A Practical Management and Engineering Approach to Offshore Collaboration

James Cusick and Alpana Prasad, *Wolters Kluwer Corporate Legal Services*

The Global Software Development model provides a tactical approach for companies pursuing or planning to expand into offshore development.

Global software development is the norm more than ever before.[1] Firms developing or maintaining software products can't ignore global software development's impact. Our experiences in leading Web development efforts with global teams have led us to develop a set of successful, concrete practices that minimize risk and maximize effectiveness. Managing multiple simultaneous projects and production support with a global staff over the last several years has also made us aware of traps to avoid. Technical managers and project leaders and managers can apply our findings to help ensure consistency, institute a common approach, and develop best practices around global collaboration.

We present a practitioner's view of our model for offshore development and insights into our management and engineering techniques, which can be replicated in other environments. This article adds to the existing literature by providing a structural framework and the guidelines necessary to maintain the quality of offshore engagements.

We have over 35 years of combined software development experience in settings from start-ups to Fortune 10 R&D. We also have experience teaching software engineering and developing new ideas and processes in the field. Additionally, we've both managed numerous offshore projects over the past decade, working with a variety of partners in China, the UK, India, and Japan.

Background

Wolters Kluwer (WK) is a Netherlands-based publisher and information services provider with global operations. The experience we document here focuses on global development teams managed from the New York-based Corporate Legal Services Division. The practices we describe are grounded on numerous projects performed primarily in the US (onshore) and India (offshore). Much of our traditionally onshore work has migrated offshore over the years through outsourcing deals with multiple vendors. This move mimics the market at large, where 90 percent of US executive boards have discussed global delivery options.[2]

Our offshore engagements today are mature. We're in the final phase of our outsourc-

0740-7459/06/$20.00 © 2006 IEEE

Table 1
Summary of recommendations

Issue category	Recommendation
Organizing for offshoring	■ Set clear criteria for offshoring, and analyze whether a project is an offshore candidate. ■ Break large projects into medium-size bundles for offshoring. ■ Plan for interim deliveries to allow sufficient time for review. ■ Limit phase durations to keep control. Shorter phases are easier to track and manage. ■ Document and baseline requirements. ■ Require formal document review and sign-off to move into development.
Communications and management	■ Require both structured and unstructured communication. Structured communication provides a regular means for status updates, and unstructured communication encourages team bonding. ■ Track all issues assiduously. ■ Maintain open channels of formal and informal communication. Track and review issues diligently so that information doesn't get lost in emails or chat. ■ Create and baseline comprehensive analysis and design documents before transitioning a project offshore. Code review and other standards documents should support these documents. ■ Onsite staff can be skittish when work moves offshore, so communicate career paths.
Managing staff	■ Retain key leads over time, enabling them to develop application, architecture, and process expertise. Carry these leads across projects so they can guide new team members on processes and standards. Your goal should be continuity of offshore staff. ■ Retain domain expertise onsite and offshore. Protect and reward senior staff. ■ Manage vendor experience level—rigorously vet candidates and take only the well-qualified. ■ Select leads carefully. Leads can ensure success, so take only ones with lead-level experience. ■ Lead time in getting appropriately skilled staff is increasing, so forecast resource needs early. ■ Transition some key leads to production support to ensure that application and process knowledge is smoothly incorporated within product support releases.
Infrastructure issues	■ Infrastructure is necessary to begin a project, so resolve network and facilities issues early. ■ Our model depends on a core framework, so develop core code infrastructure in advance. ■ Mirrored computing environments require specifications and investment, so start early.
Managing development	■ Require adherence to best practices and standards to drive quality. ■ Prepare and enforce a standard ramp-up guide. ■ Require interim deliverables to ensure quality. ■ Establish performance standards and tools for review ahead of time. Use these as a benchmark to review new or modified pages' performance. ■ Manage support through bundles of defects. ■ Review all defect fixes onsite.
Quality and data privacy	■ Enforce quality through coding standards and verification. ■ Protect data sent offshore through confidentiality agreements.

ing agreement in the model John Berry described,[3] which comprises

1. strategy development,
2. selection process,
3. relationship building, and
4. sustained management.

This article focuses on sustained management and provides a complete management and engineering approach to working with offshore collaborators. Table 1 shows the set of recommendations we've developed, which summarize our key findings from our work in managing offshore projects. These recommendations can help guide offshore work.

