



Global Agile Development

CEI's Approach to Successful IT Outsourcing

Number: 04062006

CEI
1910 Cochran Road
Manor Oak Two, Suite 230
Pittsburgh, PA. 15220

www.ceiamerica.com

Table of Contents

AUTHOR’S NOTE	3
EXECUTIVE SUMMARY.....	3
WHY GLOBAL AGILE DEVELOPMENT?.....	4
THE COMPONENTS OF GLOBAL AGILE DEVELOPMENT	5
TRADITIONAL FRAMEWORKS	5
<i>eSCM</i>	5
<i>CMM and CMMi</i>	5
<i>RUP</i>	6
AGILE DEVELOPMENT PROCESSES	6
<i>Extreme Programming</i>	6
<i>Iterative and Spiral Development</i>	7
GLOBAL AGILE MANAGEMENT PROCESSES	7
<i>PMP</i>	7
<i>Scrum</i>	7
OFFSHORE CHALLENGES WITH GLOBAL AGILE DEVELOPMENT	9
CLIENT ENABLEMENT THROUGH ESCM.....	11
CASE STUDY: GLOBAL AGILE DEVELOPMENT IN PRACTICE	12
APPLIED AGILE PRACTICES.....	13
CORE CMM PROCESSES APPLIED TO THE AGILE SPRINTS	14
CONCLUSIONS	15
SOURCES.....	15

Author's Note

This paper addresses the integration of leading-edge software development concepts applied to business processing and IT outsourcing efforts. Because CEI is one of few IT service providers to adopt this integrated model, it offers a unique value-add for clients considering outsourcing. Additional CEInformant white papers can be found on CEI's website at www.ceiamerica.com.

Executive Summary

In an IT climate of fluid requirements, multi-phase schedules and distributed software development teams, Global Agile Development enables CEI to provide business processing and IT outsourcing services to independent software vendors (ISV), business process organizations, Fortune 1000 and mid-market companies.

Global Agile Development is a combination of traditional software development methods and Agile development processes that provide solid software development and management with the agility needed to address changing requirements and foster communication and interaction among clients and onshore and offshore development teams.

This paper accomplishes the following:

- Outlines and describes the components of Global Agile Development
- Discusses how Global Agile Development addresses the unique challenges of IT outsourcing using an offshore business model
- Presents a case study in which CEI used Global Agile Development to develop a software product for an ISV

Terminology	
CMM and CMMi	Capability Maturity Model. Methodology frameworks that provide a core foundation of software development principles. CMMi is the newer methodology framework of the two.
eSCM	eSourcing Capability Model. A best practices outsourcing framework for service providers that has the guidelines for successful business processing and IT outsourcing engagements.
ITO	Information Technology Outsourcing. These services typically involve a multi-year business partnership between the client company and the outsourcing vendor.
PMP and PMI	Project Management Professional and Project Management Institute. PMI developed a certification for PMP based on traditional development processes. Describes the roles and responsibilities of project managers and team members.
RUP	Rational Unified Process. Focuses on Agile processes including short incremental development cycle, close teams and testing early in the cycle.
Scrum	Management concept around which Agile management techniques are built. Defines how deliverables are estimated and created.
XP	eXtreme Programming. An Agile software development process that emphasizes simple design, testing, refactoring, and teamwork.

Why Global Agile Development?

Global Agile Development is CEI's ITO software development process for developing new applications and application maintenance using distributed US-based and offshore software teams. CEI developed Global Agile Development to meet the changing requirements and multi-phase delivery schedules endemic to most IT projects.

IT firms are turning their focus to Agile development methodologies because of poorly scoped projects and integration issues. In Agile development, firms anticipate and embrace changing requirements, instead of resisting them. In addition, development centers on short iterations and stresses people and communication rather than processes and heavy documentation.

CEI's Global Agile Development combines Agile methodologies with traditional software development processes to provide ITO services. The structure of Global Agile Development can be compared to that of a skyscraper, which has a core foundation below ground and a flexible framework above ground. Global Agile Development is grounded in solid software principles promoted by the Capability Maturity Model (CMM and CMMi), traditional processes promoted by Project Management Professional (PMP) certifications and emerging new process frameworks supported in the eSourcing Capability Model (eSCM). Upon this foundation, Global Agile Development uses Agile development methodologies, including Rational Unified Process (RUP), Extreme Programming (XP) and Scrum.

