WHITE PAPER

Thriving in an Era of Disruption and Disconnected Development

Best practices in adopting open ALM platforms for hybrid and multi-source development

Executive Summary

Rapid growth in mid to large sized companies has resulted in development organizations made up of a proliferation of disconnected teams, tools, processes and core software assets. Core applications required to run the business (e.g. high-value financial transactions or decision-support systems based on big data and analytics) are frequently managed in Waterfall and manual methods. Conversely, customer-facing, high-volume interactions are frequently developed and delivered in the most agile way possible, with processes and tool chains optimized around dynamically changing market use cases.

Developer freedom of choice has had its price, as workgroups have been free to acquire their own tools (open source, proprietary or legacy favorites) and processes, exacerbating the problem. Too often the tool or processes optimized for one workgroup may not work for all or even cause conflict at the system integration level.

Managing conflict is critical as increasingly, distinct teams, applications, and modes of work are compelled to work together. Disruptive opportunities such as IoT and the move to mobile-first and Cloud-first models present inherent challenges in reconciling the development approaches, especially as concern scaling and governance. Coupled with increased marketplace pressures to “release early and often,” many organizations are confronting the very way they approach software development. Some enterprises have embraced the promise of Application Lifecycle Management platforms to address the increased requirements and pressures, but many first-generation attempts have fallen short.

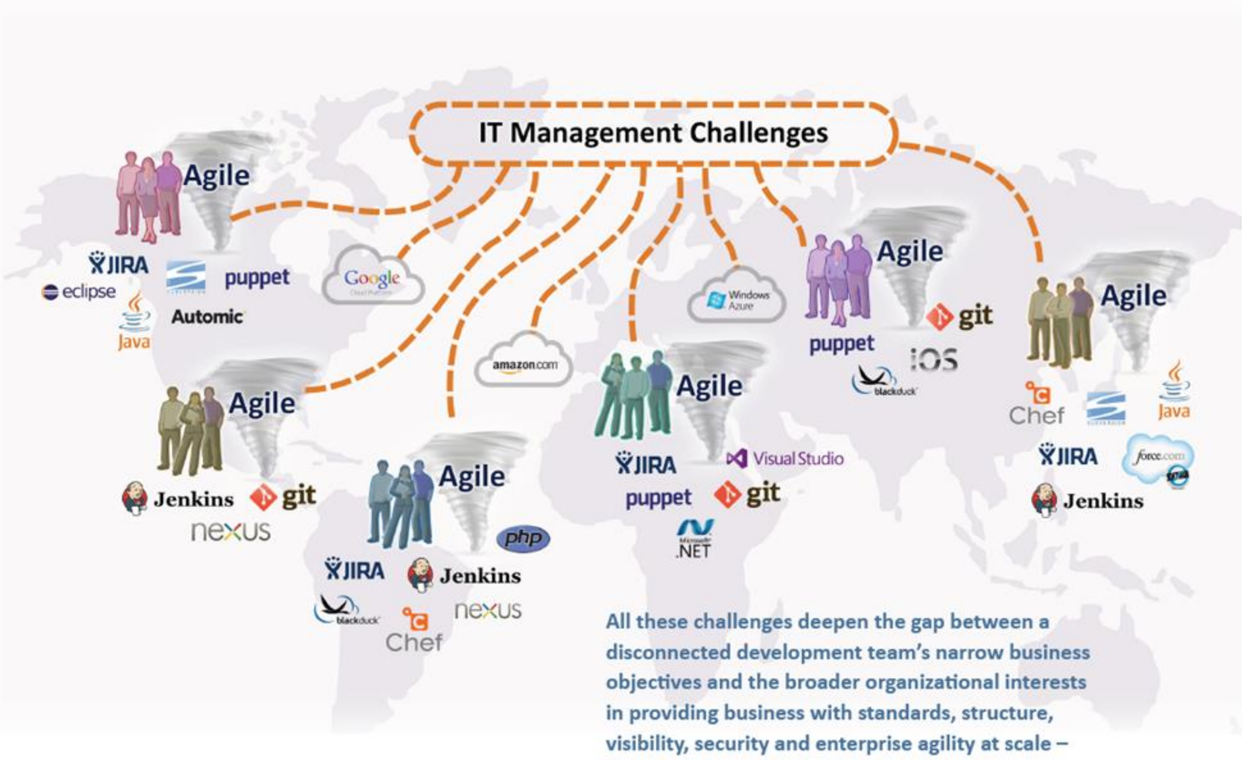
Application development in the enterprise is becoming – both by accident and by design -- Hybrid, Distributed and Multi-Source, and practical best practices have emerged. CollabNet contends that Hybrid Development has been successful when it is supported by an ALM system grounded in community-driven development principles.