The projects within our scope of experience have had as few as one or two developers and as many as 40. All the projects employed a multitier Web-based application service provider ar-

Table 2
Life cycle model

	Phase				
	Concept	**Analysis**	**Design**	**Construction**	**Quality assurance**
Location	Onsite	Onsite	Mostly onsite	Offshore	Onsite and offshore
Number of resources	2–4 IT resources	2–4 IT resources	2–4 IT resources	10–20 IT resources	10–20 IT resources
Key deliverables	High-level requirements estimates Assumptions and risks New infrastructure	Detailed requirements	Class model Data model Interface definitions Sequence diagrams	Code Unit tests Code review	Defect fixing
Actors	Onsite and offshore lead	Onsite and offshore lead	Onsite and offshore lead	Onsite and offshore lead with development team	Onsite and offshore lead

chitecture chiefly written in Microsoft's C# and ASP. Cycle times for these projects varied from three to nine months. The processes we used to develop these projects adhered to a CMMI Level 2 framework.

Our model

Our model follows the implementation model Hans Nissen suggested.[4] In Nissen's model, the client retains key requirements and design functions, and the vendor carries out detailed implementation, supervised by onsite leads, within a defined framework. Our model requires retaining an onsite team lead for each offshore initiative throughout the project life cycle. We treat the offshore team as an extension of the development team, not as a replacement.

Key success factors underlying our model include

- careful setup and planning,
- knowledge transfer and training,
- using a proven Web delivery foundation (WDF),
- establishing policies and procedures, and
- focusing on communication and checkpoints.

We follow a modified waterfall approach in which we implement the concept, analysis, and design phases primarily onsite. Construction and testing are primarily offshore. Table 2 documents the resource breakdown and responsibilities, indicating the staffing levels we typically deploy across the life cycle along with the key deliverables and who's responsible for each.

Early in the concept phase, we identify the onshore and offshore technical leads, who will guide the project from the concept phase through completion. We bring the offshore lead onsite to participate in the concept, analysis, and design phases as suggested in the literature.[5,6] We handle any necessary knowledge transfer as a planned activity with clear deliverables. The onsite and offshore leads jointly sign off on the estimates and resource planning. During construction, the offshore team is primarily responsible for artifact development (code and tests) and delivery in accordance with defined coding standards and best practices.

The onsite team developed our core WDF early on, comprising infrastructure code, coding standards, best practices, and value-added tools. The WDF established the technical foundation to support offshore development; we explain it in more detail later.

As each project matures, we ramp up the remaining offshore team as we get into the detailed design and construction phases. We also use an established set of process procedures, a standard estimation model, and a resource-planning sheet to forecast staffing needs and plan specific responsibilities.

On average, our current onshore and offshore teams are distributed in this manner. The onsite CLS (Corporate Legal Services Division) team makes up 40 percent of our staff and includes

- the business stakeholder,
- the project manager,
- the technical manager,
- the technical project lead,
- the business analyst, and
- the quality assurance lead.

Our offshore partner, comprising 60 percent of our staff, includes

- the technical project lead,
- the development team,
- the QA lead, and
- the QA team.

Figure 1 shows how we divide tasks between the onsite team and offshore partner during a project's life cycle. Offshore participation in the project increases during the project's later phases. CLS maintains onsite oversight and control at all times.

Key management guidelines

Our model comprises management and engineering guidelines that complement each other to provide a comprehensive offshore development and management process. This process starts with management decisions, including determining which projects we can send offshore, then leads into planning and communication.

Is the project offshorable?

Not all projects are well suited to an offshore model. To determine whether to utilize offshore development, we consider several key project characteristics, including[7]

- the business process,
- interaction requirements,
- complexity,
- current cost,
- control requirements, and
- risk of failure.

In particular, we've found that new, medium-sized development projects on existing frameworks are best suited for an offshore approach. We also consider whether the project involves other enterprise applications or state agencies that might not be available offshore. In our business, the applications interact with government agencies and other external organizations, and if we have significant touch points with these external bodies, our offshore strategy might vary. Unless we can replicate the entire infrastructure footprint offshore, we might discover numerous issues during the integration phase, making offshore development cost prohibitive.

Planning, policies, and procedures

Projects that are candidates for offshore development must be planned as such from inception. Our process documentation defines all deliverables required from both onshore and offshore teams. The service-level agreement clearly defines and specifies roles and responsibilities. In addition, suppliers are responsible for delivering code per our coding standards and guidelines.