The Components of Global Agile Development

This section provides an overview of the software development processes and frameworks that comprise CEI's Global Agile Development.

Traditional Frameworks

eSCM

Developed by the Software Engineering Institute at Carnegie Mellon University (CMU), eSCM is a set of best practices for creating and sustaining an outsourcing arrangement between a vendor and a client. The model focuses on key sourcing business processes, such as contracts, financial processes and knowledge transfer. These processes typically are not part of software-based development methodologies but are critical for maintaining a long-term relationship with an outsourcing partner.

CMM and CMMi

CMM is a five-level assessment model developed by the Software Engineering Institute at CMU to describe best engineering and management practices. CMMi is a newer model and prescribes additional process improvements for software development organizations. The models include the following process areas: requirement gathering, project planning and tracking, quality assurance, configuration management, training, collaboration and peer reviews. **Figure 2** shows the five levels of CMM and CMMi.

Level	Focus	Key Process Areas
5 Optimizing	<i>Continual process improvement</i>	Defect Prevention Technology Change Management Process Change Management
4 Managed	<i>Product and process quality</i>	Quantitative Process Management Software Quality Management
3 Defined	<i>Engineering processes and organizational support</i>	Organization Process Focus Organization Process Definition Training Program Integrated Software Management Software Product Engineering Intergroup Coordination Peer Reviews
2 Repeatable	<i>Project management processes</i>	Requirements Management Software Project Planning Software Project Tracking & Oversight Software Subcontract Management Software Quality Assurance Software Configuration Management
1 Initial	<i>Competent people (and heroics)</i>	

Figure 2. Five Levels of CMM and CMMi

CMM and CMMi strive to achieve process consistency, predictability and reliability. CEI's Global Agile Development uses CMM and CMMi as a methodology that explains *what* to do to achieve successful software development.

RUP

RUP is a framework that provides a full lifecycle set of processes for software development. The methodology combines iterative and spiral development practices which can apply to projects that employ traditional software processes as well as projects that use Agile techniques.

RUP focuses on the use of modeling tools, such as the Rational software suite. Currently, Agile projects do not use formal process tools for modeling. However, formal modeling tools are expected to become more adaptable to Agile processes and therefore will play a more significant role in future Agile development.

The traditional frameworks described in this section serve as the solid process foundation upon which Agile development and management processes are based.

Agile Development Processes

Extreme Programming

XP is the most prominent of development processes under the Agile umbrella. XP supports Agile philosophies that stress individuals over process, working software over documentation, customer involvement over negotiation and response to change over a rigid plan. XP works well with changing requirements, incorporates small iterations and focuses on people and interaction.

XP improves a software project by enabling communication, simplicity, feedback and courage. It emphasizes simple design, testing and refactoring and teamwork. In particular, testing using harnesses, frameworks and automated processes is critical for success of an XP project. Furthermore, small releases enable the software development product to be released quickly in sets of related features. These sprints can be likened to traditional phases in a spiral or iterative software development process, which can then be managed with RUP and CMM best practices.

Some of the core XP practices that CEI employs include the following:

XP Practice	Description
Planning Game	Quickly determine the scope of the next release based upon business priorities and technical estimates
Small Releases	Implement shorter release cycles to keep projects on-time, on-budget and enable quicker time-to-production. Also, get feedback earlier in project.
Simple Design	Great systems are typically built from fine level components, and the design of a software system based upon XP should consider this.
Testing	Testing is very important. Unit tests are very important and automated tools

	facilitate the process so that the XP can scale through continual regression processes.
Refactoring	A properly modeled system allows for the restructuring of the system without changing behavior. Duplicate effort is removed; communication is improved; and code is simplified and more flexible.
Collective Ownership	Anyone can change any code anywhere in the system at any time.
Continuous Integration	Able to ship the system at any time. In essence, every time a task is finished, an integration and build occurs. Enabled by automated testing, this may happen many times per day.
On-site Customer	Live user on the team answers questions as they arise.
Coding Standards and Examples	Coding standards are voluntarily adopted by the development team and not imposed by someone higher up.

