

December 2009

Global Delivery  
White Paper:



# Seven Pillars of Success for Global Software Delivery Adoption

Tim Blackmon  
Executive VP, Global Delivery

**COMFRAME**  
precision software innovation

## Contents

Software Delivery Results .....	3
The State of Software Development.....	3
Why Current Offshore Models Do Not Work .....	3
Pillars of Global Software Success .....	3
Establish Effective Communications .....	4
Maintain Software Quality .....	5
Deliver Software Accuracy.....	6
Ensure Team Unity .....	7
Manage Internal Perceptions.....	8
Manage External Perceptions .....	8
Establish a Successful Adoption Plan .....	8
Solution Summary .....	9
Choosing a Global Software Delivery Partner.....	10
About the Author .....	11
About ComFrame .....	11

## Software Delivery Results

*Exceed customer expectations while reducing development costs by adopting an integrated team approach and shared delivery responsibilities*

### THE STATE OF SOFTWARE DEVELOPMENT

---

As information technology professionals work to reshape the software delivery landscape during an economic recovery, the ability to do **“more for less yesterday”** has become a desirable competency for companies across all industry sectors. Organizations that survive and thrive will be those that can successfully integrate the use of offshore resources into the software engineering process as standard practice rather than experimental dabbling. Cost, however, will not be the sole driver of this trend. Markets will continue to demand faster delivery, better functionality and higher quality than previous generations. Effective use of the global software labor pool will be essential for meeting these demands and achieving competitive advantage.

### WHY CURRENT OFFSHORE MODELS DO NOT WORK

---

Offshore development gained a foothold in the early part of this decade. The BRIC countries (Brazil, Russia, India and China) spawned offshore development through extensive investment in information technology education and infrastructure. The major problem with many of the traditional offshore models, however, is that they tend to separate responsibility of U.S. and offshore teams with respect to software engineering disciplines, including requirements, design, development and testing. This segmentation consistently produces less than desirable results and significantly slows the adoption process for companies that want to tap into the global labor pool. Successful models are those that emphasize integrated teams with shared delivery responsibilities.

This white paper focuses on common challenges and mitigation practices companies must consider in planning for and adopting a global delivery approach to software development.

## Pillars of Global Software Success

Over the past decade of using offshore resources, a set of common challenges has emerged for the software industry. These challenges, including *lack of communication, poor software quality, missed customer expectations, lack of team unity, misperceptions* and *low user adoption*, hinder full realization and inhibit efficiencies of a global delivery model.

There are seven key areas that must be addressed by any company looking to establish and successfully execute a global software delivery strategy. These “pillars of success” serve as a framework for overcoming each of the challenges listed above and provide the foundational elements for a competitive, future-ready software development program:

- ➔ Establish effective communications
- ➔ Maintain software quality
- ➔ Deliver software accuracy
- ➔ Ensure team unity
- ➔ Manage internal perceptions
- ➔ Manage external perceptions
- ➔ Establish a successful adoption plan

## ESTABLISH EFFECTIVE COMMUNICATIONS

---

When software projects fail, technology is rarely the culprit. Most challenges in a software development effort are centered on team communications. The challenge is amplified when offshore resources are utilized, due to factors that include physical separation, time-zone differences, unawareness of business culture, language barriers, etc. In some cases, formal communication is the best solution. In other situations, an informal approach may work best. The bottom-line, however, is that lack of effective communications will result in failure 100% of the time.

### COMMUNICATION CONSIDERATIONS

- 1) Institutionalize a methodology:
  - a) Select a **prescriptive methodology**<sup>1</sup> – This will establish a framework for lines of communication and content. Make sure the development process is iterative and interactive.
  - b) Modify the methodology to suit the project’s needs – Don’t be afraid to borrow activities from other prescriptive methodologies if you think the project would benefit.
  - c) Utilize **feature-driven**<sup>2</sup> principles to assist in team building.

---

<sup>1</sup> A **prescriptive methodology** is one that is well defined in the industry and typically there are books, training, and many materials covering details of the processes of the methodology. Examples include AgileUP, USDP, Scrum, Adaptive, Crystal, and eXtreme Programming.

<sup>2</sup> **Feature-driven development** (FDD) is an iterative and incremental process for developing software. The main reason for its consideration in an offshore model is to build team cohesion across regional boundaries.

- 2) Globalize project management:
  - a) Project management becomes a part of every role – Train your onshore staff on basic project management principles. Choose project managers with multiple skill sets (i.e., business analysis, technology, etc.)
  - b) Create and execute a communication plan – Include daily, weekly, and monthly communications. Embrace the use of collaboration technology (WIKIs, blogs, professional network sites). Improve the communication processes with each iteration.
- 3) Establish KPI goals for development – Measure and post results for team consumption.
- 4) Utilize a change management tool that links software changes to work items.
- 5) Meet face-to-face whenever possible.