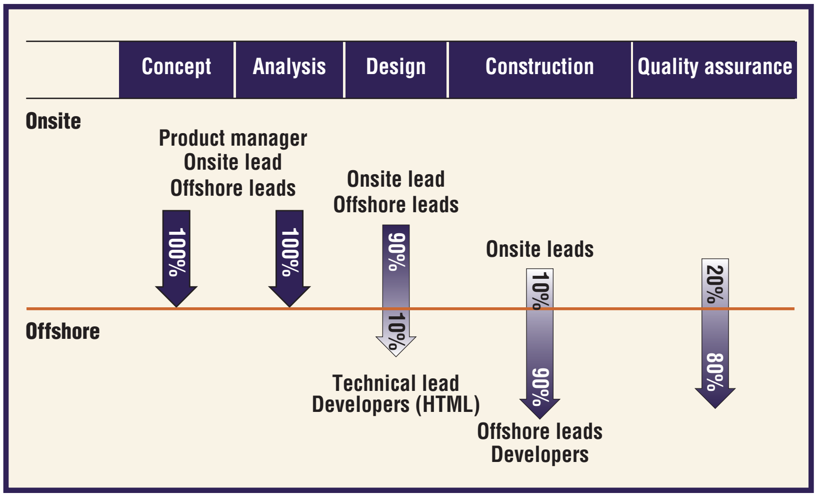

Figure 1. Onsite and offshore balance—the relative effort levels between the onsite and offshore teams throughout a project life cycle.

The onsite leads and technical managers sign off and baseline all functional and technical artifacts before we transition the project offshore. The offshore lead then moves back to the offshore site to conduct the knowledge transfer and oversee construction. The offshore lead is responsible for detailed resource planning and tracking during construction and for overseeing the offshore team.

Communication

Both structured and unstructured communications are vital to the work effort in a global setting.[8,9] To keep communications effective, we follow this set of principles:

- We maintain a direct line of communication between the onsite team and the offshore lead, retaining the closeness developed in the project's initial colocated phases.
- All onsite and offshore team members communicate issues via a tracking sheet that keeps track of the date of initiation, originator, assignee, description, classification, status, and more. Leads resolve most issues but move critical issues up to the management governance body as defined at the project's outset.
- We hold weekly meetings with the offshore team to monitor progress and discuss any open issues. These meetings also foster team spirit. We conduct them mainly by

conference call and more recently by videoconference.

- Depending on the project structure, a lead from the offshore team comes onsite to participate in the concept, analysis, and design phases. This lead typically returns offshore during the construction phase to oversee development and function as a liaison to communicate issues and status to onsite project stakeholders.

Setting up a framework for offshore delivery to ensure consistent, repeatable results is crucial for successful offshore development.

A microengineering process

In addition to management issues, our process covers engineering issues such as tools and the technical framework and infrastructure to support offshoring. Research has shown that achieving iterative and incremental development in distributed development is difficult.[10] To counteract these difficulties, our model incorporates design and code reviews, communication for fast iterations, immediate escalation and prioritization of issues, and frequent delivery.

Infrastructure and tools

Having the necessary infrastructure and tools to support a multicountry effort is critical, as Erran Carmel and Ritu Agarwal pointed out.[11] We've also found that advance planning and setup of supporting infrastructure and setting clear expectations of deliverables are critical to a successful relationship. In particular, we concentrate on

- physical connectivity,
- machine configuration and setup requirements,
- configuration management standards and guidelines, and
- defect-tracking standards and guidelines.

Because our customers and product lines require a high level of security and protection, we established a dedicated local area network at the partner site for all developers working on WK projects. Over time, we also established a wide area network between New York and our primary partner in India. We developed detailed specifications on configuration and software requirements for all developer PCs. We created templates and images that offshore developers used to replicate the onsite development environment offshore on a consistent and reproducible basis. This alleviated the potential for nonreproducible errors and ensured a stable environment for all developers.

At the outset we decided that all teams should work off a common source control repository. We clearly documented the configuration management process and standards. We followed similar processes to extend and document the defect-tracking repository to ensure its accessibility by all.

Our Web delivery framework

Brand-new infrastructure initiatives without an established foundation generally aren't good candidates for offshore development. Because many offshore engineers are new graduates, giving them free reign could result in sub-quality code. Setting up a framework for offshore delivery to ensure consistent, repeatable results is crucial for successful offshore development. This led us to create our WDF, which encapsulates the fundamental coding practices we want to follow in any project.