Iterative and Spiral Development

Iterative development breaks the project into phases, each of which produces a deliverable and builds on the code base and documentation from the previous phase. The first phase delivers the most basic functionality, which is followed in the next phase by the next logical set of features. Each phase includes analysis, design, implementation and testing.

Similarly, spiral development is broken down into phases eliminating the need to define the entire system upfront. However, instead of delivering the system by function, spiral development delivers features according to risk.

Global Agile Management Processes

CEI's Global Agile Management processes are a combination of traditional, PMP-driven management and Agile Scrum-driven management techniques.

PMP

PMP is a certification promoted by PMI that fosters good project management skills. In CEI's experience, good project management is a primary building block for Scrum management processes that form the core of a sprint-based software development approach.

Scrum

Scrum is a process for incrementally building software in complex environments. Scrum is a scalable, flexible, low-risk and hyper-productivity technique. It is not a methodology, but rather an agile, lightweight management framework used to manage software and product development. Scrum is a wrapper for existing engineering practices, including XP.

A key concept of scrum is teamwork, which ties in with the Agile philosophy of people over process. Daily status check meetings are a critical factor for the success of a scrum team. During a daily scrum, a Scrum Master helps the team to resolve issues, ensures all team members make

progress, records decisions made, tracks action items and keeps the meetings short and focused. In addition, the Scrum Master identifies initial backlog in the sprint, works to reduce product risk and empirically measures progress toward the goal of delivering an incremental set of product functionality.

The basic premise of scrum is “trust but verify”. If members are committed to the team and the project, and if management trusts the team members, then individuals spend time being productive instead of justifying their work.

Figure 3 illustrates the scrum framework, in which quality evolves not from micromanaging the process, but from establishing appropriate exit and review practices.

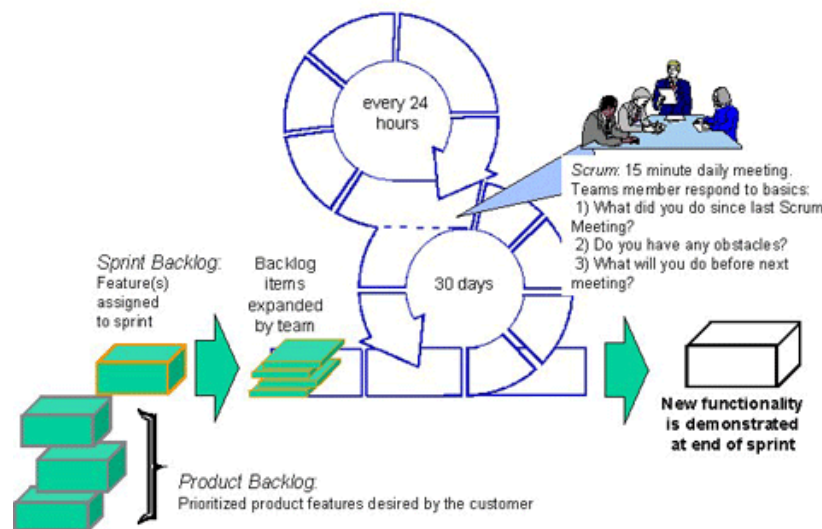


Figure 3. Scrum Framework (source: www.controlchaos.com)

Scrum is a powerful concept given the following observations:

Focus on teamwork. The team operates as a unit and moves, assembles and prioritizes as needed to accomplish goals.

Daily status communication of status. Daily meetings create a high degree of visibility. The team focuses on making decisions and removing impediments in real-time. Over time, management overhead is low.

Offshore Challenges with Global Agile Development

In an Agile State-of-the-Art report by the Data & Analysis Center for Software, participants in an Agile workshop identified three important success factors for Agile processes: culture, people and communication. The culture of the organization and its employees must be conducive to Agile processes. In addition, people must be well-trained and work in close communication with each other. With these considerations, implementing Agile processes within multiple development teams across different geographies and time zones presents a unique set of challenges.

The table below outlines key practices that CEI implements to make Global Agile Development work for distributed, multi-national projects.

