SERENA

White Paper

Move Fast
Without Breaking Things

High Velocity Software Development in Highly Regulated Enterprises

Greg Hughes, CEO Serena Software

Contents

About the Author

Greg Hughes serves as the CEO of Serena Software and is on the Board of Directors. For 35 years, Serena has focused on the mission of helping large, highly regulated enterprises improve the software development lifecycle – delivering software faster and with lower risk.

Greg started his career as an engineer and software developer writing applications in multiple languages and even practicing the lost art of assembly programming.

Prior to joining Serena, Greg worked at Silver Lake Partners, Symantec, VERITAS Software, and McKinsey & Company. He holds degrees from the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology and from the Graduate School of Business at Stanford.

Today, there is tremendous pressure on enterprises to accelerate the velocity of software innovation because of competition and the opportunity to exploit disruptive technologies such as mobile. This pressure is driving application development organizations to apply a range of modern practices, developed in smaller Web companies, to the software development lifecycle (SDLC).

In the race to deliver software faster, small companies can afford to adopt the early Facebook credo "Move Fast and Break Things." However, in the quest to move faster and adopt modern development practices, larger enterprises must avoid increasing security, compliance, and performance risks in the software development lifecycle. Managing SDLC risk is particularly critical in sensitive, highly regulated sectors such as financial services, government, healthcare, automotive, and defence.

In many highly regulated companies, proprietary software underpins differentiated business processes, unique products, and valuable services. Risks in the software development lifecycle can jeopardize valuable enterprise assets, including:

- Intellectual property
- Customer reputation and goodwill
- Legal and regulatory standing
- Financial capital

How should executive management in highly regulated sectors act to ensure their application development teams adopt the modern development practices while also managing security, compliance and performance risks? In other words, how can enterprises "Move Fast *Without* Breaking Things"?

The pressure to move fast

We are living in an era of breathtaking and relentless change, driven by software innovation.

Software development is more strategic and critical than ever before, leading to tremendous pressure to compress development cycle times. In response, application development teams across all sectors are adopting modern practices including:

- Agile development, which emphasizes delivering working software frequently to satisfy customers and welcomes changing requirements even late in development (Kent Beck 2001)
- Parallel development, which enables multiple developers to work on features in parallel and merge changes back into a single "main-line" (Appleton 2003)
- Continuous integration, which encourages frequent automated builds and integration testing of parallel branches to identify defects early and reduce rework (Fowler 2006)
- Continuous delivery, which focuses on using automation, tight version control practices, and collaboration to speed completed software changes through a deployment pipeline into production (Farley 2011)

- Lean, which adapts principles of the Toyota Production System to software development, including Kanban boards, eliminating waste (i.e., activity that doesn't contribute to customer value), and small batch sizes (Jez Humble 2014) (Vodde 2009)
- DevOps, which aims to improve the collaboration and shared goals between development and operations to address the conflicting goals between these two teams (Gene Kim 2015)

Innovative software development is the key to unlocking the potential of disruptive technologies. According to research done by McKinsey & Co., disruptive technologies have "the potential to drive economic impact on the order of $14 trillion to $33 trillion per year in 2025" (James Manyika 2013). These technologies include:

- Mobile: The iPhone 4 at $400 is equal in performance to the fastest supercomputer of 1975, which cost $5 million.
- Artificial Intelligence: There's been a 100X increase in performance in computing power from IBM's Deep Blue (which was the chess champion of 1997) to Watson (the Jeopardy winner in 2011).
- Internet of Things: Over the past 5 years, there's been a 300% increase in connected machine-to-machine devices.
- Cloud technology: The price-performance of a public cloud server is doubling every 18 months.

Adopting modern development practices needs to happen – it has become a matter of survival.

Why moving fast can break things: practices vs. policies and systems

Software developers should be free to adopt modern practices in order to do their work better.

However, while these practices are beneficial, the systems and polices built around them are often weak and introduce vulnerabilities into the software development lifecycle that can be particularly troublesome for large enterprises in highly regulated sectors.