The WDF has four pillars:

- enterprise library and framework,
- coding standards and code review guidelines,
- value-added tools, and
- best practices.

Enterprise library and supporting standards. Establishing the enterprise library with corresponding coding standards and code review guidelines provides several benefits:

- *Code consistency*. The infrastructure code establishes a "sandbox model" of application building blocks with supporting coding guidelines to provide consistency within the development architecture. Such an approach helps improve the overall code base's consistency and reliability.
- *Organizational standards*. We developed well-defined policies around the build, deployment, and management of runtime configurations across multiple development, test, and production environments. The infrastructure layer, serving as the sandbox, helps enforce these policies and best practices.
- *Developer productivity*. Systematic ramp-up of all developers around the infrastructure layer, coding standards, and implementation best practices is critical to increasing

developer productivity, thus reducing development timelines.

Value-added tools and best practices. While the enterprise library code base and guidelines provide a foundation architecture, a set of supporting value-added tools and best practices ensure and validate application health on an ongoing basis. Key health indicators such as performance monitoring and error tracking must be "baked in" to the overall delivery to ensure the application's quality and maintainability. Addressing these needs led us to extend the WDF.

We developed value-added tools to capture key metrics such as page execution time starting from the development phase. The tools enabled developers to examine the components of page execution time (data access, business object load, and save) to tune slow pages early. We initiated corresponding best practices to ensure that all pages correspond to agreed-upon performance criteria before development shakeout.

We developed similar tools to monitor other key criteria such as SQL (Structured Query Language) execution time and exception reports. We introduced additional best practices to mandate the execution of all test cases as part of development shakeout. These tools let the onsite leads keep an eye on key criteria and maintain code quality and performance while working with large offshore teams.

Interim functional delivery

To clearly segregate onsite and offshore work, our model incorporates fine functional separation of tasks and responsibilities between onsite and offshore teams. We often faced numerous challenges, however, when attempting to integrate the code and then test end-to-end integration scenarios. Additionally, because the offshore team delivered functionalities close to code freeze, we frequently had little time to recover from integration issues. Teams would attempt to address these last-minute issues in panic mode, leading to patchy code.

To address these issues, we introduced *interim functional delivery* during construction. This milestone represents a form of incremental development. We learned to require that the offshore team deliver all main scenarios within the use cases in a functionally complete stage to the onsite team midway through the project. This enforces a practice of work allocation where all capabilities are functionally complete midway through the project as opposed to a subset of functionalities being fully complete while others haven't even been started. If the project has an eight-week construction cycle, the team delivers the IFD at the end of four weeks.

The IFD can be used in any environment, whether offshore or onsite. However, it's especially useful in an offshore delivery mode to ensure sufficient time for integration testing and review of code by the CLS onsite technical lead.

Figures 2 and 3 compare a normal delivery cycle with an IFD. As table 2 indicates, the IFD provides multiple benefits:

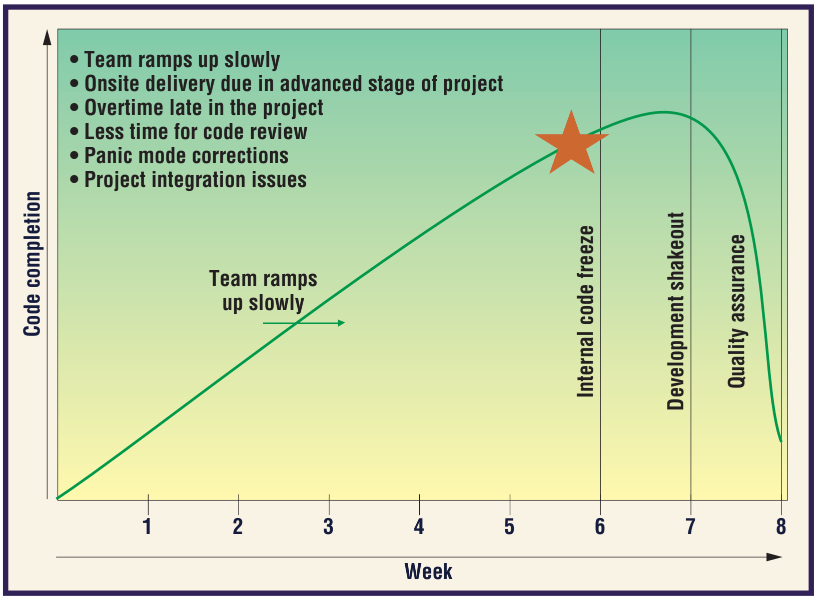

Figure 2. A standard delivery model for code completion.