This paper covers:

1. **Challenges organizations are facing** as they attempt to merge and manage multiple types of development and practices
2. **Limits of first-generation ALM platforms** in adopting and managing the emergence of “Hybrid” or “Bimodal” development practices
3. **Best practices** with respect to toolchains, process reuse and Community Architecture to enable “Hybrid” or “Bimodal” development

IT Management Challenge – Managing Disconnected Development Chaos

Between Waterfall and Scrum as points on a spectrum lie individual companies' unique blends of these practices and teams' particular processes, selection of tools, and areas of expertise. Layer on the globalization of customers as well as a company's particular development resources, and the result is a fractured landscape of disconnected tools, broken manual processes and isolated team members.



All these challenges deepen the gap between a disconnected development team's narrow business objectives and the broader organizational interests in providing business with standards, structure, visibility, security and enterprise agility at scale – the ultimate goal of Enterprise Architecture.

An amalgam of easy-to-acquire coding languages and tools, expensive and limited legacy systems, and application code contributions from internal, external and open source communities, has led to:

- Divergent, brittle and expensive-to-maintain development tools and processes – resulting in tool and infrastructure redundancy, development slippages and rising costs
- Lack of enterprise collaboration and leverage – limiting visibility and implementation of common processes to improve a company's development efficiency across the organization
- Limited control of Intellectual Property – leading to vulnerable corporate software development and usage governance plus the risk of IP (software assets) loss, as well as inability to integrate applications at the system levels, including the support of Enterprise Architectures and IoT strategies

While a majority of organizations are experiencing a bifurcation of IT, many are observed failing in systematically exploiting the potential benefits of a multi-modal approach, because of:

- Trying to apply a traditional IT mentality to next-generation applications and services
- Failing to adopt new technologies, relying on traditional vendors to solve next-generation challenges

These issues and outcomes can't be prevented by imposing single styles of software development practices.

Today's discussions on software development regularly include consideration of "Agile" versus "Lean" versus "Waterfall" as if these were by necessity or in reality "either/or" choices. However:

1. The need for agility will regularly confront the need for stability and vice versa. Some projects will remain inherently best suited for one development style versus another, and sophisticated organizations will adopt filters to match processes to a project, or to a project's phase.
2. No development team or development process is the same. Agile and Waterfall are two poles, and many teams select to work with a combination of processes at either pole or the large myriad of processes and toolchains in between.
3. While Agile adoption has made great strides in enterprise IT, the fact is that in the enterprise as a whole this adoption has been very uneven.

Attempting to shift workgroup approaches for the sake of participating in a unified system adds possibly unnecessary cultural and change management challenges to the mix. Indeed, development approaches don't need to be reconciled in order to be effectively managed.

