the release schedule can have an impact on a great number of releases or release management teams.

Automatically track dependencies. Having multiple releases in motion can create complex dependencies between components and releases, and this can lead to rework, regression, and lost productivity. The Enterprise Release Option automatically tracks dependencies and opens communication between release teams.

ERO makes it possible to establish and automate a consistent release management process. Organizations can manage releases as they move from development, testing, and finally into production. ERO provides a development path in each release that progressively consolidates application package components into areas until they reside in a single set of area libraries.

Enterprise Release Option Datasheet



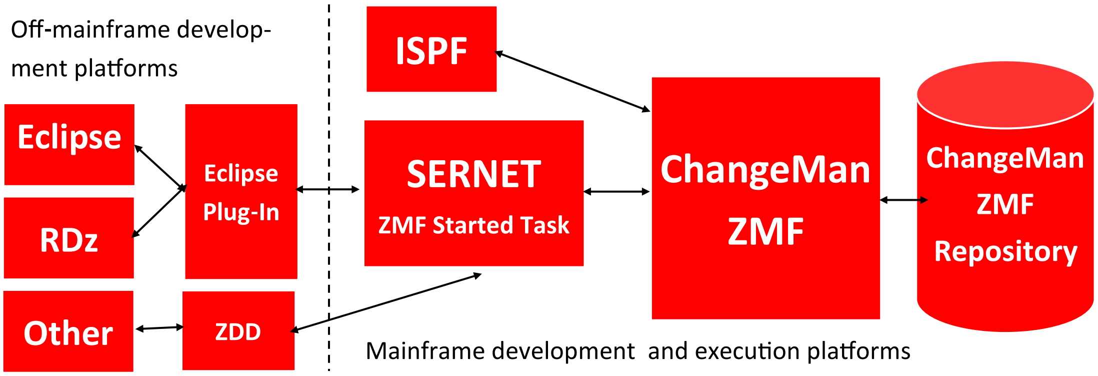
Enterprise Release Option Datasheet

info@serena.com

Serena® ChangeMan® ZMF Client Pack

Develop Anywhere: Deploy Everywhere

Modern mainframe software development occurs on both mainframe and distributed systems. To take full advantage of your mainframe development resources, the ChangeMan ZMF Client Pack provides multiple client interfaces. Choose your interface and use the Eclipse Workbench, the IBM Rational Developer for System z (RDz) environment, the ChangeMan ZDD Windows client interface, or the traditional Serena ChangeMan ZMF interface to develop and maintain your mainframe applications.



MODERN DEVELOPMENT ENVIRONMENT FOR THE MAINFRAME

ChangeMan ZMF Client Pack: Eclipse Plug-In

ChangeMan ZMF for Eclipse supports both the native Eclipse Workbench and IBM RDz environments. Now you can develop off the mainframe and continue to benefit from the version control, build management, release management, and deployment capabilities of Serena ChangeMan ZMF.

The sheer scale and complexity of enterprise applications on the mainframe require robust build management and software deployment capabilities across the entire enterprise. Serena ChangeMan ZMF has always satisfied these requirements for enterprise software repositories.

However, traditional enterprise programming languages are not ideally suited to the development of today's applications. Java has established itself as a preferred language for large-scale enterprise development. In addition, with so many prebuilt software development toolkits (SDKs), software libraries, code generators, debuggers, test tools, automated documentation tools, and other development tools available for numerous operating system platforms, Java has become a highly productive language for cross-platform and Web-enabled application development.

BUILD DESKTOP-DEVELOPED MAINFRAME APPLICATIONS SECURELY AND SEAMLESSLY

For these applications, ChangeMan ZMF and Serena ChangeMan ZMF for Eclipse now support the z/OS Unix Hierarchical File System (HFS) as personal development libraries and for baseline, staging, promotion, and production libraries. HFS libraries provide support for long file names and Java classpaths.

Further, Java productivity enhancements in ZMF for Eclipse include native integration of ChangeMan ZMF functions with the Java perspective of the workbench through the "Team" contextual menu. Eclipse integration is important because the Java developer community has largely adopted the open-source Eclipse Workbench IDE as its development platform of choice.

For the maintenance of legacy enterprise mainframe COBOL applications, including applications built with DB2, the IBM Rational Developer for System z (RDz) is a developer workbench that provides a wide-ranging set of development, debugging, and testing tools with a GUI ease of use.

ChangeMan ZMF for Eclipse enables developers to check out source code, JCL, associated DB2 plans, and any other artifact from a ChangeMan ZMF repository baseline into an RDz development project. Once developers are finished with the development and testing of the application code using RDz, they can leverage the rich set of ChangeMan ZMF commands to check in, freeze, and promote the changes on the mainframe. This is done securely and accurately every time, providing a complete audit trail.

ChangeMan ZMF for Eclipse provides a unified and configurable view of software resources in all personal development libraries on the mainframe, as well as baseline, staging, and promotion libraries in the repository. Compressed output listings can be read in the Serena perspective using the Serena-editor installed with the Eclipse plug-in. The z/OS Explorer view of RDz provides access to project resources, and job output in the JES job queue can be accessed in either view. With ChangeMan ZMF for Eclipse, you can save money, increase productivity, and develop modern mainframe applications.

ChangeMan ZMF Client Pack: ZDD

Using ChangeMan ZDD, you can browse your mainframe system as if it were a network file system. From a Windows desktop, ChangeMan ZDD provides access to data sets, job submission, job output, and to Serena ChangeMan ZMF components as though they were local files on your PC. And all your existing access and security is still enforced. No special execution or programming interface is required.

INTEGRATED TO LEVERAGE LEGACY MAINFRAME APPLICATIONS

Desktop development of mainframe applications is commonplace today. Powerful IDEs provide a productive environment for editing and testing today's applications.

Serena ChangeMan ZDD solves the integrity, security, and resource issues associated with desktop development by mapping the mainframe server as a drive on the desktop. This enables developers to access mainframe assets directly and immediately from their desktop.

ChangeMan ZDD integrates desktop IDEs into the software development lifecycle, providing direct access to mainframe assets, eliminating mass file transfers, and extending the protection of Serena ChangeMan ZMF to development on the desktop.

Because ChangeMan ZDD integrates intuitively into the Microsoft® Windows Explorer interface, it eliminates the need for mainframe training, which provides tremendous staff flexibility. The "edit in place" capability of ChangeMan ZDD prevents software components from getting lost or overlaid during manual upload/download procedures. Code can't be lost due to desktop crashes or laptop theft. Audit trails are kept intact Serena ChangeMan ZDD gives desktop programmers direct access to software assets residing on the mainframe without file transfers, and it extends its powerful enterprise change management solution by transparently integrating the desktop development tool of choice.

XML-ENABLED ACCESS TO ZMF ASSETS AND INFORMATION FROM POPULAR DESKTOP TOOLS

Serena ChangeMan ZDD is a software infrastructure technology that makes mainframe data sets and job output appear in Windows Explorer and other desktop applications as though they were local files or files on a Windows network. It simulates a network file system on a Windows platform that is networked with an IBM z/OS® operating system. From your PC, you can access files, execute jobs, and examine output from jobs that reside on a z/OS server. No special execution environment or programming interface is required.

ChangeMan ZMF instances, applications, packages, and libraries appear as folders in Windows Explorer and other PC applications. You perform any ChangeMan ZMF functions from your PC, and all of the software configuration management rules and restrictions set up in ChangeMan ZMF apply.

ChangeMan ZDD enables ubiquitous access to the mainframe from Open Database Connectivity (ODBC) and XML tools, including Microsoft Excel, Microsoft Project, and SAP Crystal Reports. In addition, it provides an elegant interface for submitting mainframe jobs, viewing JES input/output queues, and integrating mainframe test environments.

Serena ChangeMan ZDD, included in the ChangeMan ZMF Client Pack, provides integration with ChangeMan ZMF to enable efficient and secure development of mainframe applications on the desktop. Combine the flexibility of the Windows desktop with the governance of the mainframe.

For more information or to see a demo contact Serena Software at 1-800-457-3636.

Making the case for ChangeMan ZMF

Measuring the value of Software Change, Configuration and Release Management solutions is not an exact science. However it is possible to derive a representative figure that is indicative of the likely value that can be released through the implementation of a modern SCCM solution such as ChangeMan ZMF.

Method

The approach followed here is based on the Forrester Total Economic Impact (TEI) methodology. TEI articulates how products and services benefit users both in terms of cost and ROI.

In the Serena version of the TEI methodology, determining the value in terms that each stakeholder can understand and articulate is critical. It is necessary to place a monetary number to the value assessed as this is the true test of the precision of the calculated value.

No part of this method is designed to identify individuals, groups or functions that will be eliminated as a consequence of the identification of value. Rather it identifies the amount of each person's activity that is currently consumed unproductively. For most individuals this means returning time to them in each day so that they may carry out their assigned tasks with improved efficiency, quality, completeness and timeliness.

The test

To test how much value is being delivered it is important to look at each potential stakeholder separately and ask very simple questions:

Who is the beneficiary?
What are they getting?
How do they benefit?
Can they measure that?
Is there a dollar amount?

This is achieved by using a measurement statement for each stakeholder:

For **<stakeholder>**
we provide **<feature/function>**
to make **<activity>** better
as measured by **<KPI>**
which is worth **<financial measure>**
as calculated by **<typical effort and cost>**

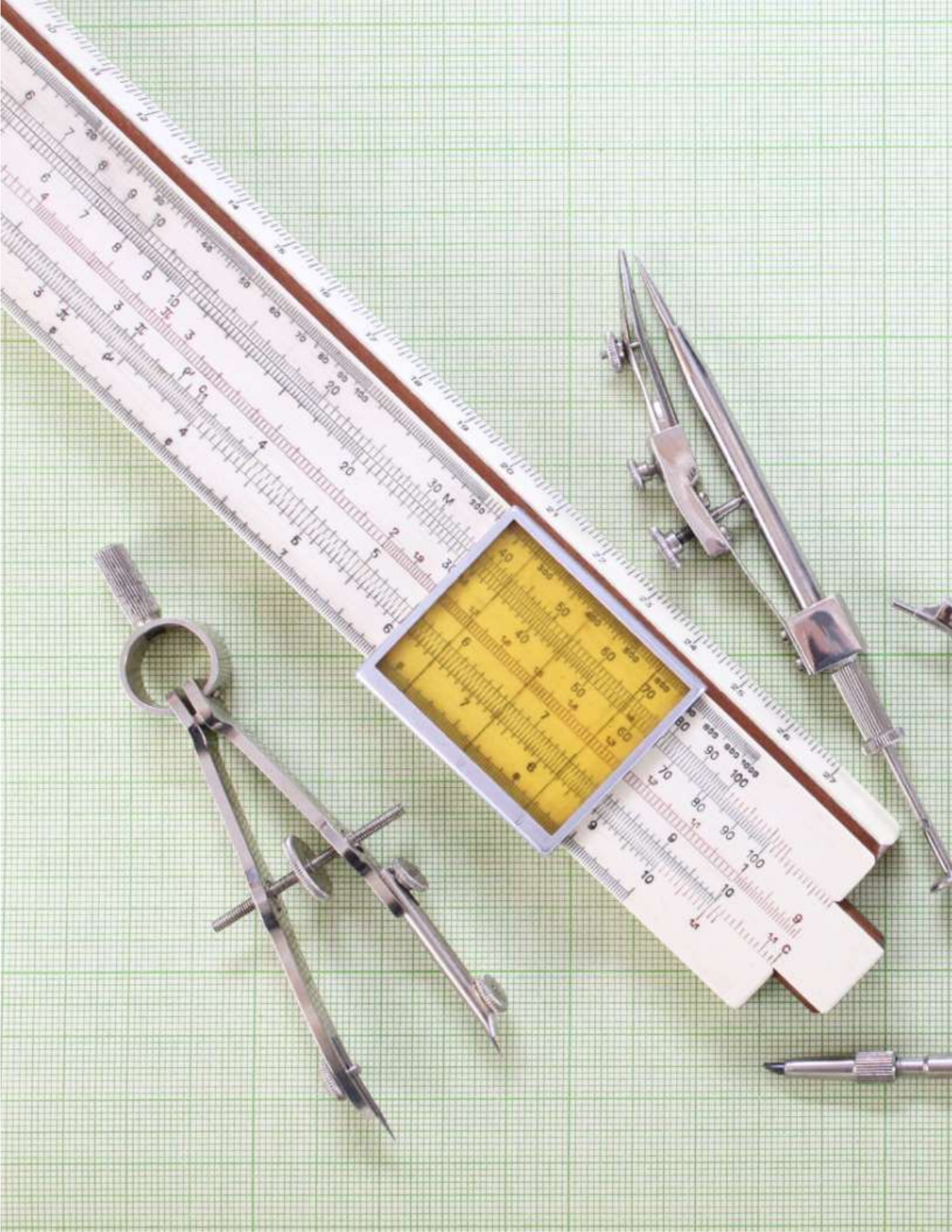
Example

When we think about the work that the troubleshooter faces at 3:00am; their main concern is working out what changed and fixing it. But in the process of doing that they may change code that someone is working for a normal, expected development task. This brings us to the complexity and danger inherent in doing parallel development. ChangeMan ZMF is a powerful tool that facilitates this kind of activity in the development team and prevents the kind of regressions that always cause outages. Here is how we would calculate the value of the parallel development features of ZMF.