Agile Process/Principle	Offshore Challenge	Best Practice Solution
Planning Game/Scrum	High turnover in India can cause team synergy issues	<ul style="list-style-type: none"> • Shadow and backup developer and trainee for every 10 developers • Do not distribute EJB & UI work across locations for the same use case. Thus, organize by functionality and not by role or activity. • Keep each developer assigned to just one use case. The developer takes responsibility for the deliverable (including bug fixing). • In order to allow the team to be autonomous, a cultural issue of getting the Indian team to acknowledge progress is minimized by the CruiseControl-based builds and Scrum-based visibility meetings. <ul style="list-style-type: none"> ○ This is not a patronization issue but rather an observation on the command and control nature of India. • Each sprint has the following components: <ul style="list-style-type: none"> ○ Design on tasks for next sprint ○ Development of current sprint tasks ○ Bug fixing on prior sprint tasks • Sprint kick-off entails review of: <ul style="list-style-type: none"> ○ Use case ○ UI prototype ○ Design ○ Task list • Sprint wrap-up: <ul style="list-style-type: none"> ○ Demo software ○ Review code quality ○ Post-mortem to identify and mitigate issues
Small Releases	Coordinating between multiple distributed teams stretches a sprint	<ul style="list-style-type: none"> • Given additional management issues, a sprint of 4-6 weeks is reasonable compared to Agile-recommended 1-2 week sprints.
Simple Design	Complexity must be factored out with the Indian team	<ul style="list-style-type: none"> • Examples, standards and checklists are created so that the Indian team gets the buy-in. • This becomes important for the offshore team in that they are implicitly autonomous given time zone differences.
Testing	Distributed teams across	<ul style="list-style-type: none"> • Separate features by functionality and subsystems so

	time zones can slow down progress	<p>that a geographic group is not as affected, as in case of build issues, test failures, etc.</p> <ul style="list-style-type: none"> Unit tests created by the offshore team provide a feedback loop that ensures they understand the concept.
Refactoring	Refactoring can impact current features due to split feature sets	<ul style="list-style-type: none"> Standards for EJB, Flash (frameworks, coding standards, etc. Detailed designs to be created by individual programmer and approved by offshore lead/designer prior to beginning work on the task Incremental reviews of code by offshore lead/designer
Continuous Integration	Slower line speed with offshore can cause productivity delays	<ul style="list-style-type: none"> Scripts to sync up code during evening hours in India and before each individual comes to work Separate modules for offshore and onshore teams Dedicated link Single source point - offshore and on-site teams use the same repository
Collective Ownership/Self-organizing Teams	Given multiple teams, collective ownership is based upon modules and features	<ul style="list-style-type: none"> Incremental reviews of code by offshore leads/designers Factor time for on-site and off-site leads to troubleshoot problems in each sprint
On-site Customer (Face-to-face communications)	Remote teams are not all in one physical place	<ul style="list-style-type: none"> Teleconference scrum meeting every day in morning EST or evening PST Scheduled weekly and monthly meetings Online collaboration with web-based tools Remote business analysts and project manager come to the US to assimilate into the context of the project. Use cases are created in the US and the offshore team creates the unit test cases to validate that they understand the requirements and specifications. UI prototypes and wire frames provide examples. Daily builds uncover issues and released feature sets in sprints enable customer feedback. An on-site demo/integration machine allows the feedback loop.
Coding Standards and Examples	Design and architecture skills are stronger in the US	<ul style="list-style-type: none"> On-site leads create design and obtain signoff from client prior to start of sprint. Formal detailed design review signoff by offshore lead

Client Enablement Through eSCM

In any outsourcing arrangement, CEI works with a client to put the core processes in place to ensure a successful outsourcing relationship. However, this is not always easy; sometimes it can be a major transformation for companies.

From experience and anecdotal evidence, CEI understands the challenges that outsourcing presents clients and therefore provides business processing and ITO QuickStart services. These are based upon eSCM, developed to enable the selection and improve the capabilities of ITO service providers.

Client enablement begins with having a key business/technical resource work with a representative from the client. This representative should be familiar with the environment and serve as a bridge between to the outsourcing and offshore teams. Process mapping is then done to map the client's processes to set of mutually agreed software delivery standards. **Figure 4** shows how various facets of a project can be divided over multiple global delivery centers. Business relationships, metrics and general service level agreements define the level of standards between the two companies. Cultural change issues must then be examined to determine the impact of the pilot or outsourcing decision to the client organization.