These vulnerabilities include:

- Allowing the proliferation of numerous open source code repositories (e.g., GIT, SVN)
- Allowing developers to configure and administer their source code repositories without adequate safeguards and an overarching policy
- Poor access controls, password protection, and authentication mechanisms for a globally distributed development team
- Weak process controls and enforcement, such as:
  - Change management
  - Automated testing and QA
  - Secure development lifecycle/static security scanning
  - Control over the use of open source code module

  - Onboarding management for employees and contractors
- Highly manual processes
- Lack of detailed event logging and traceability, leading to weak audit support
- Inadequate segregation of duties and work authorization

While small companies can adopt the Facebook credo “Move Fast and Break Things,” larger enterprises, especially in highly regulated sectors, need to “Move Fast Without Breaking Things.”

As software developers adopt new practices to speed features to market, executives and shareholders need to have confidence that the risks in the SDLC are managed well.

To balance these competing principles, executives should apply the philosophy of Enterprise Risk Management (ERM) to the SDLC. Applying enterprise risk management to the SDLC requires the Application Development, Change Management, IT Security/Risk, and Audit teams to collaborate and work closely together with experts from industry.

Applying the philosophy of enterprise risk management to the SDLC starts with identifying the security, compliance, and performance risks in software development.

Security Risks

On January 12, 2010, Google disclosed on a public blog post that it was the victim of an extremely sophisticated cyberattack targeting its most valuable intellectual property – its source code (Drummond 2010). Later named “Operation Aurora,” the attack originated from China and was aimed at stealing or possibly modifying the source code of dozens of U.S. companies across many sensitive industries including software, defence, and financial services. The attack was one of the first examples of an “advanced persistent threat” – a stealthy network of malware that goes about silently infiltrating a network, finding valuable intellectual property, and then, when the conditions are just right, stealing it before the alarms go off.

According to Dmitri Alperovitch, a VP from McAfee Labs, “[the] SCMs (Source Code Management systems) were wide open. No one ever thought about securing them, yet these were the crown jewels of most of these companies in many ways – much more valuable than any financial or personally identifiable data that they may have and spend so much time and effort protecting” (Zetter 2010).

Cyber threats are a growing problem in highly regulated industries as evidenced by JP Morgan Chase’s CEO stating that the bank will double its spending on cyber security from $250 million to $500 million per year after a recent well publicized breach (Tracy 2014).

These cyber threats emerge from two different categories of highly motivated sources – those inside the enterprise and those outside:

> The insider – After Edward Snowden, organizations are realizing the enormous damage that can be inflicted by people who are given permission to access information, e.g., employees, contractors, and offshore teams. The insider threat is not always from malicious actors, more frequently it is the "ignorant insider" that causes damage. The author recently heard of one customer in the financial services sector who had a developer who emailed source code to his home!
>
> The outsider – Outsider threats have the greatest funding, sophistication, and organization and are generally categorized as one of four types: 1. State-sponsored, 2. Organized Crime, 3. Activists, and 4. Terrorists.

There exist a wide variety of security threats to the SDLC, including:

- Stealing intellectual property – Source code may be the most critical intellectual property that an enterprise creates. In "Operation Aurora," Google, Adobe, Juniper Networks, and Rackspace all confirmed that their systems were targeted. Media reports also suggested that dozens of other important companies were attacked.
- Unauthorized changes to the source (the "backdoor") – Attackers who gain access to source code can install "backdoors," giving them "command and control" over critical functions.
- Using source code to identify vulnerabilities – Studying the stolen source code can give attackers sufficient knowledge to find new vulnerabilities in targeted systems.
- Corruption along the path to production – In the majority of enterprises, getting completed code from development into operations is a highly manual, uncontrolled process, jammed into nights and weekends. Multitier applications require coordinated changes across many people, platforms, and environments, e.g., mainframe, web-tier, cloud, database, etc. If these changes are not made correctly, the systems can be exposed to vulnerabilities at every manual step.