As an example:
For **developers making emergency changes**
we provide **a fast and robust way to develop the product fix**
to make **parallel and concurrent development** better
as measured by **the total elimination of code regressions**
which is worth **$6,000 per code regression eliminated**
as calculated by **10 hours of diagnosis @ $100/h**
plus **$5,000 loss of opportunity due to an outage**

What follows are some of the use-cases that are used to calculate the value of using more modern and complete tools for SCCM, such as ChangeMan ZMF. They are given in example form only and there is room to enter your own data and do you own calculation. The numbers are based on what we have typically found based on 100 detailed assessments carried out in 2011 and 2012.

> **"After only six months of use, we are already experiencing benefits from ChangeMan. With very little training, we have seen administration savings with elimination of meetings and emails we needed to use before."**
>
> **Sr. Project Leader, Transportation Company.**



Software Developers

ChangeMan ZMF provides a rich set of tools for mainframe software developers. The modern mainframe development paradigm is changing. ChangeMan ZMF is the only solution that offers full function capabilities for developers whether they are on ISPF, Eclipse, Rational Developer for z/Series (RDz), Micro Focus and other off-host development environments.

Developers who do their development solely on the mainframe now have access to the leading SCCM+RLM tool that has been designed around their needs. Whether they are developing in COBOL, PLI or even Assembler, FORTRAN, REXX or almost any other language ChangeMan ZMF has been designed to make the process as simple as possible. If you are developing for CICS, DB2, IMS or other major functionality, once again, ChangeMan ZMF has all your options covered.

If you are now developing for execution on the mainframe platform but the development is being done off the host, perhaps in Eclipse or Micro Focus, Serena provides plugins that let you maintain the full effectiveness of a mainframe-based SCCM+RLM tool while exploiting to the full the capabilities of the off-host IDE's (Integrated Development Environments).

For all developers, irrespective of platform, the value of ChangeMan ZMF is easy to define:

- Native development environment (ISPF or IDE of choice)
- Full support of the organization's development process
- Support for waterfall, agile and hybrid methodologies
- Fully automated code movement through the lifecycle at the touch of a button
- Comprehensive code impact analysis improving developer productivity
- Automated detection of out-of-synch conditions preventing accidental production errors
- Bi-directional notification and management when two developers work on the same code
- Rich tools to assist with parallel and concurrent development

For **mainframe developers (typical figures)**

- we provide **a software change, configuration and release management environment**
- to make **parallel development, impact analysis, regression avoidance, editing and development** better
- as measured by **development productivity, zero regressions, improved collaboration & 100% accurate and complete releases**
- which is worth **$<** *1,000* **> per developer per release**
- as calculated by **<** *10* **> hours x $<** *100* **> per hour**

Software Quality Specialists

ChangeMan ZMF manages and controls test environments guaranteeing the integrity of the code deployed there. It has fully capable mechanisms for managing promotions to test areas on the development host-mainframe or to remote mainframes. ChangeMan ZMF tracks approvals through the development lifecycle and provides an audit trail of those approvals.

Support for the organization's process is paramount to ChangeMan ZMF and so the solution provides maximum flexibility to define and control test areas. Access to the test areas is controlled as are the roles that can promote code to the test area. Only executabled are deployed to test areas maintaining the separation of duties for highly regulated organizations. Even the sequence in which code can be promoted can be controlled.

For emergency changes or non-standard applications, as little or as much of the test cycle gets executed as needed but a strict audit log is kept and special approvals are required.

In addition ChangeMan ZMF automatically creates the new system's installation JCL as a by product of the management of the assets that are being changed and delivers this JOB to QA for testing.

The Audit function provides security for QA teams because it guarantees the consistency and completeness of the code that has been promoted into the test area.

For **mainframe testing team members (typical figures)**

- we provide **code promotion automation, automated install JCL creation and approvals management**
- to make **deployment to test environments and test management** better
- as measured by **100% complete, accurate and automated delivery of executables to test environments and guaranteed frozen code that cannot be changed by developers**
- which is worth **$<** *500* **> per tester per turnover**
- as calculated by **saving <** *5* **> hours x $<** *100* **> per hour**

> *"The number of people and applications we were able to bring up in the Serena ChangeMan environment during the first year was much higher than expected."*
>
> ***SCM Administrator,***
> ***Financial Services Provider****.*

Change Managers

There is no more critical role in the organization than that of controlling changes to the production environment. ChangeMan ZMF, the product names says it all, is specifically designed to support mainframe change management; and it has done so for the past three decades for hundreds of organizations and thousands of change managers across hundreds of thousands of releases.

The solution provides the three critical elements that make for effective change management:

- Visibility: into the project lifecycle and into the content of the change that is moving through the process
- Traceability: that automatically makes sure changes are tracked and logged against the initiating request and that are traceable throughout the lifecycle to their deployment as code changes.
- Control: controls to ensure that only authorized users are able to approve changes through the stage-gates of the organization's software development lifecycle (SDLC) and to ensure that changes happen only when they should

Change managers who use ChangeMan ZMF experienced unrivalled control over the request-to-release process. Built into ChangeMan is its own ticketing system but the open architecture of the product makes it simple to integrate with all the common mainframe and non-mainframe-based tools.

For **mainframe change managers**

- we provide **a software change and configuration management environment**
- to make **traceability of change requests to code changes to deployments with control and visibility** better
- as measured by **100% complete and up to date audit trails, 24x7 available status reporting, elimination of status meetings, manual reporting tracking efforts**
- which is worth **$<** *5,000* **> per change manager per release**
- as calculated by **saving <** *50* **> hours x $<** *100* **> per hour**

"We had a homegrown system that would not scale. With Serena ChangeMan, we could move a lot more changes without adding any staff even though we grew 30 percent. Most importantly, Serena ChangeMan allowed us to handle very large clients who helped fuel our company's growth."

Manager, Service Provider.

Release Managers

The complexity of modern systems is at an all-time high. Application-to-application dependencies and the labyrinth of infrastructure components makes deploying new code one of the most challenging tasks in our industry. ChangeMan ZMF has been deploying code to production systems for three decades in a repeatable, auditable and controllable manner: and backing out code too with the same precision and care.

Throughout the development, test, change and release lifecycle ChangeMan ZMF is monitoring and tracking the activities of the team. It builds up a profile of the application and uses this to create the JCL that will be used to install the changes. Because ChangeMan ZMF understands the application and its entire collection parts it is able to make deployment simple and effective.

The use of standard technologies means that ChangeMan ZMF can be customized very quickly to integrate with other mainframe tools and products. This means, for release managers, that they are able to orchestrate the whole mainframe deployment. Using ChangeMan ZMF's built-in calendaring solution, or a third party tool, it is possible to schedule deployments and have them triggered on completion of the final approval.

One unique feature is ChangeMan ZMF's ability to install code on a temporary basis. This allows organizations to put in a temporary fix and then have ChangeMan automatically remove it after a given time period. All the while with the same integrity, traceability and control that can be found in standard deployments.

If a release needs to be backed out, ChangeMan ZMF can do that too and return the system to its previous state. ChangeMan ZMF also returns the source code libraries to their corresponding state.

For **mainframe release managers**

- we provide **rich release management controls, visibility and organization tools with scheduling and deployment automation built-in**
- to make **management, reporting, approvals and deployment of releases** better
- as measured by **100% automated deployments, full release auditing, 24x7 visibility and control, elimination of status meetings, automated reporting, automated backouts, elimination of deployment scripting, guaranteed source-to-load integrity and internal/external-audit ready traceability**
- which is worth **$<** *2,000 >* **per release manager per release**
- as calculated by **saving <** *20* **> hours x $<** *100* **> per hour**

Project Leaders

Managing any team, no matter how small, requires great skill and attention to detail. From the beginning ChangeMan ZMF has been designed to support project team leaders by giving them the tools to manage their development and test teams in the most natural way possible.

The ability to break work into packages of changes allows the team, and the team leader, to manage project work in incremental units as small as a single change or as large as an entire release. The changes can be grouped and regrouped so that they can move together through some parts of the lifecycle, such as system testing, while moving individually through other parts, like unit testing.

The team leader has the ability to see exactly what code is being changed and by who and to see where the code is in all the release cycles that may be in motion. ChangeMan has unrivalled support for parallel and concurrent development so team leaders can be assured that code regressions will not happen because ChangeMan is facilitating the collaboration necessary amongst the development team.

A key feature of ChangeMan ZMF is the Audit Function which provides a check on the state of the project. Through Audit team leaders can see which projects streams are ready to go, which are lagging behind and which are running into serious issues. The Audit Function is unique and only available with ChangeMan ZMF.

For **mainframe project leaders**

- we provide **a complete, project-based, development environment**
- to make **developer work assignment and developer activity management** better
- as measured by **100% traceability between change requests, project tasks and impacted code**
- which is worth **$<** *1,000* **> per project leader per project**
- as calculated by **saving <** *10* **> hours x $<** *100* **> per hour**

> *"We reduced participation in our daily change meetings from 30 to three people because of ChangeMan [ZMF]. Whereas before we needed everyone making a change, now only the key people need to attend."*
> ***Team Leader, Retailer.***

Tool Administrators

Supporting modern infrastructure tools has become a significant part of the cost of development and support operations. As organizations grow the demand on the tool administrators usually grows along with it. Every tool introduces more features and complexity and begins to be integrated with more and more other technologies to the point at which it requires many full time staff to just keep the system on the air.

ChangeMan ZMF is a sophisticated and richly featured solution. It has a level of complexity that does require the skills of someone well versed in IBM mainframe technologies such as z/OS, JCL, Panels, IMS, DB2 and CICS. No proprietary language has to be learned in order to administrate ChangeMan ZMF.

Where other systems are becoming more complex, ChangeMan ZMF is getting easier to administrate. With almost three decades of use all around the world the ChangeMan ZMF development team have learned the kinds of things every customer customizes into the solution and they have made those core features that can be turned on or off. The exceptional open architecture, which is based on XML and Web-Services, means that it is easy to integrate third-party solutions and to do it in such a way as to minimize the problems that occur when one tool or the other is upgraded.

Cloning is a major strength of ChangeMan ZMF allowing administrators to take a model application and clone it for a new system. In this way it is easy for the same administrator to manage hundreds of applications, thousands of developers and tens of thousands of artifacts in the repository.