## MAINTAIN SOFTWARE QUALITY

---

In the early days of offshore development, quality was often an afterthought of the delivery process. In today's economy, however, companies cannot afford the side-effects of a low-quality software application. Poor quality can result in extended timelines, increased costs and reduced customer loyalty. To avoid the afterthought problem, quality must be built into the development process.

### QUALITY CONSIDERATIONS

- 1) The areas of quality to address during a development effort are the same for onshore and offshore. They are:
  - a) **Functional Quality** – How well does the application function in accordance with the specifications?
  - b) **Performance Quality** – Is the performance of the application adequate for the business?
  - c) **Maintenance Quality** – Is the application easily maintained?
  - d) **Compliance Quality** – Does the application adhere to certain regulatory constraints?
- 2) Perform unit, integration and system testing to address *Functional Quality*.
  - a) Unit testing should be incorporated into a continuous build process – If your tools allow, include **branch coverage**<sup>3</sup> goals and measure each build.
  - b) System testing and integration testing should begin after the initial iteration – Do not wait until the end of the project. Each iteration of the development should produce software that is testable.

---

<sup>3</sup> **Branch coverage** is a term that describes the source code coverage by a test. Establishing a target percentage encourages more complete unit testing from the development team.

- 3) Utilize **deep prototypes**<sup>4</sup> to address *Performance Quality* and *Maintenance Quality*.
  - a) Deep prototypes are created before the main development effort starts. They ensure the candidate architecture is one that is scalable, reliable, and can handle the estimated throughput.
  - b) Deep prototypes provide patterns for others to follow. These patterns serve as guidelines and produce more consistent and more maintainable source code.
- 4) For *Compliance Quality*, begin preliminary compliance checks several iterations before final delivery. This will give the team time to address issues early.
- 5) Behavior is modified through what is measured. Be sure to establish simple and easily attainable KPIs for each area of quality.

## DELIVER SOFTWARE ACCURACY

---

At ComFrame, we have a saying: “We do not strive to deliver the needs of the business at the beginning of a project. We will deliver the needs of the business at the end of a project.” Change is inevitable in almost any project you will undertake. Business needs or the interpretation of those needs are almost certain to change during a project. Many offshore software development initiatives have failed because a working piece of software was delivered that did not address the business need at the end of the project. How you embrace or resist change will determine your success or failure.

### ACCURACY CONSIDERATIONS

- 1) Use a highly iterative methodology that embraces change. With an iterative approach, deliveries are made often. This enables software to be reviewed and modified prior to product or application launch.
- 2) Establish a hierarchical arrangement of functional needs. Consider utilizing use cases or user stories. This will allow team leaders to segment and disperse functional requirements more efficiently. Be aware, however, that this does not imply the use of ‘heavy’ documentation, but the need to be consistent and hierarchical.
- 3) Produce user interface (UI) prototypes or wireframes – These will convey much more information than a document. Also, you can review these with the user community and amend before construction begins.

---

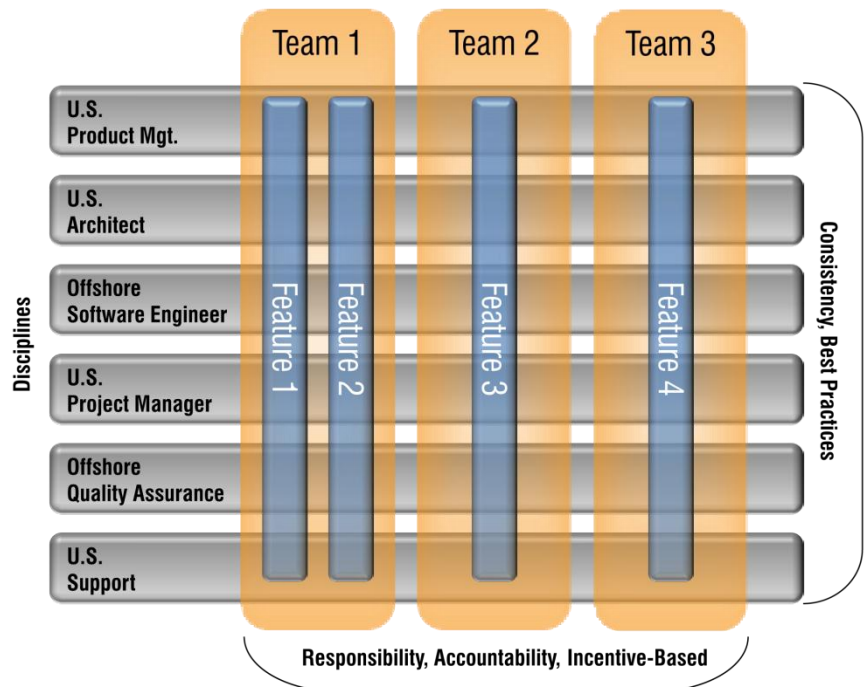
<sup>4</sup> A **deep prototype** is an architectural prototype representing a thin vertical slice of functionality through all tiers of a solution.