Compliance Risks

Since the passage of the Sarbanes-Oxley Act (SOX) in 2002, managing regulatory compliance is a growing burden on the SDLC. The increasing diversity of regulations on a variety of levels (state, federal, and international) makes it difficult to keep up with the required changes to the SDLC, particularly for highly regulated industries such as:

- Financial services – Gramm-Leach-Bliley Privacy Act (GLBA), Dodd-Frank, Basel III
- Insurance – Model Audit Rule
- Payments – Payment Card Industry Data Security Standard (PCI DSS)

- Healthcare – Health Insurance Portability and Accountability Act (HIPAA), Affordable Care Act
- Automotive – ISO26262
- Government – Federal Information Processing Standards/Federal Information Security Management Act (FIPS/FISMA)
- Aerospace/Defence – Export Control Act

Adherence to new regulations is costly. For example, the SEC's Reg SCI (Systems Compliance and Integrity) rule, which applies to 44 entities, is expected to initially cost a collective $242 million with another $191 million in annual costs.

Risk of noncompliance is a serious problem for executives in highly regulated industries. The distraction and costs of litigation are burdensome for management teams, and the penalties imposed by government institutions can be huge.

The software development lifecycle for critical systems is a prime target of these regulations (Epps and Norris 2012) (MSDN 2006). Compliance requires adopting secure development practices and rigorously managing processes such as authorization, segregation of duties, change control, and audit support (e.g., logging and reporting).

In addition to regulatory compliance, legal compliance is another issue for the SDLC. Software developers are increasingly using source code that is freely available online in order to meet pressing deadlines. How is all this source code tracked and monitored? How do enterprises make sure that the licensing restrictions are followed? And, finally, how do enterprises make sure that this software is patched and upgraded when required?

Performance Risks

The definition of performance used here is broad – *performance* means the software does what is intended. Performance risks are incurred in the SDLC when adequate testing is not completed or can be circumvented before code goes to production. The release process, promoting code from development into production, is one of the most hazardous steps of the SDLC.

One of the most catastrophic cases of performance risk is the story of Knight Capital (Heusser 2012). Knight Capital was a high-frequency trading firm that was one of the largest traders in U.S. equities. In June 2012, the New York Stock Exchange was allowed to launch its Retail Liquidity Program (RLP) and informed trading firms such as Knight Capital that RLP would go live on August 1st.

In the rush to get its trading software ready by the go-live date, Knight Capital made a terrible mistake that caused them to lose roughly $440 million dollars in the first 30 minutes of trading. This huge loss crippled the firm so badly that it had to be acquired by a competitor (Getco LLC).

According to the New York Times article "Trying to Be Nimble, Knight Capital Stumbles," competitors questioned Knight Capital's aggressive approach (Silver-Greenberg 2012). "The time between the approval of the software and the time it was implemented was incredibly quick," said a head of equity trading at another firm.

A detailed analysis of the error stated that Knight Capital mistakenly deployed into production a test software module that executed its own trades rather than responded to trades from external parties (Nanex Research 2012). This test software module should never have been deployed into production – it was accidentally included in the release package and started on the NYSE's live system.

Moving Fast Without Breaking Things: Best Practices for the Enterprise

As they adopt new practices to speed application development, enterprises in highly regulated sectors should take a disciplined and programmatic approach to ensuring the security, compliance, and performance of the SDLC. The following list gives five best practices to consider during this effort.

1. Adopt developer-friendly tools. This is rule #1 for a reason: If the developers don't like the tools they're given, they will circumvent them. Nothing else will matter. Risk management technology should act like invisible guardrails to developers. Application development will "opt out" of the systems provided by IT when they don't enable the advanced practices desired by the team or are too burdensome to use. Tools should embrace modern development practices and support modern UIs.

2. Upgrade the level of process control over the end-to-end SDLC. In many enterprises, the SDLC is comprised of many separate products without a strong foundation of cross-product workflow management. Robust process management tools make it easy to coordinate activities and information across tools and provide the required event tracking and access control needed for risk management. The key cross-product processes areas to focus on include:

   - Enforcing the use of secure development lifecycle practices, e.g., security and open source scanning
   - Robust change management, e.g., change request to source code to load integrity
   - Release management, e.g., ensuring that the proper steps are followed such as change advisory board (CAB) approval
   - Contractor/employee onboarding and offboarding

3. Automate wherever possible. Substituting manual steps with automation provides repeatability, lowers costs, and reduces the likelihood of errors (either intentional or not). In some instances, automation provides a "free lunch" – increasing velocity of operations and lowering costs while also ensuring control and traceability. For example, in most enterprises application deployment is a labor-intensive process involving many people during nights and weekends. Application deployment automation products can eliminate hundreds of manual steps, replacing them with a standardized, auditable, automated process.