It is common for the same administration team that installs ZMF initially to be the one that administers it decades later even when the number of applications managed has increased several orders of magnitude: such is the ease of administering the product.

For **software change, configuration and release management tool administrators**

- we provide **a single point of user management, configuration management and reporting**
- to make **tool administration require minimal intervention or product knowledge** better
- as measured by **reduced overall administration time, elimination of user administration, simplified customization, elimination of re-customization of new releases, no special skillset or training**
- which is worth **$<** *10,000* **> per admin per application**
- as calculated by **saving <** *100* **> hours x $<** *100* **> per hour**

Value Calculator for Productivity and Efficiency

This is our simple Value Calculator. It uses the premise of the statements on pages 20 to 22 to estimate how much of your development, testing, change, release and operations staff effort is tied up work that can be done more effectively through automation by ChangeMan ZMF. Of course this is a simple calculator but it does give results that are remarkably accurate.

		Average number of personnel in each release					One Time Benefit
	Releases per year / Turnovers per release	Software Developers	Software Quality Specialists	Change Managers	Release Managers	Project Leaders	Tool Administrators
Major	*4 / 10*	*20*	*40*	*2*	*1*	*1*	*1*
Minor	*20 / 3*	*5*	*2*	*.5*	*1*	*.5*	*1*
Emergency	*250 / 1*	*1*	*.5*	*0*	*0*	*0*	*1*
Average hourly rate		*100*	*100*	*100*	*100*	*100*	*100*
Average hours saved per		*10* per release	*5* per turnover	*50* per release	*20* per release	*10* per project	*100* per application
Annual value saved by ChangeMan ZMF		*((4 x 10) + (20 x 3) + (250 x 1)) x 100 x 10* = *$310,000*	*((4 x 10 x 20) + (20 x 3 x 5) + (250 x 1 x 1)) x 100 x 5* = *$675,000*	*((4 x 2) + (20 x .5) + (250 x 0)) x 100 x 50* = *$90,000*	*((4 x 1) + (20 x 1) + (250 x 0)) x 100 x 20* = *$480,000*	*((4 x 1) + (20 x .5) + (250 x 0)) x 100 x 10* = *$14,000*	*Assuming 50 applications* *50 x 100 x 100* = *$500,000*
Total	*$1,569,000* Equivalent to *12 FTE*						*$500,000* Equivalent to *2.5 FTE*

This example, based on a large claims processing organization, shows the typical value that ChangeMan ZMF delivers to an organization every year in terms of improved productivity of the team. It is typical of an development organization of around 100 staff comprising PMO, development, QA, change and release management.

What these numbers show is that with the use of ChangeMan ZMF at least a 10% productivity gain can be achieved simply through the elimination of errors.

On the following page we offer some insight into how to quantify other benefits that ChangeMan ZMF delivers to organizations.

Note: that these numbers are generally indicative and have the correct orders of magnitude. For a more detailed study with precise calculations of the value you can derive please contact your Serena account representative or email sales@serena.com.

Release Management Team

From:	Risk Assessment Team
Sent:	August 16th, 2013 15:21
To:	Release Management Team
CC:	Development; QualityAssurance; DevOps; ChangeControl
Subject:	Still Need Convincing?

Bill,

In the heart of the recent financial crisis when, according to one commentator, it was “buy one bank, get one free” a number of mergers and acquisitions occurred in the New York banking community. At one of these banks they decided to standardize their retail banking on ChangeMan ZMF and their wholesale banking on a competitor tool. This meant migrating some applications from one of the legacy banks ChangeMan ZMF system to another mainframe software change management solution. And it meant migration other applications to from the acquiring bank to ChangeMan ZMF.

After a few months several things became very apparent.

- **All the teams that migrated OFF ChangeMan ZMF** to the other tool found that their work was harder to do, they did not have the support from the technology they had come to expect and the migration took weeks per application.
- **All the teams that migrated TO ChangeMan ZMF** found that they were more productive, things that had been overly complex before were now handled efficiently and the migration took a few hours.

But there was one thing that was completely unexpected.

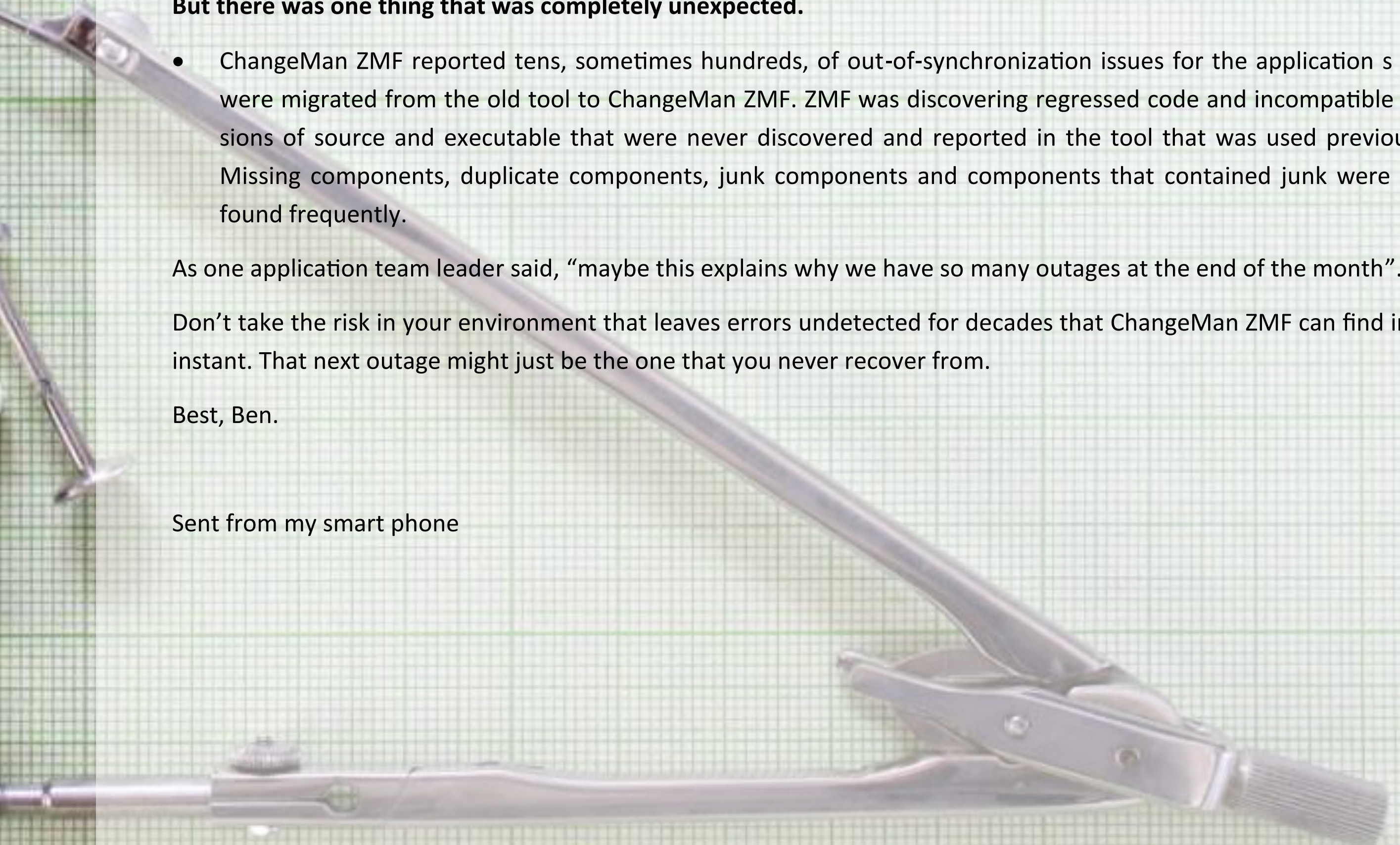
- ChangeMan ZMF reported tens, sometimes hundreds, of out-of-synchronization issues for the application s that were migrated from the old tool to ChangeMan ZMF. ZMF was discovering regressed code and incompatible versions of source and executable that were never discovered and reported in the tool that was used previously. Missing components, duplicate components, junk components and components that contained junk were also found frequently.

As one application team leader said, “maybe this explains why we have so many outages at the end of the month”.

Don’t take the risk in your environment that leaves errors undetected for decades that ChangeMan ZMF can find in an instant. That next outage might just be the one that you never recover from.

Best, Ben.

Sent from my smart phone



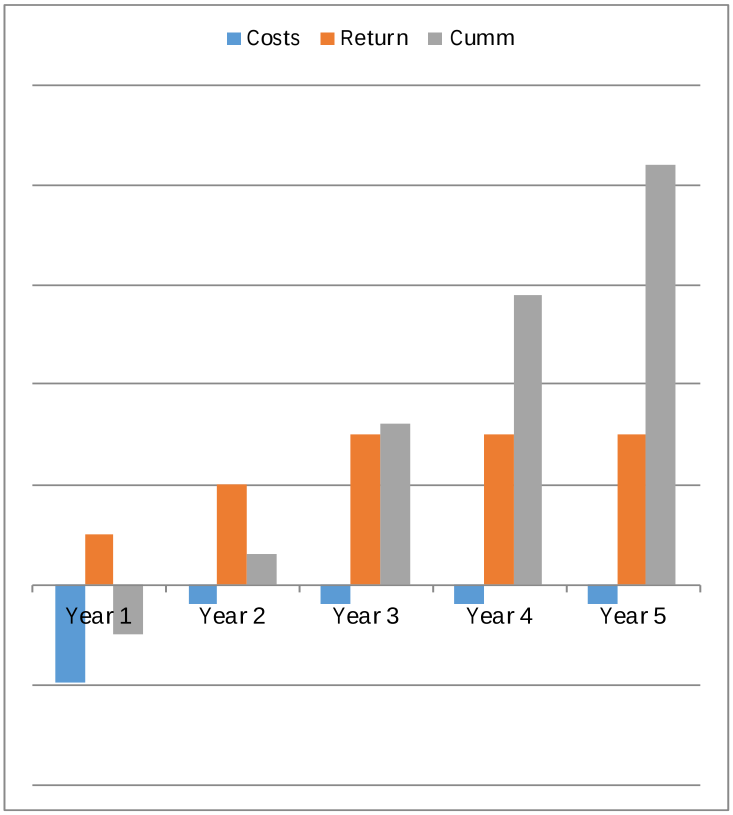
Return on Investment

Many other factors can be taken into consideration that deliver quantifiable benefits to the organization. Serena provides a full service offering to provide a detailed, documented return-on-investment case. With more than 30 years experience guiding customers in the selection of change, configuration and release management solutions we have good data to help your kind of organization. Numbers vary by industry, but we find that typically we can report:

- Payback periods are in the 12 to 18 month range
- Internal-rate-of-return between 75% and 150%
- Costs, in year 2 onwards, are less than current system

Customers frequently tell us that the prevention of a single outage will pay for the cost of moving to ChangeMan ZMF. Others tell us that the ability to pass audits in hours rather than weeks is where they see the value. Everyone talks about the improved programmer and quality productivity.

So, whatever your reason for changing, there has never been a better time to move to ChangeMan ZMF.