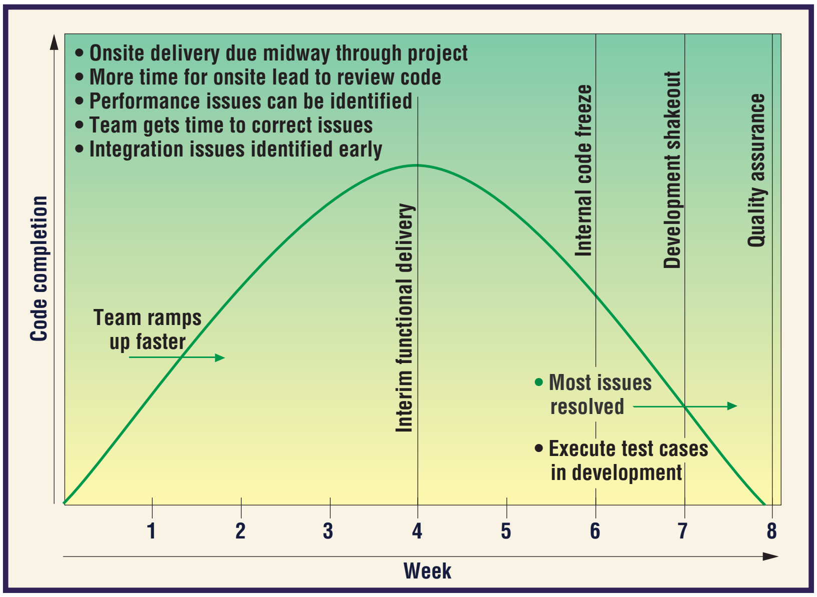

Figure 3. An interim functional delivery model for code completion.

Onshore and offshore vendor leads are critical to the project's success.

- The offshore team must ramp up faster.
- The onsite lead starts reviewing code midway through the project, providing sufficient time for corrections.
- We have sufficient time for end-to-end integration testing onsite.
- We have time to overcome the environmental and configuration challenges we encounter when deploying the new code onsite for the first time.
- The development team can make methodical rather than last-minute corrections.

Production support

The production support or maintenance work we've offshored follows a documented procedure as Werner Kobitzsch, Dieter Rombach, and Raimund L. Feldmann recommend.[12] The chief difference between new projects and PS arrangements is that knowledge transfer occurs once up-front to transition the product maintenance offshore. Depending on the product size, an onsite lead is retained at all times, working with the CLS onsite lead to act as a liaison with the offshore team to support one or more applications. Apart from emergencies, our PS process is managed as bundled releases with release cycles of approximately one to two months. We follow this procedure for PS bundles:

- A business representative reviews and signs off on the list of defects that the bundle will include.
- An onsite and offshore PS team jointly completes estimates for the bundle.
- The onsite and offshore PS team updates and signs off on analysis and design artifacts.
- The PS onsite lead sends defect fixes for construction offshore.
- Once the offshore team has finished construction and unit testing of the bundle, they send it to the onsite PS lead for review and verification.

Key criteria for managing PS include these:

- Because the PS team is typically smaller and less buffered than a project team, it typically includes only experienced offshore members who have worked on a project. The team stays the same throughout the year.
- We maintain an onsite PS lead to review the offshore team's code, which must adhere to coding standards.
- The onsite lead and technical stakeholders closely monitor and review the list of defects and enhancements entering into bundled releases to ensure that big-ticket items aren't allowed into the PS track.

Critical loose ends

In addition to the fundamental process steps, you must keep several things in mind when deploying an offshore approach.

Retain local domain experts

As the number of offshore initiatives increases, so does dependence on the offshore team. However, you shouldn't compromise the control and supervision of internal technical leads and domain knowledge. This requires keeping in-house leads in full control of project requirements and architecture.

Manage vendor experience levels

Expect entry-level talent to require some grooming. The principal offshore vendors are growing so fast that their experienced talent quickly moves up or out. So, these vendors use many junior people who need extra guidance to work effectively.

Select the leads carefully

Onshore and offshore vendor leads are critical to the project's success. Because the rank and file tends to be very junior, the lead ensures delivery. Leads must have strong communication skills to interface with the onshore team. We look for at least five years of experience, with at least two years in a lead capacity.