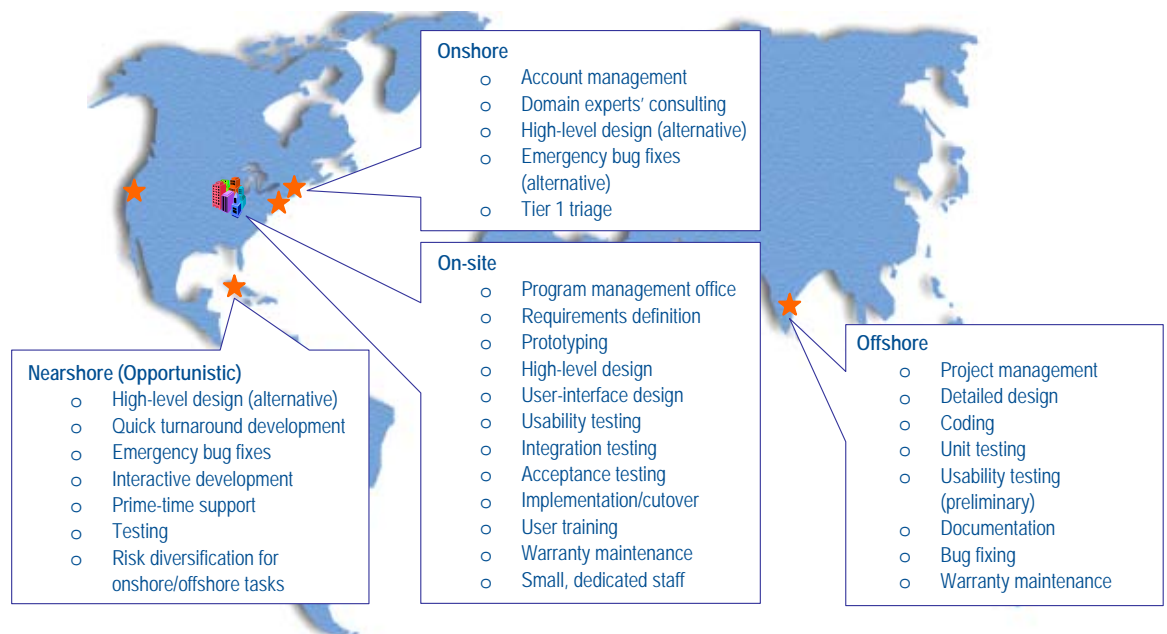


Figure 4. CEI Global Delivery Model

Case Study: Global Agile Development in Practice

CEI's client, a Denver-area ISV, is a premiere supplier of application software to the insurance industry. The client wanted to design and develop an object-based software package that would keep insurance business activities flowing smoothly. The new software package needed to interface with existing dated PowerBuilder-based systems while offering flexibility to grow as business needs expand.

The company employed CEI who composed a blended team to design and develop the system in a cost-effective and timely manner. The on-site team included coordinators, business analysts and users. The off-site teams consisted of program managers in Pittsburgh, PA and two offshore teams in Chennai, India. Each offshore team had six to eight developers (each with lead and designer) reporting to an offshore project manager. **Figure 5** shows the software governance model applied to the project.

The team applied Global Agile Development processes in 30-day sprint cycles to create a J2EE-compliant solution that achieved application server portability across WebSphere, WebLogic and JBoss as well as database portability across Oracle and DB2.