The Emergence of "Hybrid" or "Bimodal" Development Approaches

Application development in the enterprise is becoming – both by accident and by design -- Hybrid, Distributed and Multi-Source. IT consultancy Gartner's recent reports speak to the trend they term Bimodal IT in which IT organizations are meeting digital business challenges by adopting a consistently executed parallel approach:

> "Successful IT organizations are meeting digital business challenges by adopting a bimodal approach to IT — A reliable Mode 1 that is focused on safety and efficiency, and an agile Mode 2 that is focused on flexibility and speed. These two modes have different infrastructure needs. The organization must deliver integration across the application portfolio, but this does not demand identical infrastructure approaches across both modes. The needs of IT operations organizations and the needs of application development organizations are different and sometimes opposed, complicating the selection of solutions, especially when developers are the primary users of a cloud solution."[1]

Gartner writes that the move toward "Bimodal IT" is logical given a current environment and states that "Enterprises are living in a digital Wild West, where there is tremendous opportunity, significant risk and huge uncertainty. In this environment, decreased cycle times for solution design, iteration and deployment are key to success."[2] In recommending that IT governance be adapted to oversee both "mode 1" and "mode 2," Gartner writes: "IT must vary its response to opportunities by becoming bimodal. This entails managing IT in two modes that are comprehensive, coordinated, and coherent, yet deeply different – and exploiting the benefits of both approaches. Mode 1 is traditional IT; Mode 2 is agile IT (See Table 1).[2]

[1] Gartner, Best Practices for Planning a Cloud Infrastructure-as-a-Service Strategy — Bimodal IT, Not Hybrid Infrastructure 3 March 2015 | ID:G00273574

[2] Gartner, Three Steps to Successfully Implementing Bimodal-Aware IT Governance 6 November 2014 |

Table 1: Differences between Mode 1 and Mode 2

Bimodal IT: Important Differences Between Mode 1 and Mode 2		
	Mode 1 (Traditional)	Mode 2 (Agile)
Goal	Reliability	Agility
Values	Price for performance	Revenue, brand, customer experience
Approaches	Waterfall, V-Model, high-ceremony IID*	Agile, DevOps, Kanban, Kaizen, low-ceremony IID, lean, lean startup
Governance	Work-plan-driven, approval-based	Empirical, continuous, judgment-based
Sourcing	Enterprise-class suppliers, long-term deals	Small, new vendors, short-term deals
Talent	Good at conventional process, large-scale and complex projects	Good at new projects with uncertain outcomes
Culture	Risk-averse, internal performance, metrics-focused	Risk-tolerant, business-outcome-focused
Cycle Time	Long release cycle (months to years)	Short release cycle (days to weeks)
Life Span	Long production life (years to decades)	Short production life (weeks to months)
*Iterative and incremental development (IID) comes in two forms. The original IID methods can be considered condensed waterfall, with eight-week time boxes typical. This “high-ceremony IID” is suited to Mode 1 – e.g., the original Rational Unified Process (RUP). The second form of IID – “IID lite” or “low-ceremony IID” – entails much more concurrent work, reduced document needs and less process ceremony. It is better suited to Mode 2 approaches – e.g., OpenUP and Microsoft Solution Framework (MSF) for Agile Software Development.		

Source: Gartner (November 2014)

Accepting the value and suitability of each approach and explicitly pursuing a multi-modal or bimodal strategy helps to resolve organizational power struggles which Gartner, among others, have observed:

> “Mode 1 people are likely to be the most outspoken critics of Mode 2: They see themselves as having the most to lose. In their eyes, Mode 2 people are, to quote one case example, ‘a bunch of irresponsible cowboys, who break all the project execution rules and get away with it.’” Mode 2 organizations are critical of Mode 1, too; Gartner writes: “As the IT leader, everyone is mad at you – the Mode 1 people for not following the established, long-held rules, and the mode 2 people for taking so long and bogging everything down.” [2]

Early Roadblocks to Tapping ALM’s benefits

Many organizations that recognize the need to manage multiple modes of development have adapted ALM platforms. The first generation of these platforms hasn’t resolved the issues, however, and the points of failure are instructive.