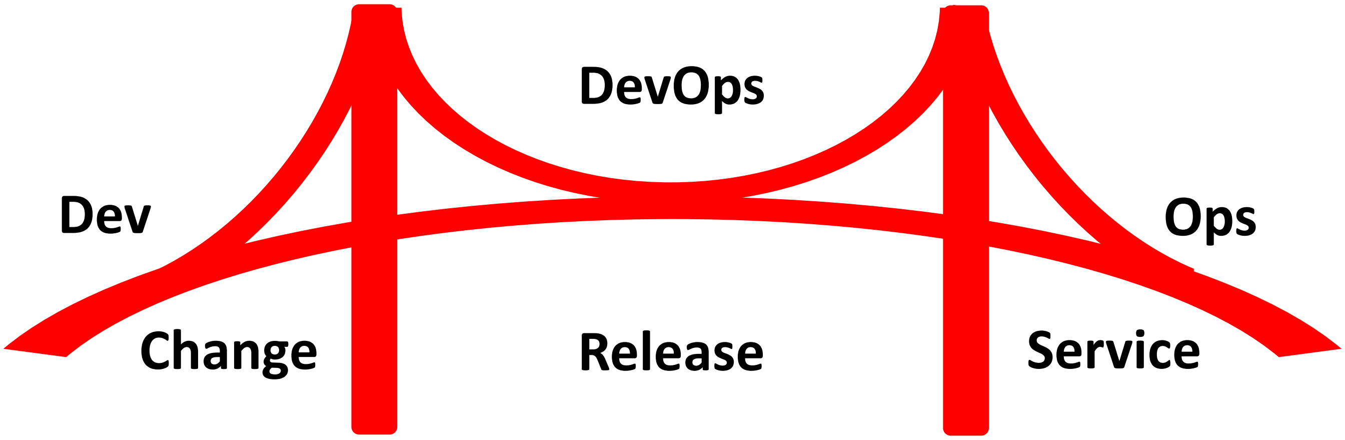
"For us, the savings were not the big issue – we needed control. In our business we have audits and regulatory controls that require us to better manage application changes."

VP, Financial Services Provider.

Enterprise Release Management: the bridge to everywhere

Enterprise Release Management

A unique offering from Serena Software



A new challenge for Release Managers

Never has **Release Management** received so much attention as it is getting today. The **DevOps movement**, the broad adoption of **ITIL principles** and the creation **IT Service Management** as a discipline within the organization: all these lead to the same conclusion, that Release Management is the fulcrum of IT.

Increasing development velocity, due to the introduction of **Agile techniques**, the pressure for **more value sooner**, in the oft quoted **time-to-market** imperative, is driving up the frequency of releases. Not to mention all too many organizations where the **fast-track, emergency fix process** is becoming the daily norm.

Reigning this velocity in, is the demand for ever more vigilance in how we change the environment. **Greater oversight** through **governance and audit** combined with tighter controls and security are all making the ability to change simply tougher to do.

When we add to this the unimagined **complexity** of a release of components that spans myriad of **technology platforms**, comes from multiple **development methodologies** and from developers and testers in **every time-zone** on the planet. No wonder release management has been receiving attention.

And there is one more mighty force in the modern world of release management that surpasses all of these for difficulty to manage and sophistication of the solution needed. Of course we are talking about the **application-to-application dependency** that is now part of every major release that we will encounter in the enterprise. **Sequencing these dependencies** in development, test and production has become a full time job for a new generation of Release Managers.

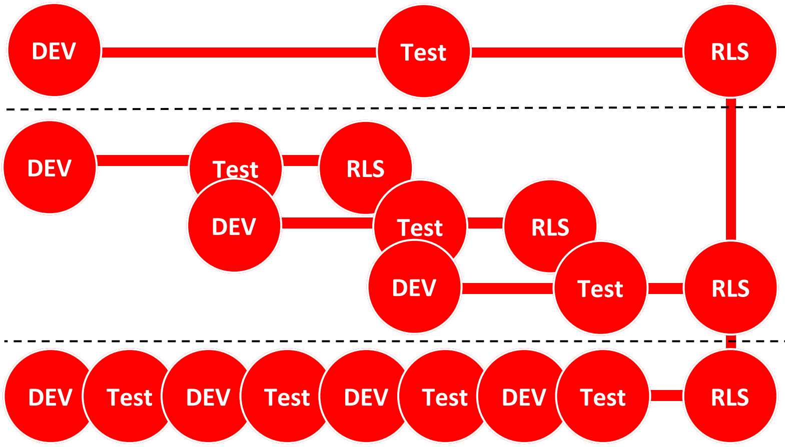
A new solution for Release Management

When we think about the critical elements needed to be successful at Release Management a number of key technologies are essential. The Serena Release Management solution provides all of them.

- **Process-centric:** It has to support your processes, your mainframe, your distributed, your agile, your waterfall.
- **Coordinating:** It must allow projects to move freely when needed and move in synch when necessary.
- **Collaborative:** It should ensure full, open communication between and amongst stakeholders with timely notifications in both “push” and “pull” models.
- **Calendar:** At the center of the tool it must allow all stakeholders visibility into the release calendar so that informed decisions can be made about the schedule, so that under– and over–running projects can be mitigated, so that release loads can be balanced and resource usage optimized.
- **Automated:** Wherever practical and possible the movement of the release artifacts must be automated, moving from development to test areas, from test areas to preproduction staging areas and into production with as little human intervention as possible.
- **Independent:** Neither source repository nor target topology should be proprietary. Enterprise environments are a heterogeneous collection of both development and execution platforms and the release management solution must be cognizant of them all. *(Continues on page 32)*

Enterprise Release Management

The cross-project coordination problem—Serena's unique insight



Traditional waterfall development team in a long project lifecycle mode as often found on the mainframe.

Modern overlapping waterfall development team in fast-iterating code-test-ship cadence. Mainframe or non-mainframe project teams.

Agile team developing in regular sprints but only releasing when needed. Mainframe (with ChangeMan ZMF) or non-mainframe teams.

Coordinating Releases Across Project Teams

No one is going to want to keep project teams in synch when they are on separate platforms, using different methodologies with wholly diverse business imperatives driving them.

Indeed, even if they are using the same platform, methods and business drivers, there is no need to keep them in synch unless they have to deliver code in a coordinated manner because there are cross-project dependencies.

One project many releases problem

Even in the daily ebb and flow of a single project there may be many releases in motion. Starting with, of course, the version of the project that is the current **production version**.

If there are bugs being found in the production code they may be being fixed with **emergency fixes** to selected components creating release of just one or two amended components.

The next **major release**, perhaps a quarterly release, will contain big new features to delight the business users. It will have hundreds of components being changed. And perhaps several of these are being developed simultaneously.

Then there are always **minor releases** with tweaks to the existing systems. These releases often contain many changes too.

It is not uncommon to have the very same component being changed in multiple releases simultaneously. This is why it is critical to have a comprehensive **parallel and concurrent development** tool in place to manage all of this complexity such as **Serena ChangeMan ZMF** and **Serena Dimensions CM.**

Having modern software change, configuration and release management solutions is essential in todays app dev community.

One release many projects problem

Enterprise releases are collection of many separate projects being managed, coordinated, together to a common date.

With the heterogeneity of development platforms it becomes a serious problem to be able to monitor the status of so many projects in a consistent, coherent and up-to-date manner without **interminable and numerous project status meetings.**

Visibility into the moving parts of a project has become the most sought after release management tool for managers.

But for change managers the issue is the **coordination of all the approvals and notifications** across the complex topology that is today's business and IT infrastructures.

The release manager needs to be able to balance the needs of the business to get new features into production against the IT resources available to make this happen. This requires a **release calendar** that shows the duration of projects and their **milestone dates** where they enter various **stage gates**. From this project running late can be identified and the impact calculated.

Unique insight, unique solution

What all this means is that release management is a process automation problem. You need a process for the macro milestones of your releases and you need a process for each of the your projects and their micro milestones. These macro and micro processes need to synch up from time to time but at other times can be free to operate at their own cadence.

Serena has, for more than 30-years been perfecting our Release Management solutions and providing them to the most demanding and the most agile customers in the world.

Serena® Release Manager

Solution Overview

Almost every organization depends heavily on its IT capabilities to support its success. New applications and services are essential to driving competitive advantage, maintaining services, and building new ones. Most businesses today are concerned about limited IT capacity, which threatens to limit growth. But existing IT groups can rise to the challenge, developing more relevant applications with greater complexity and reliability and bringing them online faster than ever before, provided that they have excellent release management processes and tools.

Award-winning Solutions for Release Management and IT Operations

A formal, automated release management process helps organizations maximize the value of their existing IT staff. Release management is an application lifecycle management process that guides IT efforts from application code development through testing and into production, helping to focus resources on timely delivery of the functionality that the business needs. Well-designed and comprehensive release management enables organizations to:

- **Improve visibility** into project timelines and progress to help eliminate surprise and keep all stakeholders on the same page
- **Increase release flow**, getting critical changes into production sooner with existing resources
- **Improve release quality** to increase the number of successful deployments and reduce downtime, whether planned or unplanned
- **Simplify compliance** to enforce consistency and traceability of changes

Comprehensive Solution for Application Release Management

Serena Release Manager is a comprehensive solution that allows you to manage the implementation of software changes into your production environment, whether you are deploying a new application or delivering a single patch for an existing application. Release Manager is designed to support the application release management process from demand to deployment.

Release management teams require formal processes for planning and tracking the steps of releasing applications, greater visibility into those processes, and a way to enforce stage gates. Only with these tools and processes can the team meet the expectations of the business, ensure the quality of the released applications, and protect the integrity of mission-critical production systems.

In contrast to most vendors, who usually want to rip-and-replace your existing solution, Serena takes a collaborative and consultative approach. Serena evaluates your current release management process, advises you on best practices that fit your organization's needs, and recommends tools that complement your existing stack of technologies and new processes more consistent with modern and future Release Management needs. Serena's goal is to help you make the most of the people, processes, and technologies that you already have so that your business can achieve its full potential.

Bridge Dev and Ops

Application development and IT operations groups both exist to support new and ongoing business requirements, yet the two are often in conflict. The top priority of application development teams is to create and launch new applications—and enhancements to existing applications—as quickly as possible. For IT operations teams, the priority is ensuring availability of production systems, and the greatest threat to the production environment is change. While IT operations groups must incorporate new applications and changes into the production environment, they proceed slowly and cautiously.

The enthusiasm of application development groups and the caution of IT operations combine to create a backlog of changes and new applications waiting to be released into production—and a backlog of unmet business needs.

DEPLOY APPLICATIONS INTO PRODUCTION FASTER

RELEASE MANAGEMENT: HELPING IT DELIVER THE RIGHT APPS AT THE RIGHT TIMES

IMPROVE BUSINESS AGILITY AND INCREASE REVENUE

Release Manager Product Datasheet

Mainframe DevOps: we're ready already!

"The more things change, the more they stay the same" goes the old French proverb. Since we moved computing from the air-conditioned silo on to the air-conditioned desktop we have been inventing, and re-inventing, better ways to deliver business information. All too frequently we have seen the same growing pains repeat themselves as new technologies stumble and falter over the same issues the previous technology stumbled an faltered over too.

How many innovations can you name that are truly a product of micro-computer revolutions? How many were identified and solved several decades ago by the mainframe community? The irony of the tablet is that it is really a dumb-terminal in a pretty wrapper but relying for real computing power and access to big-data from some remote computing source. Tablets are certainly and more convenient than the monster 3270 terminals of yesteryear but they are, nonetheless, merely places to express data from a distant source and place to enter data for processing at a distant source.

We've come full circle, again.

The tempo of development has reached a dizzying cadence causing code to pile up awaiting deployment at the door of the datacenter. Meanwhile the pressure on operations for strict governance, unending uptime and ubiquitous availability is straining the infrastructure and we all know one more change might bring it crashing down. So in a universe where these worlds collide there is bound to be friction.

So the latest innovation, DevOps, is born out of the realization that the needs of development and operations should focus on what is common rather than what is adversarial. This takes organizational maturity to even consider it and takes executive will to make it happen. The exciting news, some may think it a curse however, for anyone in Change and Release Management today is that there is a focus on your profession. Never has the ability to deploy code had more business scrutiny, been more of a C-level strategy conversation or become the primary target of IT automation at the expense of other projects.