4. Reign in "repository proliferation" and create a centralized "hardened source code management system." This is the corollary to rule #1. Too many enterprises let the number of source code repositories get out of hand because the developers can easily standup their own open source-based systems, e.g., GIT and Subversion. Software change and configuration management (SCCM) should be centralized and "hardened," e.g., configured securely, administered correctly, and installed on a separate and secure server (McAfee Labs 2010). Vendor selection is critical – not all SCCM products support a sufficient level of security and control, e.g., fine-grained and integrated access controls; detailed auditing and logging (to identify anomalous behavior, provide auditability, and underpin forensics post-incident); integrated peer review; baselines. The highest performing companies focus on strong version control practices (Puppet Labs 2014). For example, Google has a single code repository that supports 15,000 engineers and 5,500 code commits and automated tests daily. If the executives at Google can reign in repository proliferation, everyone else can too!

5. Continue using the mainframe for the core transactional systems. Many enterprises continue to grow their mainframe capacity, for, among other reasons, the unparalleled security provided by these systems. Mainframe vulnerabilities (according to NIST and US-CERT) are in the low single digits as compared with the thousands for Windows, Linux, and UNIX.

Why Serena?

For 35 years, Serena has focused on the mission of helping large, highly regulated, enterprises improve the software development lifecycle – delivering software faster and with lower risk.

Serena understands the unique issues of large highly regulated enterprises – serving over half of the Fortune 100 in industries that include insurance, commercial banking, payments, asset management, government, hospital and medical plans, national security/defence, and telecommunications. For example, in the U.S., Serena's customers include:

- 8 of the top 10 banks
- 5 of the top 5 aerospace and defence firms
- 6 of the top 10 mutual life insurance firms
- 3 of the top 3 health insurance and managed care companies

To meet the challenging demands of these leading companies, Serena provides award winning, "white glove" customer service with an exceptionally experienced staff of customer support employees with an average tenure over 11 years. This customer focus leads to strong customer loyalty – the top 100 customers have been with Serena for an average of 18 years.

Serena's four key products accelerate the adoption of best practices for the SDLC:

- Serena Business Manager – a workflow platform to upgrade the level of process control across the end-to-end SDLC
- Serena Deployment Automation – for automating the application deployment step
- Dimensions CM – a developer friendly, scalable, traceable, and secure Software Change Management (SCM) system to reign in repository proliferation
- ChangeMan ZMF – the leading mainframe SCM for critical transactional applications

Serena is dedicated to best practice rule #1, making sure that all products are developer-friendly. Serena software developers use the company's own products for the SDLC and employ customer advisory boards, virtual user groups, and events to gain feedback from customers.

Serena's solutions embrace the complexity of the enterprise SDLC, integrating with third-party solutions (open source, proprietary, mainframe) and easily adaptable to custom processes.

Serena Business Manager

In a large enterprise, the software development lifecycle is made up of workflows across specialized teams that are reporting into different functional areas, each using different tools for their work. In the process of releasing software, for example, the development, Q/A, and operations functions are involved and the tools used include source code management, defect databases, and test management. Other workflows include change management, issue and defect tracking, contractor management, and employee onboarding.

Many organizations have these workflows in an "unmanaged" state and leave employees to use an ad hoc collection of technologies (e.g., email, spreadsheets) to support the coordination required to get their job done. However, while appearing to save a little money, there are a number of hidden costs and problems with this "unmanaged" approach.