- Monolithic, Costly: First generation ALM solutions have proven to be too monolithic or costly to manage in the face of the industry's demands for increasingly fluid and diverse requirements.
- Inflexible: Legacy ALM tools are rigid, lacking the ability to adapt to the needs of the more agile workgroup. Some recent first generation SaaS platforms have been specifically focused on agile, and, in particular, a rigid form of agile, ignoring the myriad of diverse process and toolchain needs of the medium to large sized organization. Once defined, prescribed static processes are hard to change, with only little room for project members to adapt and evolve. Legacy core app teams are often further silo'd, and usually the last to be migrated.
- Lacking Openness and Flexibility: First generation solutions lack the openness and flexibility to embed preferred tools or to accommodate evolving or mixed methodologies. This often results in high 'configuration debt,' as new project teams struggle to cope with inadequate setups and processes. It is not uncommon for entire project teams to revolt and deploy their own development tools and processes in isolation, uncontrolled by corporate IT. This 'shadow IT' further impairs developer productivity, and compromises governance and transparency.
- Lack of Collaboration and Application Integration: With silo'd information, fractured processes, disparate tools, disjointed teams, and fragile integration, first-generation ALM has fallen short of its potential to calm the chaos of distributed software application development; the solutions do little to enable development and delivery collaboration within the workgroup, or drive larger scale IP and integration leverage gained through cross project collaboration.

CollabNet contends that multiple modes of development can occur successfully, in parallel, particularly with the support of a truly open, scalable and community-based ALM framework. If bimodal development is too binary to be realistic for an organization, and multi-modal development is a more realistic fit, it can also be managed via an open ALM platform.

Essentially, a managed open ALM approach for enabling hybrid development tools and processes has emerged – CollabNet TeamForge. This approach allows agile workgroups to use the required tools and processes to be quick and responsive, while at the same time providing the governance, toolchains and processes for legacy and regulated application development as well as overall system level-integration.

Benefits of Adapting an Open ALM Platform

A collaborative development architecture increases business agility through improved developer productivity, both within and across workgroups. Within workgroups, fast project startup is possible with process templates and quick access to assets and artifacts to get team members – even distributed and third-party collaborators - up and running fast. The positive collaborative work environment provides peer code review, reuse and safe builds. Across workgroups, the best IP assets and development resources of a company can be leveraged, and still further, enterprise architecture strategies can be employed to improve system-level application integration.

From an infrastructure cost and overall development productivity perspective, an optimized ALM Platform reduces overall operational expenses in administrative and infrastructure costs supporting globally-distributed teams and assets. Common resources and encouraged reuse creates more efficiencies as teams are aligned to follow the "best set" of practices as defined by the processes chosen to be codified into automated toolchains; as such, automation = quality and governance.

Finally, the same architecture that promotes open collaboration also governs and secures all software assets as IP with an enterprise-wide hierarchy of RBAC (Role-Based Access Control), providing audit controls, governance, and a system of record to all planning, development and delivery tool transactions.

Specific business benefits include:

- **Gaining real-time visibility** into project process, artifacts, and momentum, to ensure predictability. Better connect the business and development sides of the organization, avoid project failures, missed deadlines and budget overruns. The ability to aggregate development activities and data through all stages -- from requirements through release and even into deployment and application monitoring -- can provide this visibility and traceability. Management can take action to improve application development and delivery speed and quality from the initial iteration to the final deployment to upgrades and patches, over the lifecycle of the software.
- **Setting and keeping standards high** with a consistent set of on-demand development processes codified into repeatable toolchains across global teams supported and integrated with non-agile processes to enable adoption at the right pace for the enterprise. The ability to provide process and toolchain templates to regulate and automate any modes of application development and delivery brings consistency and offers a fast track to quickly start up new projects. Process templates support scale in any dimension and leverage software assets across teams & projects. A social, web-based collaboration mode accelerates innovation, enables distributed problem-solving, and ensures faster feedback and reuse of code.
- **Bringing Order to Chaos**: Data definitions, tool integrations and workflows are codified to provide end-to-end visibility and traceability at the project level AND overall cross-project/departmental view – enforced by RBAC.