But just wait one minute!

For more than four decades the mainframe community has lived this life. The number of users impacted by change stretches into the hundreds of thousands, even millions, for the large banks and Telco's. So getting change and release right has been a major focus for both the mainframe development and mainframe operations teams for a long time.

Our new generation of DevOps practitioners can learn much from mainframe change and release managers. It starts with the shared understanding of the impact of change and the shared responsibility to ensure that change happens without drama. But it also comes from mutual respect for the roles involved and for the clear delineation of the boundaries of those roles. Much of that delineation is enforced through the mainframe security system with access control very tightly limited.

Technology is important too. Mainframe change and release management solutions have had 40 years to evolve to where they are today. For many shops they have comprehensive vendor solutions that manage the change, development, testing, release and deployment lifecycle. Others have a combination of vendor and homegrown and, just a few, have completely homegrown infrastructure. The point to note though is that no one tries to release code into production on the mainframe without some level of automation and significant amounts of control infrastructure.

And we are not talking about spreadsheets here. Many mainframe organizations use spreadsheets to back up the controls they have implemented but no one hands off that level of responsibility to what is, nothing more grand than, documentation. Beware of thinking that because it is written down it is a) the truth, b) up to date or c) anyone is going to pay attention to it. You must have mechanisms that enforce the practices, policies, procedures and processes of the organization as well as ones that document them. This we learned on the mainframe, the hard way, decades ago.

Thirteen years ago the mainframe faced its greatest challenge. Almost every piece of code that had ever been written up to that point faced some form of change and the go-live date was not something that could be moved. Y2K brought about a new industry, off-shoring, almost overnight and created a middle class in India and in China. Change on this level had never been seen before and when it was all over the only sound you heard was the press complaining about what a monumental disappointment it was because nothing happened. Precisely the point: nothing happened because the million or so mainframe professionals, who did not sleep that New Year's Eve, put in place the process and technology infrastructure to make sure they didn't become front page news.

Since then, the scourge of terror, the collapse of the banking industry, the nightmare of privacy, the blight of hacking, the fear of security breaches and data exposure, and a dozen other disruptions, has driven change into the way we deliver business systems. At each stage though the mainframe has been able to adapt without experiencing the distraction these events usually create. The reason for this is simple and it goes back to those well-defined roles we spoke of before.

So if you are starting up a new DevOps initiative go and talk to your mainframe team. They have already solved this problem. They know what it is like to deal with the competing priorities in your organization and have been doing so for decades. Their approach is one that was hard won and one that suffered many setbacks along the way. Their customers have been the hardest taskmasters and the most vocal critics. The systems that they develop, support and manage are the backbone of your organization and critical to its success. They will appreciate being asked. They will share what they know. They will offer to help.

DevOps is about nothing unless it is about attitude. Sloughing off preconceived ideas about how it should be and collaborating to create how it needs to be. This is the way of DevOps: this is the way of the mainframe.

First published in The Cloud Developer Review, June 2013

Enterprise Release Management: the Top 10 Myths

1. One process fits all

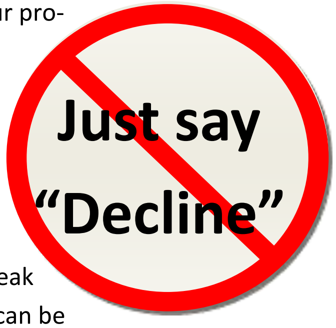
“We need one process for all release management!” is the guiding principle of so many organizations that are trying to implement a modern release management infrastructure. The reality is that there needs to be many processes. One main, high-level process that defines the major milestones that we measure our progress against, the macro milestone process. And many minor processes that accommodate the various factors that guide each development team such as its technology, time-to-market pressures, risk aversion, development methodology and project complexity, the micro milestone process.

2. You need one repository

Apart from the sheer impracticality of the statement, anyone who suggests that you can only do release management if there is one repository is just not trying hard enough. Of course having one repository is easy to say—it’s just not easy to do. Migrating all your code from the developer’s repository of choice is error prone and usually results in unwanted compromises over things like how many revisions can be kept. Retraining the team is expensive and re-tooling the team can be prohibitively costly. What you do need is the ability to coordinate the activities of the teams irrespective of the repository they use. You need to be able to move the code from their repository to the test areas repeat-ably and automatically and safely deploy to production. You don’t need to disrupt your development organization and spend money on solutions they will resent using.

3. You need one solution from one vendor

No one vendor has the best-in-class solution for all of your release management needs. Who has the best repository technology, or the best parallel development capabilities, the best support for Agile or DevOps will always be a matter for conjecture and disagreement. The point is that an organization needs to select the best tool for the job and needs the vendors to agree to make sure they work together. But beware, point-to-point integrations are fragile. You need to make sure your vendor is exposing their API through web-services and that the integrations supports your process and does not impose the vendor’s process.



4. Project status meetings are essential

Project meetings are a monstrous waste of time and resources. One customer describes the weekly release meeting as their “million dollar meeting” as it requires 70 members of staff, including several very senior members, to be in attendance for more than 4 hours. Each person gets to speak but it is often little more than “I’m good.” Imagine the time and money that can be saved by eliminating meetings alone through the use of a good process-centric workflow tool that guides everyone on the team through their part of the overall process. Status meetings can now become about exceptions and only involved the stakeholders who are impacted.

5. Release management is just about deploying the code

Well, actually, that is called “deployment”. Release management has many definitions as to the scope of the lifecycle that it covers. One useful definition says that a release management begins when the release is given a name and ends when the name is no longer used. For example “the fall marketing release” might start in planning long before any code is checked out or worked upon. Nonetheless it now needs to be tracked, deployment windows identified in the calendar, the resources applied and so on. Whatever your definition of release management your infrastructure should support your lifecycle from when you define the start point to where you define the end point.

6. Not every change needs to go through the release management process

Every change needs to go through *some* process of verification (testing) and approval. It doesn't, necessarily, mean though that every change goes through the *same* process. Emergency fixes needs a fast -track process with minimal stage gates and approvals. Major releases need more rigorous processes and more complete stage gates.

7. Putting release management infrastructure will slow down my project

Imagine if there were no air traffic controllers: would the free-for-all that ensues be good for on time departures and arrivals, would it be safe, would stakeholders be happier, would it be quicker? Of course not. In fact the point of ATC is to improve the throughput of flights to maximize the use of the runways and to do so safely. The same is true of release management. Without the infrastructure in place you will never be able to deal with the volume of changes, the complexity of the dependencies between releases and the competing needs of the many stakeholders. With the right release management solution in place you can increase the number of releases you deploy, improve control and governance, eliminate errors and downtime and do it with fewer resources.

8. We cannot make our releases any smaller

A major reason why release management has become more complex is because the size of the releases has increased to a point now where they are often bigger than the original system they are based upon. This size means that the releases are very difficult to test and the inter-dependencies of other changes in other projects in the release make it nearly impossible to have any confidence in the testing outcome. Many organizations are taking a leaf from the Agile-playbook and moving to smaller, incremental releases more often. By breaking the release into dependent and non-dependent changes it improves the testability of the code and the deployment is no longer held up waiting for other changes. Also moving to thematic releases, keeping the changes to a small area of the code base also improve the ability to test and deploy with confidence. But many more releases cannot be achieved without automated infrastructure in place.

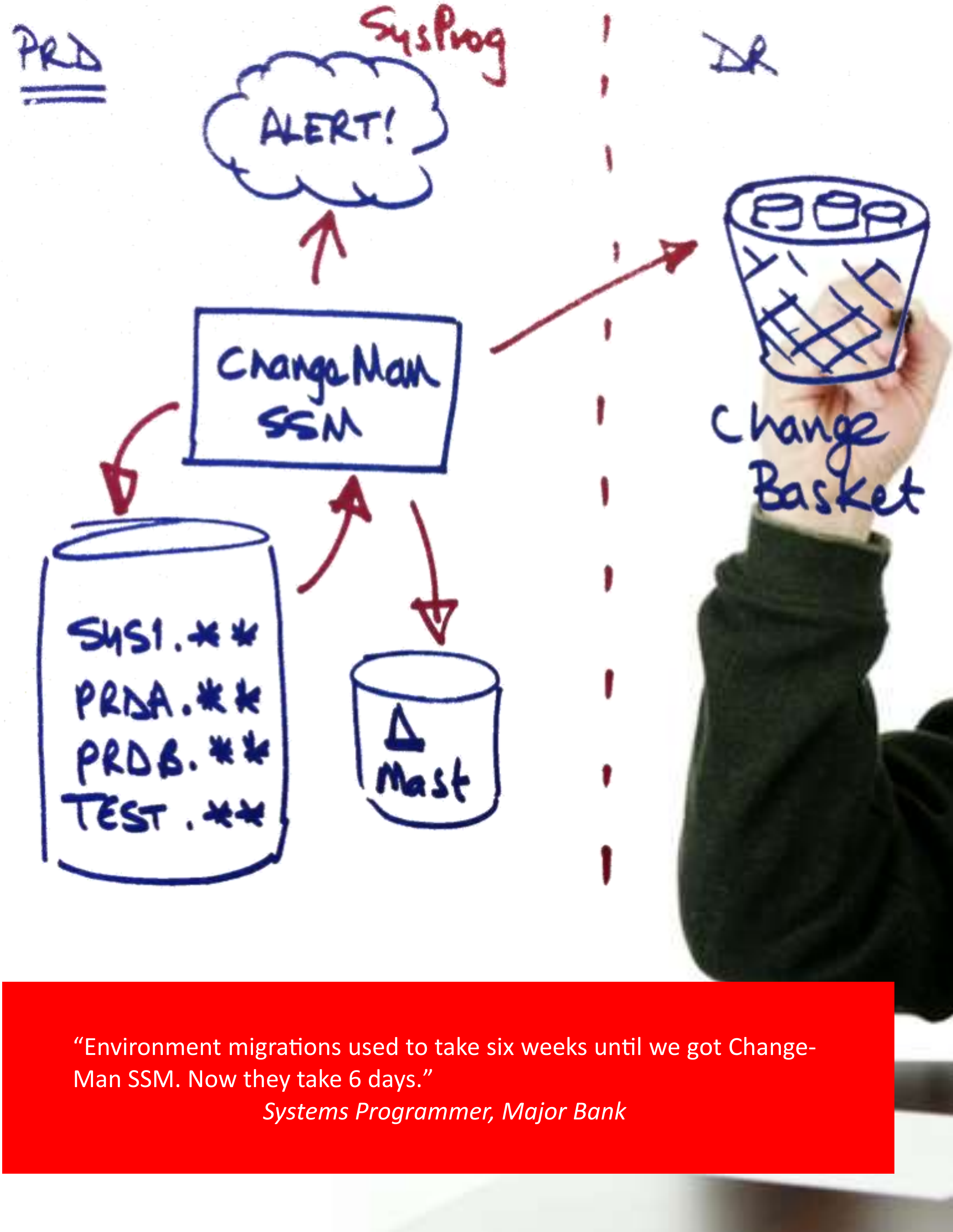
9. The business wants us to change things less frequently

No they don't. What they want is that the changes are deployed more successfully. Think about your smart phone: you get updates every day there and you have become the release manager on your own device. You now update without caring because you know the changes are small, unlikely to disrupt the functioning of your device and are, in short, safe. We need to get the business to have that same confidence in our releases. Small changes more frequently are easy for the users to absorb, have a smaller impact, require little training and are low risk. The business gets to prioritize the changes they want and constantly adjust those priorities almost up to the point we put things into production.