Larger enterprises should move from "unmanaged" to "managed" workflows in their SDLC, using a process application to get the following benefits:

- Drive a standard, consistent process
- Track steps to comply with policy and regulations and support audits
- Provide transparency and accountability – showing who is doing what and who isn't
- Zero in and fix process bottlenecks to support continuous process improvement
- Automate and orchestrate steps to reduce time spent on non-value-added activities

Process-based applications in the SDLC have traditionally been developed through one of two approaches: (1) If they exist, buying a packaged application or SaaS service; or (2) Building homegrown systems. Homegrown systems are built in a myriad of ways – custom developed applications in modern programming languages, on top of scripting tools in individual products, or even using Lotus Notes and SharePoint.

Packaged applications are often inflexible – difficult to customize or adapt to future changes. Homegrown systems are often hard to maintain and can lack basic enterprise features such as security and access controls.

Serena offers a "third way" to develop process-based apps for managing workflows in the SDLC: using the Serena Business Manager (SBM) process automation platform. SBM is a business process management (BPM) platform for IT designed from the ground up to automate core processes across the application development and release management, IT operations, and broader IT servicing needs.

Serena's customers typically apply the SBM platform in three phases over time: 1. Achieve quick wins in the SDLC, 2. Apply SBM to manage broader IT processes, 3. Extend SBM into business processes that intersect with IT (See Table 1).

Table 1.

Phase	Examples
1. Achieve Quick Wins in the SDLC. Upgrading the level of process control over the SDLC is usually a "target rich" opportunity for SBM.	• Serena offers process templates for a range of SDLC workflows, including: – Issue and Defect Management – Change Request Management – Work and Project Management – Test Case Management – Open Source Management • Release management – Serena provides an application built on top of the SBM platform for release management. Serena Release Control helps mature the release management process, increasing transparency, traceability, and control.
2. Apply SBM to Manage Broader IT Processes. SBM is an excellent platform for managing IT processes.	• IT service management, e.g., Serena Service Manager, an application built on top of the SBM platform, which helps drive progress towards ITIL-based process management for incident, change, and problem management • Security management, e.g., security incident management, certificate management • Asset management and tracking • Server provisioning • Root cause investigation
3. Extend SBM into Business Processes that Intersect with IT. The most aggressive users of SBM extend it to manage a wide range of business processes.	• HR – Employee onboarding/off boarding, contractor management, new hire provisioning • Sales – Quotation process, discount approval, customer references • Finance – Expense reimbursement, travel request approval, purchase requisitions, capital expenditure requests • Legal – Copy approval, litigation hold • Industry-specific – Claims processing, clinical trials management

SBM provides the following benefits as a platform to manage process workflows:

- Rapidly create process-based applications**.** Start with one of the basic process templates and then use Composer, a unified process-based application designer to refine it. Composer is a completely visual designer that models all aspects of an application, including human workflows, system workflows, integrations, dynamic forms and user interaction, business rules, roles, privileges, reports, and dependencies.

- Easily customize and adapt. Business analysts can easily change workflows, forms, data, integrations, rules, privileges, and other taxonomy elements using SBM Composer, and deploy them without any loss of history, data, or process integrity. Optionally, changes can be applied to in-flight process items as well. All process changes are automatically version controlled at both design time and runtime and can be easily rolled back.

- Integrate with other systems. Use the built-in orchestration engine to manage sophisticated integrations with other systems through Web services and REST-based APIs. Gather information from disparate sources and present it to users in real time. Automate steps that previously required manual interaction.

- Leverage the rich reporting and auditing capabilities. Zero in on process bottlenecks with a built-in library of listing, distribution, trend, and other reports and dashboards. End users can create powerful reports to manage and measure the work automated in SBM using easy-to-use report creation and editing wizards. In addition, SBM features comprehensive change auditing, with automated change capture and reporting, supporting regulatory governance.

- Fully customizable extensive role-based privilege model in SBM Composer. Process designers can define role-based, application-level security down to the data/field level and specific workflow actions to fulfill security and regulatory compliance requirements. In addition, SBM's single sign-on (SSO) engine ensures that the identity of the participating user is propagated securely across any triggered system workflows for transaction integrity.

- Support the entire range of user devices (mobile, tablet, and PC). This includes fast track approvals and timely notifications wherever and whenever people work.