An important aspect of bimodal IT is the need to be able to integrate applications built in distinct modes into a larger system of things (IoT, Enterprise architecture). Gathering all an organization's projects in one collaborative platform improves:

1. **Standards and Integration** - The ability to create common standards for interconnecting components and applications built in distinct modes,
2. **System Debug** - The ability to debug interconnected applications by tracing back into data formats and tools in a standard way, and
3. **Reuse and Sharing of IP** - The potential to drive reuse and sharing of IP across the applications built in these modes to drive costs down, quality up, and improve time-to-delivery.

Best Practices in Adapting an Open ALM Platform

CollabNet has successfully partnered with thousands of organizations to implement flexible ALM systems, featuring codified processes that fit the nature of each development team as well as organizational wide IP reuse, application integration, and governance. For 15+ years CollabNet has pioneered collaborative and agility-driven software development for globally distributed teams. Instead of struggling with disparate and disconnected processes and tools, we realized that organizations and distributed development teams could realize greater innovation through a set of modern and transparent collaborative methodologies on a productized Web-based platform available in the cloud as a privately managed services or on premise - CollabNet TeamForge.

In building and iterating on TeamForge, CollabNet has identified the following best practices, which are directed, in the chart below, to managers and executives seeking to tap the benefits of hybrid development:

Table 2: Features of Collaborative Architecture for Hybrid Development

Features of a Collaborative Architecture for Hybrid Development	
Manage Formal Software Assets & Informal Artifacts	In addition to 'formal' development artifacts (code, issues, builds, tests, releases) pay attention to the management of 'informal' assets as well, such as architectural blueprints, discussions and the people associated with them. Every asset should be versioned, indexed and associated with its relevant items to establish traceability.
Gather Enterprise Software Lifecycle data	Capture and report key development and delivery metrics across teams, projects, tools and methodologies. Empower all levels of management to make informed decisions on resources and projects.
Leverage Your Point and Legacy Tools	Ensure that your platform allows you to embed and synchronize your favorite point tools (proprietary or open-source). You should also be able to snap-in tools installed on a separate server or hosted in a private or public cloud.
Enforce Access Control, Enterprise-Wide	Govern access by your internal and external teams across all projects by setting role types. Role-based access control greatly simplifies the on- and off-boarding of project members and contractors, and automates the process of managing entire groups of projects and developers.
Organize Your Software Projects	Foster organizational alignment, collaboration, and reuse by organizing your business, technology and enterprise architecture into collaborative and hierarchical categories, groups, and projects. Standardize community structures across your organization, simplifying the navigation and management of large, complex user communities.
Fast-track Project Launches	Simplify the initiation of new projects by using standardized templates to codify your corporate development processes and ease the on-boarding of project members with automatic role inheritance. Templates accelerate the learning curve for new project members, by synthesizing large amounts of project data into instantly recognizable standard formats.
Securely On and Off-Board Project Members	Ensure that you can efficiently and securely manage all your users and projects. Whether there be several, hundreds or thousands of users, you must be able to on- and off-board them quickly and securely.