10. Developers won't accept having their access to production being removed

With the requirement for the "separation of duties" now being the law, the very idea of anyone having access to the production area seems very 20th century. Still, in many organizations, for whatever historic reasons, many people have access to the production areas and they use this privilege wisely and carefully. But the time for this has to stop now. As we have seen the addition of a layer of process automation improves the speed of the release process but it also improves the audit trail. In an emergency outage situation the first question anyone asks is "what changed?" Unapproved changes, even those with the best of intentions, are often also undocumented changes and even if they are not the cause of the outage the inconsistencies they represent hinder and delay the analysis and remediation of the problem. If someone needs access then ensure there is a process to let them have it, with appropriate approvals and a mechanism to withdraw it when the need has gone away.

For more release management best practices please contact Serena Software at info@serena.com



"Environment migrations used to take six weeks until we got Change-Man SSM. Now they take 6 days."

Systems Programmer, Major Bank

Outages:

40% operator error

40% application errors

20% technology failures

Source: Gartner

Tracking & synchronizing system changes

Even in the most controlled mainframe environment changes occur that we don't expect. There are any number of reasons for this, from accidental changes, like a wrong DDNAME so software changes datasets that were not intended to be changed, or mistyping the a dataset node name or a member name. To malicious and fraudulent changes designed to cause the corporation harm. With the right access controls in place these issues can be mitigated but still changes occur.

What is most worrying though is not noticing and not knowing that it has happened. Perhaps our first inclination that something is not right is a production outage. What would be worse a system going off the air or a system doing the wrong thing undetected for a day, a week, a month?

SYS-ONE-DOT-SPLAT-SPLAT

To manually check every system dataset would be tedious at best and error prone at worst. What if there was a way to track every member of every critical dataset, like SYS1.**, CATLG.**, PRDA.MASTER.**, and have an alert raised to the systems programming team when an unexpected change occurs?

Even better, what if the changes were recorded so that the author of the change and what they changed is logged in a database so that, if needed, the change can be reset to its original value?

This is just one use case for ChangeMan SSM.

Accessed never?

Well if you can track when something is changed you can track when something isn't changed, not even accessed.

How many datasets are there out there, how many applications, how much DASD is being taken up, by systems that are never executed. If you could show those applications and their datasets and see that they had not been used for a long time you could safely recover that space.

That alone can delay buying more space and save a considerable amount of money.

Installing for ever

Ever had to install a product? On more than one machine? By the third time you do it you are pretty good at it and wondering why you just can't copy the changes over instead of doing all the install all again.

If you take a ChangeMan SSM snapshot, a baseline, of the product datasets, do the install, and then use ChangeMan SSM to identify the changes you can then just ship the changes to each environment you need. And, of course, ChangeMan SSM will do that for you too.

The answer is yes, what is the question?

ChangeMan SSM does two things and it does them very well.

First it is able to track changes to any dataset and record what those changes are and who made them. It can let the systems programmers know when that happens in real time too.

Second it can take those changes an apply them back to the original datasets or to some other datasets in another environment on demand. It can do one change at a time or a whole week's worth of changes in what we call a "change basket".

So if tracking and synchronizing data is any part of your job … ChangeMan SSM needs to be your technology partner.

Serena® ChangeMan® SSM

System-level change tracking across z/OS systems

CHANGE TRACKING

ChangeMan SSM maintains a database of all changes to designated z/OS datasets and members. This single product supports hierarchical file system (HFS) and non-HFS data sets. Changed data is packaged and efficiently transmitted to a disaster recovery or business continuity site.

IMPROVE MAINFRAME GOVERNANCE AND COMPLIANCE

Serena ChangeMan SSM enables you to deliver business process applications while maintaining visibility and control. Using a unique fingerprinting technology, SSM will detect, track, and store changes to the system and document this for audit purposes. All changes to critical libraries are under version control and can be backed out – which allows the system to be returned to a compliant state. Additionally, SSM can collect changes and synchronize the changes with remote environments.

RAPID DISASTER RECOVERY AND BUSINESS CONTINUITY

ChangeMan SSM maintains a database of all changes to designated z/OS datasets and members and supports both Hierarchical File System (HFS) and Non-HFS datasets. Changed data is packaged by increments and efficiently transmitted to a disaster recovery or business continuity site. The unique ability of ChangeMan SSM to synchronize environments and automatically identify changes is perfect for ensuring that hot-sites are immediately ready to take over production application support in the event of a massive production site failure. Unlike hardware-based disk mirroring solutions, logical corruptions caused by human errors or software failures can be prevented by configuring a delayed APPLY at the standby site. ChangeMan SSM also takes care of all file attributes to keep your disaster site in sync.

INCREASE APPLICATION AVAILABILITY

ChangeMan SSM provides details about all software infrastructure changes. With SSM, you know what changes were made, when they were made, and who made the changes. Also, you are assured that every change can be backed out reliably and quickly.

ELIMINATE REDUNDANT SOFTWARE LICENSES AND DUPLICATED DATA

The Member Reference Tracking (MRT) feature of ChangeMan SSM reduces wasted disk space by identifying duplicated and unused Partitioned Dataset (PDS) members. The “Zero Reference” report identifies datasets and members that were not referenced during a specified time period. By tracking program product usage, you can identify opportunities to reduce software costs.

REDUCE PLANNED DOWNTIME THROUGH AUTOMATION

ChangeMan SSM reduces downtime by automating the shipment of SMPE target volumes to multiple systems. This assures that only tested changes are applied to target systems, thus minimizing downtime for servicing system images. ChangeMan SSM is not limited to system datasets; it can also distribute application images and application parameters.

ChangeMan SSM Product Datasheet

International Bank

COMPANY PROFILE

The banking system is one of the most highly regulated in the world. Nowadays the consequences for non-compliance result in heavy fines and imprisonment.

When you are also one of the world's largest banks too, the rate of change through merger and acquisition, and the adoption of modern technologies to meet consumer demand means that controls and personnel are constantly changing.

COMPLIANCE AUDIT

In the middle of one of these large acquisitions the internal audit team decided to check on the governance procedures in the systems programming area. What they found was excellent processes and tools in place for authenticating, granting and tracking access. They also found regular testing of the security systems and strong disciplines around both physical and digital access mechanisms.

They did make one audit finding though. They found that when vendors sent in patches, or product updates, or when vendors recommended configuration changes to their products, there was no change request process with approvals and notifications and there was log of the actual changes the vendor's installation processes made.

Though the risk was small, the fact that these changes were happening on a production system, that contained live customer data, made this a critical matter for the business.

TRACK AND SYNCHRONIZE

As a long time ChangeMan ZMF customer they turned to Serena for help. They knew that they had full control and comprehensive audit trails for all the changes their own developers were making. But what about third party software?

Using ChangeMan SSM they were quickly able to set up profiles that monitored each set of production libraries used by each vendor, including IBM. When a request to install an upgraded version comes along they now use SSM to create a baseline definition of the current version. They then apply the maintenance to the product and use SSM to report on what has changed. This report gets added to the audit information presented.

In addition SSM is monitoring a number of critical system datasets to make sure that the changes are not spilling over into them without anyone knowing.

BIG PICTURE

From operating in a world hoping that nothing was changing that shouldn't, the bank has now moved into an area where they are fully aware of every change that is occurring and they check daily the SSM logs of what has actually changed against what was supposed to change.

Recently they have extended the use of SSM to help them with migrating newly acquired banks and are also using it as part of their disaster recovery solution.

This bank certainly knows the value of ChangeMan SSM. From one audit finding the solution is now key to securing their environment from unexpected changes, an essential part of the disaster recovery mechanism and central to the merger and acquisition methodology.

Customer Success Story

15mm

Meet the StarTool Family of Products

Introduction

You know that feeling you get from having just the right tool for the job when you're at home tinkering under the sink? That mobile app that exactly meets your needs for tracking your stock portfolio is one you turn to every day. Then there's that trick you know for getting a movie on to your tablet for your next plane trip.

We all have our favorite ways of doing things and we've collected a number of tools and technologies to help us do them. We're reluctant to part with them and often carry them forward from house to house we live in, and from computer to computer we use daily.

Imagine if all those cools tools you've found, all the cool tools every one of your peers have found, were all collected together in one place.

Welcome to the StarTool family

From the beginning, StarTool set out to be the most complete developer's toolkit every assembled. The two main products in the family are StarTool FDM and StarTool DA.

StarTool FDM, or File and Data Manager to give it its full name, is the most complete integrated development environment for mainframe developers that has ever been seen.

The list of features is almost endless and to pick on just a few does disservice to the rest. Here are some of the main ones.

- Editing DB2 tables and rows directly
- Editing IMS segments with copybook overlay directly
- Editing VSAM records with copybook overlay directly
- Dataset search by partial member name
- Dataset search by member content
- Dataset search across multiple high-level qualifiers
- Compress a dataset while still using it
- Select a subset of a dataset as a test file

And hundreds of other features

StarTool DA, or Dump Analyzer, is used when an application ABENDs. StarTool DA intercepts the ABEND and converts the dump dataset into a readable diagnostic report which shows the source code of the program, the data content of all the variables in the program, the line where the error occurred, an explanation of the ABEND code and a suggested remediation for the type of error that has occurred.

If the customer is also using ChangeMan ZMF then StarTool DA opens up a ZMF Package, checks the component into the package and starts and edit session at the line where the error is indicated.

For the troubleshooter this saves many hours of analysis and diagnosis and searching for the correct version of the code to remediate.

StarTool IOO is the third member of the StarTool family. StarTool IOO, or Input-Output Optimizer. Dataset access is by far the slowest part of any computer process and has the biggest effect on the overall elapsed time a program takes to execute. Tuning the IO block sizes to optimize the data access is a difficult job that is subject to too many variables. Besides, we only get to choose this when we either write the code or define it in the JCL.

StarTool IOO has the ability to tune the setting for hundreds of programs and datasets simultaneously in real-time during the execution of the those programs. It shifts resources from program to program quietly and efficiently so that each races through its assigned data in the shortest possible time. All the while StarTool IOO is balancing the resources to maximize the process power of the machine.

By using StarTool IOO it is possible to shrink batch windows, often by several hours, and thus postpone the data when a machine upgrade is needed.

In a class of its own

Serena is most proud of our **Comparex** product. It was the first product we created and continues to be one of our strongest solutions in the mainframe market.

Comparex compares any mainframe data source to any other. It supports QSAM, VSAM, IMS, DB2, HFS, USS, XML and a number of 3rd party data sources too such as IDMS and ADABAS.

Its high performance algorithms look ahead to identify upcoming data and it optimizes its own execution accordingly.

Though not strictly part of the StarTool family, it is one of the most invaluable production tools available in any organization.

StarTool FDM Product Datasheet

Serena® StarTool® FDM

File and Data Manager

Today's z/OS data centers operate under competitive business pressures and challenging budget limitations. "Do more with less" represents a standard management expectation rather than a temporary austerity program. And Serena® StarTool® FDM puts today's IT on the road to really doing more with less with a wide and unique set of data management capabilities.

Serena StarTool FDM provides a unified data management workbench for z/OS that can be employed across the entire data center. With StarTool FDM, users can edit and manage PDS, PDSE, sequential, extended sequential, DB2, IMS, and VSAM data – as well as their file characteristics – with one comprehensive tool. StarTool FDM simplifies complex data and file management tasks. The need for multiple tools, batch utilities, and maintenance - intensive custom routines is eliminated. Application and systems programmers, operations specialists, production control, and any staff who use ISPF quickly become more productive.

Features

- StarTool FDM edits a wide range of data set types: sequential or flat files, partitioned data sets (PDS) and extended partitioned data sets (PDSE), VSAM clusters, and IMS and DB2 databases
- StarTool FDM manages both the data in the file and the control characteristics of the file
- StarTool FDM accepts customer-specified copy books in COBOL, PL/1, or assembler language to PL/1, or assembler language to describe and format file contents
- StarTool FDM supports bulk file processing in batch mode, tailored by commands in JCL.
- Key commands work in both ISPF and TSO line command mode for maximum interface flexibility

Benefits

- A unified tool to edit mainframe data sources minimizes learning time, maximizes productivity, and eliminates the cost and complexity of managing multiple products
- Delays for file reallocation and copying are avoided
- Programmers see and correct data fields using familiar names, without error-prone offset computations
- Complicated member or data set manipulations can be automated in batch processing
- Productivity gains happen across the entire data center

Working With Data

Serena StarTool FDM provides all the file editing flexibility one would expect in the z/OS environment. But file editing is only one facet of FDM's value.