Serena Business Manager and its process applications are the foundation of a strategy to upgrade process control across the end-to-end SDLC, then apply that rigor to IT and the rest of the business.

Serena Deployment Automation

As already discussed, many enterprises are moving to a high velocity software development approach to releasing new and enhanced applications. High velocity software development drives revenues because "software doesn't make money until it is in the hands of users."

Currently, however, most companies use an application deployment process that is highly manual, involves detailed custom scripts for each tier of the application (e.g., database, Web, server) and can involve dozens or more people all on the phone for hours at a time.

This labor-intensive deployment process simply cannot cope with higher frequency software releases without significantly increasing the risk of failed performance or security vulnerabilities. In addition, highly regulated enterprises must support the detailed tracking and segregation of duties required for compliance reasons, which is challenging in a manual process.

Serena's solution is to automate application deployment by using Serena Deployment Automation.

Using Serena Deployment Automation provides the following benefits:

- Simplify, standardize, and automate the deployment of the most complex multitier applications across all environments
- Seamlessly deploy applications across heterogeneous, distributed physical, virtual, and cloud
- Reduce application failures in data center production by up to 90%
- Cut the time and cost of managing deployments by up to 80%

Serena Deployment Automation features include:

- Easy-to-use graphical editor for process and deployment automation
- Model-based deployments through application snapshots
- Artifact repository providing secure storage and traceability
- Full visibility, out-of the-box audit and compliance reports
- Plug-in architecture with out-of-the-box support for the major application environments
- Role-based security, approvals, and notification support
- Seamless integration to third-party tools: SCCM, build and release, QA, help desk, and ticketing

Serena Deployment Automation is the key component of a strategy to automate the SDLC wherever possible.

Dimensions CM

Serena's Dimensions CM has a proud heritage as one of the leading software change and configuration management systems, with a strong foothold in the highly regulated enterprises where absolute control over the software development process and artifacts is required.

However, in the past 5-10 years, as large enterprise development teams adopted the modern practices of smaller companies, independent pockets of Subversion and GIT, popular open source version control tools, spread rapidly.

The resulting "repository proliferation" has caused many problems for the highly regulated enterprise:

- Multiple repositories of source code and other artifacts are creating potential security, access, and data loss issues.
- Control over the policies governing source code access is inherently weak.

- Adherence to a documented development process, e.g., static security checks, is compromised.
- Configuration and change management are done manually through labeling and/or dealing with differences upon check-in/merge in open source tools.
- Open source tools track changes through log files, not full databases like a true SCCM tool, limiting auditability and tracking capabilities.

Several years ago, Serena recognized this situation and put a large engineering investment in creating a true SCCM tool that is easy to use and powerful for developers, while making sure that there were no compromises in the areas of process and artifact control that were absolutely critical for the highly regulated enterprise. The result, Serena's biggest release in seven years, is Dimensions CM 14.

Dimensions CM 14 has many features that developers love:

- Powerful visual branching and merging capability.
- Streamlined and integrated peer review process.
- High-speed access over WANs through a local library cache.
- Integration with the major integrated development environments (IDEs) (e.g., Eclipse, Visual Studio) and the modern continuous delivery tool chain – Jenkins, Hudson, Chef, Puppet, etc.
- Support for mobile development.

Dimensions CM 14 also has features that operations, QA, and audit appreciate:

- Fined-grained access control that can manage who can do what, to which objects, when and why.
- Full audit trail for development activities enabling compliance with standards such as ISO9000, Capability Maturity Model Integration (CMMI), etc.
- Integrated change request management with configurable degrees of control.
- Scalability to thousands of concurrent users, hundreds of terabytes of storage, and millions of file revisions. Dimensions supports load balancing for high availability and failover for disaster recovery (DR).

Dimensions CM serves as the secure and compliant enterprise hub for source code management outside the mainframe environment. Moving all source code (including that stored in open source repositories such as GIT and Subversion) into a single secure Dimensions CM repository will buttress the enterprises' security and compliance posture on the dimensions of confidentiality, integrity, and auditability:

- Confidentiality – Dimensions supports many authentication mechanisms with fine-grained, role-based access control. GIT and Subversion provide minimal support for granular role-based access control; authentication is limited and difficult; and the level of authorization is crude (e.g., whole repositories for GIT).