Forecast resource needs early

Getting good onsite staff from our vendors on short notice is becoming more and more difficult. You must plan for competing projects, resource shortages, and visa availability in advance so that you can get staff when you need them.

Things you must live with

Through our work with offshore teams, we've also gained some key insights into aspects of such projects that you must accept and manage. These things are part and parcel of any offshore initiative, and we do our best to accommodate them.

Higher documentation overhead

Projects for offshore implementation necessarily have higher documentation overhead. The concept team must produce detailed documentation to ensure clarity. CLS leads and management must sign off on all documents before transitioning offshore.

Locality

You can't walk into a developer's cube to get a status report or make a request, and you often must wait a full day to get a response. This means you must plan further ahead and anticipate problems better. The issue-tracking sheet is an important tool for dealing with this reality.

Higher management overhead

Close management is critical to offshore initiatives' success. Lack of clear assignments or monitoring leads to gaps in delivery. Detailed planning, weekly checkpoints, team meetings, regular review of issues, and status reports are necessary.

Environments

In our environment, setting up a replica of some legacy or enterprise packaged software infrastructure offshore has been difficult. Projects that have only partial environments replicated offshore have had significant integration problems once the code was bought back onsite. Teams must evaluate their environments for portability and might need to execute tactical projects to convert code bases before offshoring.

Quality

We've learned through experience that the client drives quality levels—you get what you ask for. This has made us more careful in specifying our needs. You must reiterate this on every release to ensure compliance. We recommend specifying coding standards in detail and enforcing them. Also, the customer (CLS) and vendor can agree upon technical measures such as transaction throughput and monitor them for achievement.

Cultural differences

Cultural differences play a role in running software projects between diverse locations. In our case, differences between American and Indian approaches often became apparent. In our early experience, we found that Indian engineers infrequently pushed back or reported problems. However, over time, the relationship has matured, and the Indian engineers developed the requisite domain knowledge and now take initiative and offer creative solutions. These findings mirror those of Geert Hofstede,[13,14] who characterized Indians as scoring lower than Americans in his individuality index. We mitigated this difference by bringing the offshore lead onsite to participate in activities early on in the project, exposing that person to our processes and requirements. We also probed carefully on status and progress in calls and maintained peer-to-peer dialogues between the vendor leads and the CLS engineers. This encouraged direct communication without requiring the channeling of all communication through an onsite vendor resource or project manager.

Effects on staff

Our model provides new opportunities for people with excellent communication and leadership skills, good architecture ability, responsibility, and a keen sense of how to leverage partner teams onsite or offshore. Junior developers and senior software engineers must actively develop the soft skills that are becoming crucial to their success. Managers must communicate with staff regarding future opportunities, to keep people engaged. In our experience, there have been job impacts following our move toward using offshore partners. Primarily, we've moved toward having more managers, project leads, and architects at the expense of more junior staff engineers.

Customer data privacy

Offshore development requires us to periodically transfer data offshore. Customers sometimes express concern over that data's security. We require all offshore vendors and employees on the team to sign a confidentiality agreement with WK. Additionally, to ensure confidentiality, we mask any private data with a set of prepackaged scripts before sending it offshore.

Risks

Offshore development has some risks. One is reduced productivity due to distributed team locations. Stephanie Teasley and her colleagues reported that in colocated teams, productivity

We've learned through experience that the client drives quality levels—you get what you ask for.

Logistical issues can slow down a project on a tight schedule if you don't adequately plan for them.

and job satisfaction are much higher.[15] Dale Karolak described common risks for global software development projects, including decreased morale, loss of face-to-face interaction, and a lack of trust between teams.[16] We've been largely able to avoid these risks by colocating team leads with the onshore teams and circulating them to the offshore location periodically to provide a human bridge.

From a sustained-management perspective, it has proven much more difficult to communicate changes and make sure everyone understands the modified approach and requirements with offshore teams. Over time, we also face the risk of domain knowledge and expertise diminishing onsite because most of the construction occurs offshore. We mitigate this by ensuring that a CLS staffer is involved in all key projects and works hands-on to fix defects as necessary.

Collaborating with vendors

Our offshore partners bring a variety of strengths to each project. First, they're eager for the business, and the staff is willing to work long hours. They're usually smart and routinely get the job done to match specifications. We count on them, and they're very reliable.