With Serena StarTool FDM, users can easily customize views of data set, member, and file lists. The key to its flexibility is the FDM member list – the set of members or data sets which are of interest to a programmer. FDM combines data from multiple sources, including the systems catalog and VTOC, to provide more than fifteen types of information about the status and characteristics of data sets or members. Programmers can add or remove entries from the member list, merge in entries from another member list, and otherwise isolate the data of interest.

Additionally, StarTool FDM includes many facilities needed for working with load modules or program objects. As an example, FDM displays the link-edit date of load modules in a member list, and provides unique commands to describe the history and other information about a load module.

Further aiding productivity, StarTool FDM member lists can be named and saved for recall in a later session. This is a valuable productivity contribution for multi-day research projects, to identify one use case.

Serena StarTool FDM offers one-step solutions to the most common problems that applications and systems programmers face when working with PDSes.

FDM can expand PDS directory blocks and space allocations "on the fly," restore deleted or previously edited PDS members, verify and fix physical and logical errors in members (if necessary, altering the DSCB), and rename data sets, even if the data set is locked by an ENQ (data set enqueue).

The StarTool FDM IMS and DB2 options also browse or edit IMS data bases and DB2 tables with an easy-to-use, ISPF-oriented user interface. The IMS option even edits non-keyed segments.

Advanced Functionality Even Outside ISPF

With Serena StarTool FDM's standard Batch Facility, large data and file management jobs can be run under TSO native line mode in batch, rather than interactively through ISPF. The Batch Facility allows users to use standard Serena StarTool FDM commands and sub-commands as input to a batch processing job. With Serena StarTool FDM, large one-time jobs and repetitive jobs can be run more efficiently from saved JCL and control streams.

This uses TSO in batch (IKJEFT01) rather than ISPF, removing the need for manual intervention in complicated repetitive processes. Most FDM commands are available in both ISPF and in TSO line mode.

The standard Started Task Facility in Serena StarTool FDM provides a unique method for accessing and altering data sets, and their contents, without the availability of VTAM, TSO, or JES. Using MVS's subsystem interface, this ability for systems programmers to communicate through a z/OS console to perform line mode commands makes it feasible to make changes that can recover an otherwise unusable z/OS system and prevent the need for an IPL.

Load Module Expertise

FDM's extensive load module management facilities allow users perform complex tasks in a single step. With Serena StarTool FDM, users can interrogate a single module or selected subset of modules to assist in identifying properties, such as COBOL compile attributes and IDR DATA. The load module data can then be used to generate JCL to re-link a module, or disassemble modules for which the source code cannot be located.

Systems programmers can also use this data to generate SMP/E control statements, assign member aliases, or alter load module attributes without relinking modules. FDM can display all types of data associated with a load module, search for a load module by name, attributes or contents, and process selected modules with a wide variety of built-in tools.

One used case is tracking down missing link-edit JCL, which may be a time-consuming and frustrating chore. Serena StarTool FDM simplifies this often tedious task by providing tools to recreate link edit JCL and control cards using only the load module itself as input.

The FDM MAP subcommand reveals the structure of any member or group of members giving a list CSECTS. This gives immediate insight into the makeup of a load module and its subroutines. Identifying programs which use an obsolete version of a subroutine is a typical use case for the MAP or HISTORY subcommand capabilities.

Integration with ChangeMan ZMF

Seamless integration with Serena ChangeMan ZMF quickly and accurately identifies the latest application code changes and listings. This speeds up the process of solving application availability problems associated with outages or poor application performance.

This integration allows you to view the change history to see why a change was made, who made the change, and when it was made. And it capitalizes on the change management investment your organization has made with Serena ChangeMan ZMF.

StarTool FDM Product Datasheet

Serena® StarTool® DA

Dump Analyzer

Intelligent Dump Analysis and Debugging
StarTool DA provides intelligent dump analysis and debugging for mainframes. It helps developers diagnose the cause of ABEND conditions through automated dump diagnostics, interactive call tracing support, dump archiving and maintenance, and a full-featured debugging interface.

Batch, CICS, DB2 and IMS
StarTool DA supports analysis and debugging of both batch and CICS ABENDs. Optional debugging support for IBM DB2 and IMS database ABENDs is also available.

Detailed Analysis and Reporting
StarTool DA displays failing instructions down to the source code level. It manages the archival and retrieval ABENDs and monitors trends over time to deliver high availability for infrastructure and applications.

ChangeMan ZMF + StarTool DA

Software Change and Configuration Management (SCCM) and z/OS dump formatting tools have traditionally been separate entities in program development and maintenance. SCCM products like Serena ChangeMan ZMF, control program source code and other development assets. Dump formatting tools, like StarTool SA, processed the information from z/OS abnormal end (ABEND) conditions, and simplified or interpreted the contents of z/OS SYSUDUMPs (or other dump types) and other error information. Both tools improve application programmer productivity: ChangeMan ZMF during the program development cycle and StarTool DA during the debugging cycle. Additionally, StarTool DA is invoked during ABENDs, often long after the traditional program development cycle is complete.

Serena has moved these two products to a higher level by integrating them. This provides a seamless, automatic access to the change management information of most interest to programmers when they are creating software, both during development and in production. One immediate benefit is faster diagnosis of production problems—seeing what changed most recently is automatic. Integration also improves accuracy, as the programmer is less likely to look at the wrong version of a program listing under the stress of debugging a serious production outage.

DA-ZMF integration enables sharing of the source and listing repository, which saves storage space and ensures usage of the correct source code. Other dump formatting products may require a separate listing repository with specially processed version of the program listing in order to provide source code analysis.

There is no chance to get the wrong version of a listing, or to attribute the problem to the wrong change with the DA-ZMF integration.

Debugging always starts with the right information—and since problems invariably occur because of the latest changes to a program—debugging starts with the most likely cause of the problem. Using the XML capabilities of ChangeMan ZMF, DA retrieves the listing and change history information for the affected program, then formats the dump down to the source code level providing:

- Program source statement at the failure
- Data fields and content at the failure
- ZMF Package and program information
- Change history

StarTool DA Product Datasheet

Serena® StarTool® IOO

Input-Output Optimizer

You know that the best manual tuning cannot produce the performance savings of an automated and dynamic tuning utility. Until now, no single integrated and comprehensive tool existed to tune all of your I/O processing.

Serena Star Tool IOO is an integrated optimization system that automatically and dynamically tunes the major components of z/OS's I/O processing functions to achieve dramatic throughput improvements both in batch and online. It is based on a proprietary analysis and intelligence gathering process during OPEN processing and I/O request to any VSAM or SAM (QSAM, BSAM and EXCP) dataset. IOO delivers the performance needed for applications that operate in today's fast-changing business environment.

VSAM Optimization

Eighty percent (80%) of all VSAM mainframe applications use Non-Shared Resources (NSR) because most only support NSR. IOO allows programs written in any language to use Local Shared Resources (LSR) when the program is dynamically or randomly accessing a VSAM file. IOO builds the optimal number of LSR buffers to maximize performance. LSR and NSR buffers are built above the 16M line to alleviate virtual storage constraints in the private area. For LSR, expanded storage using hyperspace buffering is automatically and transparently used when the file size is larger than the LSR pool.

Dynamic Memory Management

For those programs that are coded in Assembler and already use native LSR, the perfect quantity of above the 16M line buffers and hyperspaces are transparently allocated based on file size. For all LSR processing (NSR-LS R or LSR native), a separate LSR pool is used for each file to eliminate buffer stealing. When applications change processing modes from Direct to Sequential, IOO detects this and re-optimizes tuning parameters.

Sequential Optimization

IOO dynamically optimizes SAM to provide the best throughput and response time for I/O requests.

Blocking

Most sequential processing is highly dependent upon block sizes. IOO replaces or enhances SDB's reblocking function and performs reblocking on all sequential data sets, including those with hard-coded DCB's. The product reblocks all device types and reblocks without any JCL changes. IOO is smart enough to reblock only 35% of all your eligible data sets but achieve 90% of the potential savings.

Buffer Management

IOO dynamically optimizes buffers and Start I/O's for QSAM , BSAM and sequential EXCP I/O's. Specifically, it optimizes physical sequential data sets, permanent and temporary, including data sets generated by most utilities. It improves QSAM, BSAM and EXCP performance as much as 75%, saves up to 50% elapsed time for DB2 unloads and reloads and saves up to 40% elapsed time for IDMS.

Summary

Customers can override the rules, exclude datasets or jobs from IOO 's optimization process. Most users, take the default parameters and achieve outstanding results out of the box. IOO automates your tuning efforts and boosts user productivity. Before you decide you need a new CPU, additional DASD or start a manual system tuning project, find out your true capacity and performance potential with Serena StarTool IOO.

StarTool IOO Product Datasheet

Serena® Comparex®

The intelligent file comparison tool

As companies drive competitive differentiation through their enterprises, change to the mainframe systems that support those enterprise is a daily fact of life. Mainframe development teams are constantly challenged to attain higher levels of quality & productivity, reduce errors during program development and shorten application rollout times. To be successful, mainframe development teams must understand changes to their systems.

Comparex, Serena Software's intelligent file comparison tool, can help. Comparex, with hundreds of licenses in production at large mainframe sites around the world, helps teams discover errors, increase productivity, reduce overall development costs and shorten application deployment times. The combination of proven reliability, ease of use and fast processing speed make Comparex the comparison tool of choice for development teams.

Powerful file comparison

Comparex, available on z/OS, z/VM and z/VSE platforms, is the most powerful tool on the market for comparing data, text and directory files. The product performs fast, accurate, single-step comparisons of the contents of virtually any two-file types - libraries, directories, sequential files or databases. Additionally, file records of any size are supported.

Customers have reported that Comparex typically reduces comparison job times between 50 - 80% compared to other solutions!

Easy to use

Comparex is flexible, offering a wide variety of control parameters to meet the needs of all users, from the beginner to the expert. Accessible via the online ISPF interface or through batch execution, even novice users find it easy to take full advantage of Comparex's advanced comparison functions with minimal training.

Flexible

Productivity is enhanced across multiple teams with Comparex. Production teams leverage Comparex to verify that updates to a production systems based on a scheduled change control request were actually made. Auditors might use Comparex to document the differences between two production file versions. Online reports can be compared before printing to trap changes and then can be configured to print only the differences — saving printing time, consumables and paper costs.

Text, data, and directory comparisons

Comparex is the only software comparison tool that includes text, data and directory functionality in a single product.

The SYSUT3 COPYFILE output can also include separate files of changed records from either of the two input files. Comparex users have complete flexibility with the COPYSPLIT options, and can have all, none, or any other combination of five output files for any

Data comparisons

In data comparison mode, Comparex offers a wide range of functionality. Complete data records or individual fields can be compared with differences highlighted down to the byte or half-byte level. Date fields in various formats can be compared — with automatic conversion between formats.

There are no restrictions on record length or file size. Records can be in fixed, variable or spanned formats — Comparex handles all formats with ease, even supporting comparison between different formats.

In addition, a special mode even allows CSECT by CSECT comparison of load modules.

Text comparisons

Comparex finds and isolates differences in text files. The fade facility displays a few lines above and below the difference to help with file position orientation.