- Integrity – Dimensions has strong tamperproof controls available for data in flight and at rest, integrated stages gates, approvals and review processes. In GIT and Subversion, this requires significant hardening, customizations, and third-party tools. Dimensions uses a high-speed centralized architecture to underpin data integrity.

- Auditability – Dimensions provides a comprehensive audit trail of who did what, when, and why with full transaction logging to a relational database management system

A Serena Customer

One of the largest and fastest growing credit unions in the U.S. worked with Serena to support its increasingly popular mobile applications.

The development team uses modern practices for rapid innovation and customer responsiveness; but, the audit and compliance teams require traceability and security.

The credit union uses ChangeMan ZMF to support the applications on the back-end transactional system, Dimensions CM for all other environments, and SBM to support process control.

(RDBMS). GIT and Subversion provide only a basic audit trail available in the form of a commit log and capture no record of why changes were made without third-party tools.

Dimensions CM is the ideal solution for reigning in repository proliferation.

ChangeMan ZMF

The mainframe – in existence since the 1960s – continues to thrive, particularly for large customers using it for complex, transactional applications. According to survey research by BMC, 90 percent of large mainframe shops expect stable or growing MIPS use over the next two years (BMC 2014). The top two reasons cited for continuing investment in the mainframe were mainframe platform availability advantages and security strengths.

In the era of 24X7 Internet connectivity, the mainframe's availability features (e.g., hardware/software virtualization, redundancy, hot swapping) and security technologies (e.g., hardware cryptographic co-processors) make it the centralized data server of choice for highly regulated enterprises such as payments, banking, insurance, and telecommunications.

Mobile applications are driving a rising volume of transactions on the mainframe. For example, banking customers look at their balances more frequently when they're easily accessible. Many mainframe teams are finding that developing new and supporting existing mobile applications are a top priority.

Serena ChangeMan ZMF is the most comprehensive and fully integrated solution for software change, configuration, and release management on z/OS. For nearly 20 years, large, highly regulated enterprises have trusted ChangeMan ZMF to manage and automate the process of migrating software changes from development to any test environment and then to the production environment.

ChangeMan ZMF is designed for developers, testers, and release managers:

- Developers have a choice of user interfaces, including traditional ISPF or a Windows client. Users of RDZ's Eclipse-based IDE can access the functionality of ZMF without leaving their IDE. Extensive impact analysis and safe parallel development improves productivity.
- Testers appreciate the rich, built-in approval and notifications and the features to ensure completeness and stability of code delivery. Promotion through test areas is tightly controlled.
- Release managers use the built-in release calendar, automated release schedule, automated deployment to test and production, and full back-out capabilities to reduce risk. Auditability is guaranteed through enforcing source-to-load integrity.

One of the most significant strengths of Serena for the highly regulated enterprise is the cross-platform product line that supports the software development lifecycle from mainframe to mobile applications. With the mainframe increasingly relevant to new applications such as mobile, many enterprises are finding that coordinating their software development lifecycle in key areas such as release management is becoming increasingly critical. Serena is an ideal technology partner for these firms.

Conclusion and Summary

Large enterprises are adopting modern, high velocity software development practices including:

- Agile
- Continuous integration/continuous delivery
- DevOps
- Lean

Question: What modern development practices is your enterprise adopting to increase the velocity of software development?

However, at the same time they adopt these modern practices, highly regulated enterprises must manage three key risks in the software development lifecycle:

- Security
- Compliance
- Performance

Question: What are the biggest risks in your enterprise's software development lifecycle? How are these risks identified and addressed?

In order, to speed software delivery and manage risk, there are five best practices that executives should consider. Representatives from application development, security and audit, and IT change management should be part of developing the solution.