## ENSURE TEAM UNITY

Regardless of the mix of team member roles, team unification is critical to the success of any software delivery initiative. With many projects, an “us vs. them” mentality emerges. This typically occurs when delivery responsibility is segmented by geographic location or when a “throw it over the wall” process has been established. Regardless of how a project has reached this point, a less than desirable result is likely to be the outcome.

### UNITY CONSIDERATIONS

- 1) Establish cross-functional and cross-regional teams with common goals – The methodology should follow feature-driven principles and organize the teams appropriately. The diagram below illustrates how teams should be organized around features, not roles.
- 2) Hold team-building activities that span regions – These can range from simple project tasks to cross-cultural educational events. The point is to allow the teams to learn about one another, build trust and open up communications.



## MANAGE INTERNAL PERCEPTIONS

---

Often when the term “offshore” is mentioned, the internal perception quickly becomes “Am I being outsourced?” This can cause an environment of uncertainty, angst, and eventually costly and unnecessary voluntary turn-over. Companies need to address this issue head on and stress that being part of a Global Delivery Model is a skill set enhancement opportunity and not an employee replacement initiative.

### INTERNAL PERCEPTION CONSIDERATIONS

- 1) Provide global delivery mentoring and training for technical onshore resources
- 2) Emphasize that being part of a global delivery team enhances a technology professional’s skill set, rather than diminishing it.

## MANAGE EXTERNAL PERCEPTIONS

---

Many organizations fear that utilization of offshore resources will create an external perception that the company is eliminating U.S. jobs. Industry studies have shown, however, that the U.S. does not currently produce enough software engineers to meet future demands.

### EXTERNAL PERCEPTION CONSIDERATIONS

- 1) Consider your clients – If public opinion outweighs the benefit of utilizing a global labor pool, then a this model may not work for your business.
- 2) Consider this fact – All of us have been part of a global delivery model. For most of the products you purchase today, a global labor pool was involved in the production of the product. U.S. jobs have consistently adapted to higher-skilled positions and the software industry is no exception.

## ESTABLISH A SUCCESSFUL ADOPTION PLAN

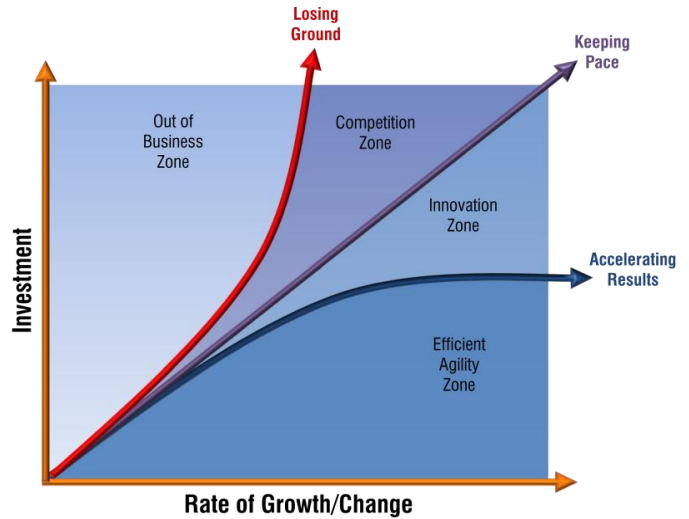
---

It can be difficult to know where to start with an adoption plan. The risk and complexity of a global delivery model can be daunting and overwhelming. Companies that have failed, have not utilized the experience of others, set unrealistic expectations, and/or tried too much too soon.

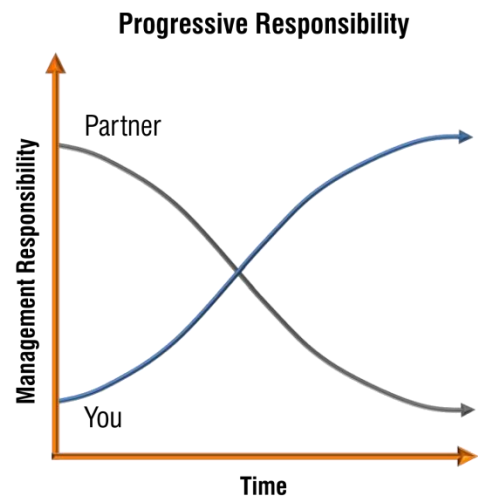
ADOPTION CONSIDERATIONS

1) Experience matters – Choose a partner that can assist you in navigating through the challenges.