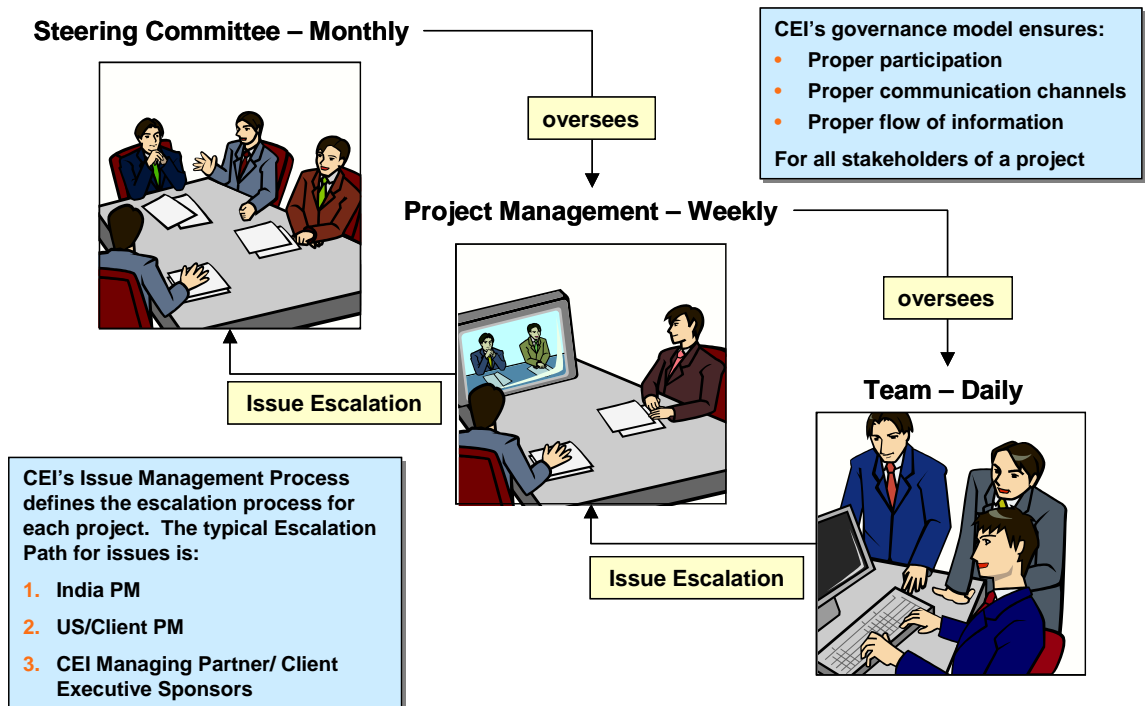


Figure 5. Project Governance Model

Applied Agile Practices

The following table describes some of the key Agile practices CEI used during the multi-phase software development project.

Agile Practice	Application
Design & Coding Standards	<ul style="list-style-type: none"> • Coding standards, checklists and examples for the following: <ul style="list-style-type: none"> ○ J2EE ○ Flash ○ EJB ○ Global navigation, interface design and behavior
Test-Driven Development	<ul style="list-style-type: none"> • Approximately 50% of source code is unit test case code (non-production code) • Write test cases first
Continuous Integration	<ul style="list-style-type: none"> • Use CruiseControl for the following: <ul style="list-style-type: none"> ○ Automated build scripts using ANT, StarTeam plug-ins ○ Self-testing code using JUnit, Cactus, ASUnit, etc. ○ The master build runs every 90 minutes <ul style="list-style-type: none"> ▪ The latest source code is checked out of the configuration management system ▪ Every file is compiled from scratch ▪ Resulting Java classes are deployed for execution (i.e., put into JARs, WARs and EARs) ▪ A suite of tests (i.e. 3500 test classes) is run ▪ Notifies the appropriate developers of build and test case errors • Integration testing where all processes are automated, enabling integration errors to be found quickly • Offshore and on-site teams use the same StarTeam repository
Simple Design	<ul style="list-style-type: none"> • Simplicity of design is emphasized • Architecture and framework are solidified in the initial sprints
Improve Quality by Feedback	<ul style="list-style-type: none"> • Reported bugs are used to improve the quality and coverage of unit test cases
Small Releases	<ul style="list-style-type: none"> • Month-long sprints give the best productivity • Sprints are nonlinear and flexible • Sprints are used to evolve the final product
Small Team Size	<ul style="list-style-type: none"> • 6-8 member development teams provided best productivity for the project (off-shore and on-site teams)
Daily Scrum	<ul style="list-style-type: none"> • The daily scrum meeting does not exceed 30 minutes • Each developer answers the following questions: <ul style="list-style-type: none"> ○ What have you done since the last daily scrum? ○ What will you do between now and the next daily scrum? ○ What is getting in the way of you doing your work?
Project Tracking - Manager's Perspective	<ul style="list-style-type: none"> • Daily scrum meetings • Work burn-down chart /graph on the task list • Task list updated everyday after the daily scrum
Product Backlog	<ul style="list-style-type: none"> • The product manager is responsible for maintaining product backlog, along with estimates for how much work is required for a backlog item • As sprints build the product, the product manager re-estimates (feature is only partially implemented) or removes (feature completed) backlog items
Sprint Backlog	<ul style="list-style-type: none"> • Team leads maintain the sprint backlog