These five best practices are:

1. Adopt developer-friendly tools
2. Upgrade the level of process control over the end-to-end SDLC
3. Automate wherever possible
4. Reign in "repository proliferation" with a central "hardened source code management system"
5. Continue using the mainframe for the core transactional systems

Questions: Does your enterprise have an initiative around one or more of these best practices? Are they progressing with speed and urgency? Who is responsible for driving these initiatives?

Serena makes an ideal technology partner for executives in highly regulated enterprises who want to increase the speed and velocity of software innovation.

- Serena understands the unique challenges of the SDLC in highly regulated enterprises.

- Serena's four key products accelerate the adoption of best practices for the SDLC:
  - Serena Business Manager to upgrade process control across the SDLC
  - Serena Deployment Automation to automate application deployment
  - Dimensions CM to reign in repository proliferation
  - ChangeMan ZMF to support core mainframe transaction systems
- Serena integrates with third-party systems and spans "mainframe to mobile" across the SDLC.
- Serena provides best-in-class customer service focused on the SDLC.

Question: How can Serena help your enterprise move fast without breaking things?

References

Appleton, Stephen Berczuk and Brad. 2003. *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*. Addison-Wesley.

BMC. 2014. 2014 *Annual Mainframe Research Results*. Survey Findings, BMC Software, Inc.

Drummond, David. 2010. google.com. January 12. http://googleblog.blogspot.com/2010/01/new-approach-to-china.html.

Epps, Cindy Van, and Nick Norris. 2012. *Software Development Compliance - Overview*. September 28. https://jazz.net/library/article/856.

Farley, Jez Humble and David. 2011. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Pearson Education, Inc.

Fowler, Martin. 2006. Continuous Integration May. www.martinfowler.com.

Gene Kim, Patrick Debois, John Willis, Jez Humble, Damon Edwards, John Allspaw. 2015. *The DevOps Cookbook*. To be published in 2015 - Early draft reviewed by author.

Heusser, Matthew. 2012. *Software Testing Lessons Learned from Knight Capital Fiasco*. August 14. http://www.cio.com/article/2393212/agile-development/software-testing-lessons-learned-from-knight-capital-fiasco.html.

James Manyika, Michael Chui et. al. 2013. *Disruptive technologies: Advances that will transform life, business, and the global economy*. McKinsey&Co.

Jez Humble, Joanne Molesky & Barry O'Reilly. 2014. *Lean Enterprise: How High Performance Organizations Innovate at Scale*. O'Reilly Media.

Kent Beck, et. al. 2001. *Manifesto for Agile Software*. http://agilemanifesto.org/.

McAfee Labs. 2010. *Protecting Your Critical Assets: Lessons Learned from "Operation Aurora"*. McAfee.

MSDN. 2006. *Regulatory Compliance Demystified: An Introduction to Compliance for Developers*. March. http://msdn.microsoft.com/en-us/library/aa480484.aspx.

Nanex Research. 2012. *The Knightmare Explained*. August 3. http://www.nanex.net/aqck2/3525.html.

Puppet Labs. 2014. "2014 State of DevOps Report."

Silver-Greenberg, Jessical. 2012. *Trying to Be Nimble, Knight Capital Stumbles*. August 2. http://dealbook.nytimes.com/2012/08/02/trying-to-be-nimble-knight-capital-stumbles/?_r=2.

Tracy, Ryan. 2014. *wsj.com*. October 10. http://www.wsj.com/articles/j-p-morgans-dimon-to-speak-at-financial-conference-1412944976.

Vodde, Craig Larman and Bas. 2009. *Scaling Lean & Agile Development*. Pearson Education.

Zetter, Kim. 2010. *Report: Google Hackers Stole Source Code of Global Password System*. April 20. http://www.wired.com/2010/04/google-hackers/.

SERENA™

Website: www.serena.com

Phone: 1-800-457-3736

Email: info@serena.com

About Serena

Serena Software provides Orchestrated application development and release management solutions to the Global 2000. Our 2,500 active enterprise customers, including a majority of the Fortune 100, have made Serena the largest independent Application Lifecycle Management (ALM) vendor and the only one that orchestrates DevOps, the processes that bring together application development and operations. Headquartered in Silicon Valley, Serena is a portfolio company of HGGC, a leading middle market private equity firm.