On the flip side, we've found some drawbacks, such as those we mentioned in the "Cultural differences" section. These tendencies force us to do more documentation than if the work were onsite. Also, infrastructure can pose problems and requires detailed planning for data sharing, intranet access, and so on. These are primarily logistical issues, but they can slow down a project on a tight schedule if you don't adequately plan for them.

Results

Over the last three years, we've made over two dozen releases using our offshore model on three major platforms. While we don't have shareable metrics on performance, empirical evidence shows that these releases were all delivered on or near their estimated release dates, with schedule variance of less than five percent. Additionally, the releases' defect counts have been similar to what we encountered with onsite initiatives. Business and customer satisfaction for these releases have been high; we've been able to deliver more releases in less time than we could have with onsite personnel only. As a further testament to the model's success, it has been extended to other WK subsidiaries in the last two years.

Our model allows for a truly global team. Today, companies like ours find talent all over the world. The technical and intellectual infrastructure required to compete are relatively low cost and are transportable. As John Berry put it, "Everything that can move down a wire is up for grabs."[3]

In the onsite world, we're moving toward a model that has a light developer core and a heavy project lead and architecture layer, with development occurring offshore. The onsite staff must think at a system-engineering level, and offshoring lets engineers focus on end-to-end problems if they're prepared to make that leap.

The offshore team needs strong management and improved architectural skills to design for completeness, modularity, and clarity. In this model, standardizing and communicating procedures are key, especially for consistency in requests and follow-up. Finally, awareness of cultural issues and clear expectations are paramount for all sides. In this way, offshore collaboration models such as ours can be even more successful in the future.

References

1. R. Hira, "Testimony to the US-China Economic Security Review Commission on Offshoring of Software & High Technology Jobs," IEEE-USA, 2005; www.ieeeusa.org/policy/POLICY/2005/021305.pdf.
2. H. Lackow, "Outsourcing Trends and Best Practices," 2005; www.cio.com/research/outsourcing/edit/trends/sld001.html.
3. J. Berry, *Offshoring Opportunities: Strategies and Tactics for Global Competitiveness*, John Wiley & Sons, 2006.
4. H. Nissen, "Designing the Inter-Organizational Software Engineering Cooperation: An Experience Report," *Proc. ICSE 3rd Int'l Workshop Global Software Development*, IEEE CS Press, 2004, pp. 24–28.
5. "Criteria for Making an Appropriate Outsourcing Selection: Insourcing, Nearshore, and Outsourcing," OAO Technology Solutions, 2005; www.oaot.com/downloads/about/library/whitepapers/Part%201%20Outsourcing%20Selection%20Criteria%20Insourcing,%20Nearshore%20and%20Offshore.pdf.
6. M. Fowler, "Using an Agile Software Process with Offshore Development," 2004, www.martinfowler.com/articles/agileOffshore.html.
7. J. Herbsleb and R. Grinter, "Architectures, Coordination, and Distance: Conway's Law and Beyond," *IEEE Software*, vol. 16, no. 5, 1999, pp. 63–70.
8. E. Carmel, *Global Software Teams: Collaborating across Borders and Time Zones*, Prentice Hall, 1999.
9. Z. Yan, "Efficient Maintenance Support in Offshore Software Development: A Case Study on a Global E-

Commerce Project," *Proc. ICSE 3rd Int'l Workshop Global Software Development*, IEEE CS Press, 2004, pp. 12–18; http://gsd2004.uvic.ca/docs/proceedings.pdf.

10. M. Paasivaara and C. Lassenius, "Using Interactive & Incremental Processes in Global Software Development," *Proc. ICSE 3rd Int'l Workshop Global Software Development*, IEEE CS Press, 2004, pp. 42–48.
11. E. Carmel and R. Agarwal, "Tactical Approaches for Alleviating Distance in Global Software Development," *IEEE Software*, vol. 18, no. 2, 2001, pp. 22–29.
12. W. Kobitzsch, D. Rombach, and R.L. Feldmann, "Outsourcing in India," *IEEE Software*, vol. 18, no. 2, 2001, pp. 78–86.
13. G. Hofstede, "Cultural Dimensions," www.geert-hofstede.com.
14. G. Hofstede et al., *Cultures and Organizations: Software of the Mind*, 2nd ed., McGraw-Hill, 2004.
15. S.D. Teasley et al., "Rapid Software Development through Team Collocation," *IEEE Trans. Software Eng.*, vol. 28, no. 7, 2002, pp. 671–683.
16. D. Karolak, *Global Software Development: Managing Virtual Teams and Environments*, IEEE CS Press, 1998.