	<ul style="list-style-type: none"> • Project manager empirically manages the product backlog so that sprint backlog and release backlog reach zero when needed • Done primarily by adjusting the sprint and release contents or by modifying the release date
Self-Organizing Project Teams	<ul style="list-style-type: none"> • Team realizes it has the full authority to do what it deems necessary - a sense of liberation and empowerment. • Diversity of skills (J2EE/EJB, Flash, ActionScript, Functional Testing, etc.)
"Demo or Die"	<ul style="list-style-type: none"> • Every sprint has a wrap-up meeting where the development teams demonstrate the part of the product developed in that sprint • The audience for the demo are development teams, UI designer, QA engineer, support, marketing, industry colleagues and senior management

Core CMM Processes Applied to the Agile Sprints

The following table describes some of the core CMM processes that were applied to the Agile sprints during the project.

CMM Process	Application
Issue Management	<ul style="list-style-type: none"> • Bug and issue tracking through StarTeam (across offshore and on-site teams) and Bugzilla (internally) • Daily Scrum
Requirements Management	<ul style="list-style-type: none"> • Requirements artifact (use cases, supplementary spec, UI prototypes and standards documents) versions are pinned against the sprint • Target to build against a specific version of these documents • Product backlog • Sprint backlog
Software Development	<ul style="list-style-type: none"> • Code and design reviews - peer and team lead reviews • Continuous integration • Functional milestones are identified by the project manager
Configuration Management	<ul style="list-style-type: none"> • StarTeam (source control, integrated bug tracking, issue tracking, client-specific branching) • Automated master build process (i.e. CruiseControl) • Requirements version management • StarTeam topics for messaging – documenting discussion threads
Software Tracking and Control	<ul style="list-style-type: none"> • Scrum sprint task lists • Product backlog and backlog graph • Sprint backlog
Risk Assessment	<ul style="list-style-type: none"> • Risk Assessment Meeting every week at the client office where the project manager and team leads raise and mitigate risks • Risks for offshore teams are presented in the weekly status meetings
Estimation	<ul style="list-style-type: none"> • Follow estimation guidelines that have been modified after every sprint (as a feedback to reflect the estimates correctly) • Task lists for every sprint are based on the guidelines
Change Management	<ul style="list-style-type: none"> • No change can be introduced in the middle of a sprint • A change goes to the product backlog and is taken up in the subsequent sprints
Training Program	<ul style="list-style-type: none"> • The on-site team trains the offshore teams in tools and new developer orientation
Software Planning	<ul style="list-style-type: none"> • Pre-sprint work • Daily scrum meetings between on-site and offshore teams

Conclusions

Traditional software development methodologies and Agile processes can be successfully applied in a scalable manner to a distributed development team across regions and time zones. CEI's Global Agile Development uses CMM/CMMi, PMP and eSCM to form the core foundation for delivering successful outsourcing projects. Agile processes, including XP and scrum, build upon this foundation by enabling flexibility in handling requirements, short release cycles, client interaction and team communication.

Incorporating Agile development processes in a blended onshore/offshore development team structure presents unique challenges. Only vendor companies that have experience in implementing Agile processes offshore should use Global Agile Development. CEI has been an early mover in this effort and is therefore well positioned to provide its thought leadership in business processing and IT outsourcing to ISV, Fortune 1000 and mid-market companies. CEI will continue to refine Global Agile Development as needs evolve.

Sources

“Agile Software Development: A DACS State-of-the-Art Report”. Data Analysis Center for Software, January 2003.

Extreme Programming Explained. Beck, Kent. Pearson Education. 1999.

www.agilealliance.org

www.controlchaos.com