Comparex can produce a delta report with one pass through the files, highlighting the differences for easy identification. Comparex support includes ADABAS, DB2, IMS, CA-DATACOM, FOCUS, CA-IDMS, Oracle, Wylbur, CA-Librarian, CA-Panvalet, JCL, COBOL, PL/1, FORTRAN, CLIST, REXX, HTML,

Comparex Product Datasheet

XML, reports and documentation files.

Copybook parsing

If you are doing a DATA comparison and the files to be compared have a COBOL, Assembler, PL/I or DCLGEN copybook layout associated with them, there is a Parser provided that can produce an accurate keyword layout which can be imported to your comparison routine.

Directory comparisons

Occasionally, you may not need to compare contents — you only need to compare directories for member names. Comparex supports directory member comparisons on PDS or PDS/E libraries.

Date field comparisons

Comparex compares dates fields in files whether they have two digit years, four digit years or a mixture. Dates can be alpha-numeric and be compared successfully.

Keywords for flexibility for flexibility

Comparex provides over 50 keywords (such as: MASK, Record FILTER, Start/Stop options, Squeeze, String Searches) to tailor the comparison criteria to meet your exact requirements for both input files and output reporting. Records can be FILTERed, fields masked and multiple selection criteria on a single record are supported.

Test data generation

Actual production data can be FILTERED to extract a subset of records for testing purposes. Testing quality is improved by using production data.

Delta decks

The differences detected during a comparison can be written to a separate file called a delta (change) deck. Comparex allows users to create delta decks in CA-PANVALET, CA-LIBRARIAN, IEBUPDTE or Serena's ChangeMan ZMF format. Delta decks can speed the reconciliation of vendor packages and allow for improved version control of internally written programs.

Flexible reporting

There are over 50 options in Comparex for report formatting, including alphanumeric and hexadecimal display.

Comparex Product Datasheet

24 x 7 follow the sun support

Maximize your investment with Serena's award-winning support

Managing change can be a formidable challenge. Serena supports you from start to finish with robust products and solutions, backed by comprehensive customer support. You get award-winning expertise around the clock so you can maintain business continuity while maximizing productivity.

Our comprehensive support program includes access to experts online or via telephone, along with Serena's global support web site which provides 24x7 access to a community of experts, product updates, and resources focused on helping you leverage the full power of your investment. On this site you have anytime-access to complete case management, a comprehensive technical Knowledge Base, product news, product updates and documentation, and licensing information. You can also participate in customer forums and interact with customers to share knowledge and experiences with your Serena product of choice.

Your registered contacts with a valid serial number have unlimited access to all information on the Serena support site. Additional read-only contacts can access the Knowledge Base and view submitted cases.

Support Site Highlights

Not only can you submit, update and view cases, you can now have access to Support representatives via our Live Chat feature.

Engage in an online, **Live Chat** session with a technical specialist, who will quickly begin working with you to answer your question or troubleshoot your incident.

Our **Knowledge Base** contains thousands of technical tips and product usage information that can be searched using keywords, phrases or other criteria. The Knowledge Base also allows you to view solutions for known issues with links to the product patches where applicable.

Collaborate, exchange ideas, or receive notification of discussions with other Serena customers. **Customer forums**, threaded discussion groups, and live online sessions are a great way to expand your business solutions and find new, creative ways to extend your product usage.

The latest Serena product updates, patches and fixes are covered under your support agreement. Documentation and current releases for your licensed products are available for download.

Don't miss important information! There are two ways to receive product news. Review the Product News on the web site for current updates and important information, or register online for **Techmail**, a subscription that provides automatic email notification of product updates and releases. We encourage anyone who needs this information within your organization to subscribe to Techmail on the support web site.

We strive to resolve all of your issues promptly via electronic submission, Live Chat, or telephone. View our detailed **Service Level Objectives** on the Serena support web site.

With a Serena Support agreement, your designated Account Representative will contact you throughout the year to assist with any questions you may have and to assist you in fully maximizing your investment. Their primary interest is your success.

Languages spoken include **English, French, German, Spanish, Italian, Japanese, Korean, and Chinese**.

Customer Support

Australia
Tel: 1 800 335 664
Intl: +61 3 8656 9650

Austria
Tel: 0800 292 783
Fax: +44 1727 869804

Belgium
Tel: 0800 724 61 (English)
Tel: 0800 774 79 (French)
Fax: +44 1727 869804

Brazil
Tel: 0800-8917639

Canada
Tel: 800-443-1601

China
North Tel: 10800-610-0408
South Tel: 10800-261-0398

Denmark
Tel: 80-885921
Fax: +44 1727 869804

Finland
Tel: 0-800-1-12971
Fax: +44 1727 869804

France
Tel: 0800 91 56 07
Fax: +44 1727 869804

Germany
Tel: 0800 1822 496
Fax: +44 1727 869804

Hong Kong
Tel: 800 900521 India
Tel: 000 80 01008028 (Toll Free)

Italy
Tel: 800 79 11 79
Fax: +44 1727 869804

Japan
Tel: 0120-74-9090
Fax: 03-5447-3240

Korea
Tel: 007 981 4800 7645

Netherlands
Tel: 0800 022 1609
Fax: +44 1727 869804

New Zealand
Tel: 0800 444 515

Norway
Tel: 800-18850
Fax: +44 1727 869804

Singapore
Tel: 800 120 5092

South Africa
Tel: 0800 98 1696
Fax: +44 1727 869804

Spain
Tel: 900 968 929
Fax: +44 1727 869804

Sweden
Tel: 0200890163
Fax: +44 1727 869804

Switzerland
Tel: 0800 836737 (German)
Tel: 0800 836736 (French)
Fax: +44 1727 869804

UK
Tel: 0808 100 2672
Tel: 01727 813 943
Fax: +44 1727 869804

USA
Tel: 800-443-1601
Tel: 503-536-7891

Other Asia Pacific countries:
Tel: +61 3 8656 9650
Fax: +61 3 9843 3484

All other countries:
Tel: +44 1727 813 943 (English)
Tel: +44 1727 813 932 (French)
Tel: +44 1727 813 937 (German)
Fax: +44 1727 869804

Recent awards and recognition

Gartner ranking: **LEADER**

Forrester Wave for Application Lifecycle Management

Forrester ranking: VISIONARY
Gartner Magic Quadrant for Application Development

EMA: Most Complete Solution
EMA Radar Report for Next Generation IT Management Solutions

Info-Tech ranking: CHAMPION
Info-Tech Agile ALM Vendor Landscape

The 2012 SD Times 100
A Software Development Superfecta

Pink Elephant: 2011 Innovation Of The Year
Pink Elephant ITIL Excellence Awards

2012 Best IT Products and Services for Service Delivery
Networks Products Guide

2012 Bronze Winner for Best Enterprise Software
Tech Awards Circle

Top 10 Influencer
The ITSM Review

Product of the Week
Network World

Ovum spotlight
Analyst Report: Why Serena Service Manager Should Make Your ITSM Short-List

ITIL Verified
Pink Elephant

From Version Control to Complete Control with Serena ChangeMan ZMF

> *"We use ChangeMan ZMF to respond faster to change requests, increase staff productivity, provide traceability for audits, and help meet regulatory compliance standards."*
>
> *- Software Engineer, Global 500 Airline*

Save Money, Save Time, and Improve Productivity of your app dev and release teams when you upgrade from your old, expensive-to-maintain version control systems such as CA Panvalet® or CA Librarian®.

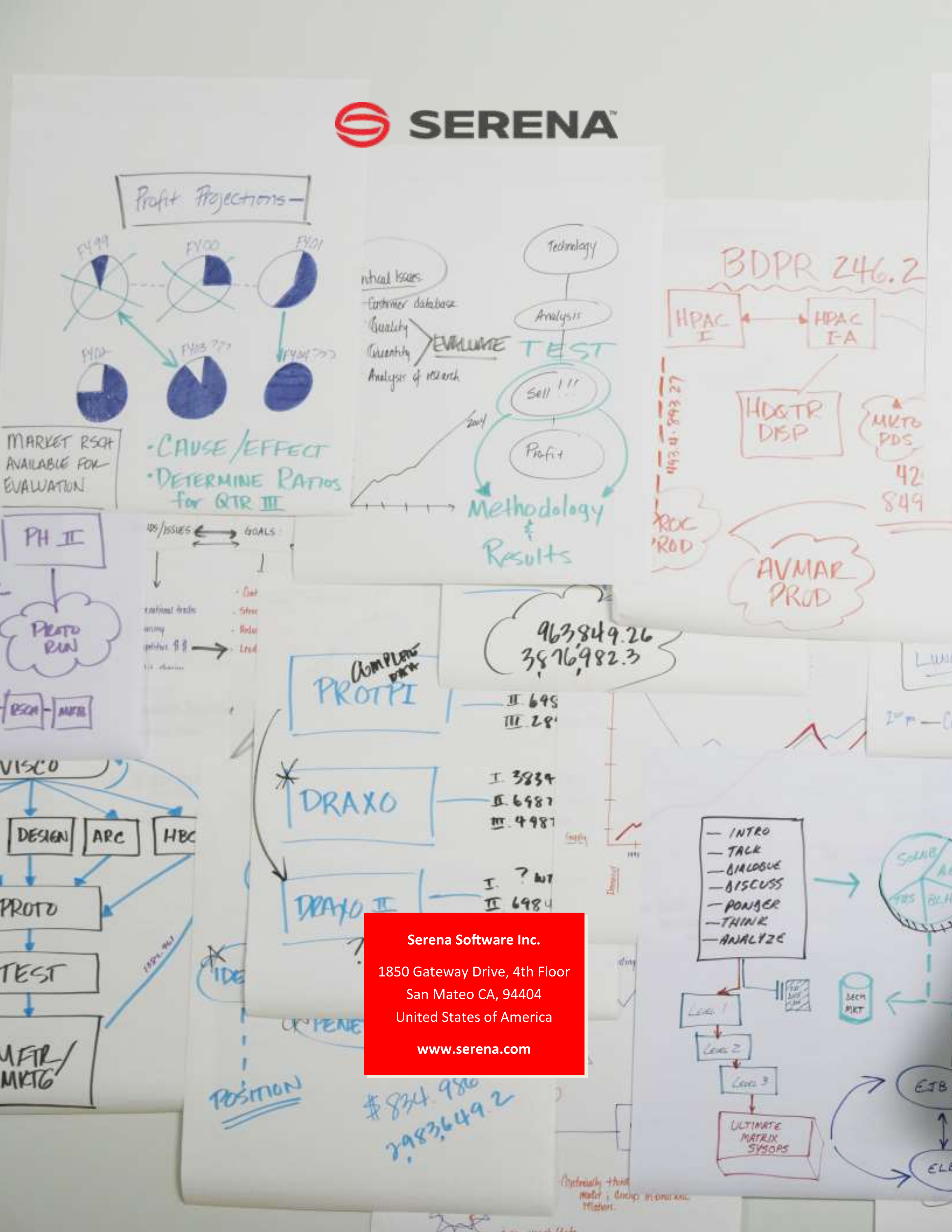
Migrate Easily to ChangeMan ZMF and modern application development solutions from Serena that support the entire application lifecycle for both mainframe and distributed applications.

ChangeMan ZMF is a complete, integrated software change and configuration management solution for the z/OS environment. It goes way beyond version control; ChangeMan ZMF automates software configuration management (SCM), enables build management capabilities, provides release management functionality across the enterprise, and generates audit trails and comprehensive management reporting.

Benefits of ChangeMan ZMF

- Reduced Costs – more efficient storage methodology
- Improved Productivity – complete automated build for all managed source code
- Improved Security and Audit – send code to other locations and automatically track when it was sent, installed, and backed out
- Support for Java artifacts and file systems such as zFS and HFS
- And much more!

SERENA



Serena Software Inc.

1850 Gateway Drive, 4th Floor
San Mateo CA, 94404
United States of America

www.serena.com