About the Authors

James Cusick is an assistant director at Wolters Kluwer Corporate Legal Services. His research interests include software engineering processes and education, performance engineering, and enterprise architecture. He received his BA in organizational psychology from the University of California at Santa Barbara and his Certificate in Technology and Applications in software development from Columbia University. He's a member of the IEEE and the Project Management Institute. Contact him at Wolters Kluwer Corporate Legal Services, 111 Eighth Ave., New York, NY 10011; j.cusick@computer.org.

Alpana Prasad is an assistant director at Wolters Kluwer Corporate Legal Services, where she's responsible for technical oversight and management of multiple offshore initiatives. She also works on other corporate initiatives and programs to improve platform and product development standards and best practices. Her research interests include analysis of new and emerging technologies, primarily in the Microsoft domain. She received her master's in business administration from the Indian Institute of Management, Lucknow. She's a member of the IEEE. Contact her at Wolters Kluwer Corporate Legal Services, 111 Eighth Ave., New York, NY 10011; alpana.prasad@wolterskluwer.com.

ADVERTISER INDEX SEPTEMBER/OCTOBER 2006

Advertiser	Page Number
IEEE Equity 2006	**16**
John Wiley & Sons	**Cover 2**
OMSE ONLINE	**9**
Seapine Software	**Cover 4**
Software Test & Performance Conference	**Cover 3**
Classified Advertisement	**13**

Advertising Personnel

Marion Delaney
IEEE Media, Advertising Director
Phone: +1 415 863 4717
Email: md.ieeemedia@ieee.org

Marian Anderson
Advertising Coordinator
Phone: +1 714 821 8380
Fax: +1 714 821 4010
Email: manderson@computer.org

Sandy Brown
IEEE Computer Society,
Business Development Manager
Phone: +1 714 821 8380
Fax: +1 714 821 4010
Email: sb.ieeemedia@ieee.org

Advertising Sales Representatives

Mid Atlantic (product/recruitment)
Dawn Becker
Phone: +1 732 772 0160
Fax: +1 732 772 0164
Email: db.ieeemedia@ieee.org

New England (product)
Jody Estabrook
Phone: +1 978 244 0192
Fax: +1 978 244 0103
Email: je.ieeemedia@ieee.org

New England (recruitment)
John Restchack
Phone: +1 212 419 7578
Fax: +1 212 419 7589
Email: j.restchack@ieee.org

Connecticut (product)
Stan Greenfield
Phone: +1 203 938 2418
Fax: +1 203 938 3211
Email: greenco@optonline.net

Southwest (product)
Steve Loerch
Phone: +1 847 498 4520
Fax: +1 847 498 5911
Email: steve@didierandbroderick.com

Northwest (product)
Peter D. Scott
Phone: +1 415 421-7950
Fax: +1 415 398-4156
Email: peterd@pscottassoc.com

Southern CA (product)
Marshall Rubin
Phone: +1 818 888 2407
Fax: +1 818 888 4907
Email: mr.ieeemedia@ieee.org

Northwest/Southern CA (recruitment)
Tim Matteson
Phone: +1 310 836 4064
Fax: +1 310 836 4067
Email: tm.ieeemedia@ieee.org

Midwest (product)
Dave Jones
Phone: +1 708 442 5633
Fax: +1 708 442 7620
Email: dj.ieeemedia@ieee.org
Will Hamilton
Phone: +1 269 381 2156
Fax: +1 269 381 2556
Email: wh.ieeemedia@ieee.org
Joe DiNardo
Phone: +1 440 248 2456
Fax: +1 440 248 2594
Email: jd.ieeemedia@ieee.org

Southeast (recruitment)
Thomas M. Flynn
Phone: +1 770 645 2944
Fax: +1 770 993 4423
Email: flynntom@mindspring.com

Midwest/Southwest (recruitment)
Darcy Giovingo
Phone: +1 847 498-4520
Fax: +1 847 498-5911
Email: dg.ieeemedia@ieee.org

Southeast (product)
Bill Holland
Phone: +1 770 435 6549
Fax: +1 770 435 0243
Email: hollandwfh@yahoo.com

Japan
Tim Matteson
Phone: +1 310 836 4064
Fax: +1 310 836 4067
Email: tm.ieeemedia@ieee.org

Europe (product)
Hilary Turnbull
Phone: +44 1875 825700
Fax: +44 1875 825701
Email: impress@impressmedia.com