Gain Transparency and Traceability	Any asset and artifact should be easily and quickly accessible. Search engines must deliver results immediately, while adhering to corporate security policies. Search results must include the context of the artifacts and users associated with them, so you can trace back at any point in the application development and deployment lifecycle to all related items such as wiki entries or code commits.
Codify Company Best Practices	Industry development methodologies evolve and change over time. As this paper lays out it is likely you are using a variety of methodologies in-house today. Drive toward standardization, without impeding flexibility, through the ability to codify any methodology, including Scrum, Kanban, Waterfall, or anything in-between.
Manage Task Allocations and Estimations	Task boards are powerful, as are visual tools that allow teams to assign tasks to individual team members and track status at a glance. Ensure that your platform supports graphical task boards that are seamlessly integrated into the artifacts repository.
Access from Desktops or Mobile Devices	Ensure that IDEs, such as Eclipse and Visual Studio, and mobile devices integrate seamlessly into your development landscape. Your developers should be able to access and update code and artifacts, efficiently and in real-time.
Collaborate and Re-Use Project Assets	To drive asset reuse, ensure that developers can discover and share information easily. Searches must include associated lifecycle assets including code, builds and file releases, artifacts, documents and discussions. Search results should let you trace back to the originating requests, such as requirements and projects.
Automate Workflows and Notifications	If a build fails, it's critical for the developer to be informed right away – including via automated notifications by email or alerts directly within their IDE. In the event of build failure, the system should assist in the root cause analysis, for example by pinpointing log files and responsible servers.

Summary

Rapid growth driven by globalization, IoT and other time-to-market pressures has resulted in development organizations made up of a wide array of disconnected teams, tools, processes and core software assets. Legacy, stable and core data rich applications required to run the business are frequently managed in waterfall and manual methods. Conversely, customer-facing, high-volume interactions with rapidly changing requirements and loads are frequently developed and delivered in the most agile way possible, with processes and tool chains optimized around dynamically changing market use cases.

Transformation to handle the range of hybrid development tools and application development and delivery processes begins with the decision to change -- to break out of the status quo and turn an organization's software assets into a strategic advantage instead of a risk. Sound change management begins with centralization and collaboration of company-wide Intellectual Property, toolchains and people into an open Application Lifecycle Development and Delivery platform – one which can provide the real-time visibility, traceability and process automation that leads to improved business agility while meeting the strictest regulatory compliance standards. A successful ALM development platform must be flexible enough to accommodate:

- a mix of technologies, stakeholders and development methods,
- the ability to codify the tools and processes of the best workgroups into reusable organizational-wide toolchain templates to drive consistency, quality improvement, time-to-market advantage, cost savings, and companywide insight, and
- collaboration and re-use of these high-value assets within and across projects to drive leveraged IP reuse and enable companywide enterprise architecture integration

Join CollabNet and our 10,000 customers in 100+ countries and build a successful and dynamic Hybrid development environment to fit your business needs.

About CollabNet TeamForge®

TeamForge, CollabNet's flagship product, is the industry's only open and extensible collaborative software development and delivery platform for distributed teams. It helps organizations of all sizes improve their collaboration, agility, processes, IPs, and tools via a centralized, secure, web-based system. With TeamForge your entire organization can develop and deploy software and become more agile at scale, with the ability to scale to tens of thousands of users.

About CollabNet

CollabNet® is the creator of Subversion® and a pioneer in cloud-based application lifecycle management solutions for collaborative agile software delivery at scale. CollabNet provides industry-leading products plus agile consulting and training services to help organizations of all sizes develop and deploy software faster. CollabNet's flagship product, TeamForge®, provides customers with an open and extensible collaborative software development and delivery platform to increase collaboration and application release efficiency across larger, distributed teams. TeamForge users also gain better governance with enhanced visibility and traceability across the software development lifecycle. For smaller teams, CollabNet provides CloudForge®, a cloud-hosted version of Subversion, Git, and TeamForge that enables fast project starts on-demand. CollabNet has been recognized for 10 consecutive years as an *SD Times* 100 industry innovator and is consistently positioned as a leader within tier one industry analyst reports, including an independent research firm report on "Application Lifecycle Management Tools."

For more information, please visit **www.collab.net**.

SB-111714-9

CollabNet, Inc.
4000 Shoreline Court, Suite 300
South San Francisco, CA 94080
Tel: +1 650.228.2500 | Fax: +1 650.228.2501 www.collab.net | info@collab.net

Blog: **blogs.collab.net**
Twitter: **twitter.com/collabnet**
LinkedIn: www.linkedin.com/company/collabnet-inc
Facebook: **www.facebook.com/CollabNetHQ**