2) Define success. Evaluate where you are and establish goals for the future. Are you losing ground, just keeping pace or accelerating your results? Be realistic. The chart at right depicts the various regions of the software delivery landscape where companies are positioned with respect to ongoing software development investment measured against rate of growth/change.



3) Leverage a partner for a strategic global delivery adoption – This will enable you to learn from someone that has already experienced all the pitfalls of global software development, while enabling you to gradually and incrementally take on increasing levels of management responsibility of offshore resources. The chart at right illustrates the benefits of a “progressive responsibility model” that allows for complete global delivery realization while mitigating initial adoption risks.



Solution Summary

Delivery of “more for less yesterday” rests squarely upon the shoulders of software development leaders. The use of a global labor pool is an essential component in the resolution of this marketplace demand. There are significant challenges and risks, however, which must be mitigated before success can be realized. Companies must establish a sound adoption plan and process that will address communication, quality, delivery accuracy, team unification, internal and external perceptions. The simplest and least risky path of adoption involves leveraging the experience of a global delivery partner. This approach will dramatically shorten the adoption cycle and reduce its total cost.

## Choosing a Global Software Delivery Partner

### WHY IS COMFRAME THE BEST OPTION TO ASSIST WITH YOUR GLOBAL STRATEGY?

---

ComFrame Software Corporation has the experience, personnel, adoption model and methodology necessary to help you successfully transition to a global software delivery approach for business and technology advantage.

#### NEARLY A DECADE OF EXPERIENCE

ComFrame initiated a global software delivery model in 2002 by partnering with a company that had software engineering resources in Bangalore, India. After four years of delivering software with this model, it became clear that ComFrame needed to own and control its own offshore development location. Early in 2007, ComFrame established an office in the Zhongguancun Software Park in Beijing, China. Throughout this time, ComFrame has continued to refine its global delivery approach, improving our development efficiency with the successful completion of each software project delivered for our clients.

#### TALENT LOYALTY

Our Beijing office is located in the heart of the leading academic and technology center of China, enabling ComFrame business leaders to select from an extremely rich pool of IT talent. The turnover rate in Beijing is lower than our U.S. offices, which is already below 10%. The general manager of our offshore office location is a long-time ComFrame employee, native to Beijing but educated in the United States.

#### PROGRESSIVE RESPONSIBILITY ADOPTION MODEL

ComFrame's progressive responsibility model enables our software development clients to learn as they go and progressively take on more and more of the management and control of software project teams based in Beijing.

#### BEST OF BREED METHODOLOGY

ComFrame's Praesideo methodology ensures precision and accuracy in our software delivery process by embracing change, ensuring appropriate communications, delivering iteratively and incorporating quality early in the development process.

## About the Author



Tim Blackmon is executive vice president of global delivery for ComFrame Software Corporation. Tim has spent the past 20 years building enterprise software applications in the healthcare, insurance and finance domains. Over the past seven years, he has served as principal consultant on a number of globally-delivered software projects that ranged in size from 100 to 100,000 hours. Consulting roles included senior vice president of engineering, vice president of software development, chief architect, architect and senior engineer. These roles established Tim's proficiency in a broad range of modern technologies, disciplined methodologies and process improvement approaches.

Tim's key project experience includes business requirements facilitation; workflow analysis; use-case analysis; technology staff recruitment, organization, training and mentoring; refactoring of software to new technology platforms; software configuration management; and software release management and delivery. Since coming to ComFrame in 1998, Tim has served in various executive leadership roles that include COO, CTO, vice president of solutions strategy and vice president of business partnerships and alliances. Tim has a BS in computer science with a minor in engineering psychology from the Georgia Institute of Technology in Atlanta, Georgia.

## About ComFrame

Since 1997, ComFrame Software Corporation has completed more than 300 elegant, large-scale software projects for clients across the southeastern and central United States, helping them reduce IT complexity and successfully deploy modern architectures and applications that improve business agility and efficiency. From offices in Birmingham, Nashville and Beijing, ComFrame delivers a broad range of custom solutions and packaged business applications. More than 110 architects, software engineers and project managers make up our highly skilled team. Expertise ranges from enterprise application development and integration to consulting, strategic business and brand management, business process and performance management, business intelligence and enterprise data management, information integration and collaboration management, project and portfolio management, IT mentoring and training in new technologies and service-oriented architectures. For more information about ComFrame, visit [www.comframe.com](http://www.comframe.com).

**Trademark information:** ComFrame, the ComFrame logo, and the ComFrame products listed in this document are trademarks or registered trademarks of ComFrame Software Corporation. All other trade names are trademarks or registered trademarks of their respective holders.