Optimizing Agile with Global Software Development and Delivery

A blueprint for integrating global delivery and Agile methodology, allowing organizations to achieve faster returns on investment, accelerate time-to-market and increase market opportunity.

Executive Summary

Agile software development and global service delivery have achieved demonstrable benefits for organizations worldwide. Although there are challenges with both strategies, and questions as to whether they can coexist, when managed properly Agile and global service delivery provide organizations with numerous benefits, including accelerated and bug-free software delivery that tightly aligns with business requirements. This white paper provides a process framework that was developed as part of our implementation of a multimillion-dollar project for a global consumer goods company delivered via a global, Agile delivery model. This paper not only examines the benefits that businesses can leverage by implementing this model but outlines the best practices that should be followed to handle challenges that are sure to crop up along the way.

Evolution of Concept

The software services industry has undergone numerous changes during the past decade. Among the visible changes in software development has been the increased use of Agile software methodology, which has emerged as a way to accelerate code creation and delivery while remaining aligned with dynamic business requirements.

Meanwhile, traditional global delivery has provided a unique selling point due to cost reductions and gains in productivity and resource management.

However, the increased adoption of the Agile software methodology has proven challenging for companies trying to seamlessly partner with software service providers. Software service providers were at first reluctant to embrace Agile since it was thought to offset the benefits of the global delivery model. However, increased adoption of Agile by organizations worldwide has forced clients to realign their relationships with service providers to incorporate the features of Agile development, such as rapid development and quick turnaround time, to fit the global delivery model.

Business Environment

Organizations are exhibiting increased levels of confidence in sourcing due to the success of the global IT and business services industry. This has led to an increase in the complexity of projects that are outsourced. Complex projects increase
the risk of failure; this has resulted in many services firms using structured development methods that call for greater customer involvement. This creates overhead for the project and reduces flexibility.

There is evidence that Agile methodology in outsourced projects can deliver better results in a dynamic environment. A survey conducted by Executive Brief, indicates that more than 56% of software industry business leaders preferred Agile software methodology over other models (see Figure 1).

Industry Challenges
Among the key obstacles to global services firms that want to embrace Agile is that it requires a single team working in one location; adapting this process to multiple teams in different geographies and time zones poses significant challenges. The process-oriented global delivery model requires the cumbersome and time-consuming creation of several artifacts, such as functional specification documents, technical specification documents, test case documents, requirement specifications, etc. The Agile model provides an alternative to overcome this challenge by providing rapid development with minimal documentation due to its enhanced interactivity among developers and clients.

Customer Expectations
Whether it’s software development or business process outsourcing, when two disparate organizations work together success can be achieved only iteratively, using as many feedback loops as possible that can be designed into the effort.

Global Agile software development offers both buyers and sellers of global business and IT services the opportunity to achieve perfection.

In a global services engagement, customers can partner with service providers to meet the increased need for rapid builds and dynamic business requirements, thereby providing:

- Increased visibility during the development process.
- Flexibility to alter requirements and incorporate change requests.
- Quick turnaround.

Our framework helps organizations embrace global Agile software development by leveraging proven global services delivery methods.

Key Components

- **Sprints:** A module/feature of a product will be developed using short development cycles called Sprints.

- **User stories:** A user-centric description of the goals that one or more people will achieve by using the end software deliverable. This includes delivery of a product with incremental features and improvements at the end of each Sprint and parallel execution of all phases of the software delivery lifecycle (SDLC) in a Sprint.

- **Stand-up meetings:** Daily Scrum calls for status updates and risk/issue resolution. Key decision-makers participate in the meeting and all issue clarifications/resolutions are provided in the same meeting.
Creating a Global Agile Development Plan

Agile Scrum-specific software configurable items (SCI) include:

- **Product backlog:** This is a prioritized list of all features needed and wanted for the end product.
- **Sprint backlog:** A subset of the product backlog that needs to be frozen before a Sprint starts.

Any change suggested in a Sprint will be added to the product backlog and will be part of the next Sprint.

### Prerequisites

Figure 2 illustrates the prerequisites needed to begin global Agile development.

### Roles and Responsibilities

One of the main criteria for a successful global Agile delivery model is the commitment from the organization and its end users to engage closely with service provider and carry out the responsibilities, as listed in Figure 3.

### Allocation of Responsibilities During a Sprint Development Project

<table>
<thead>
<tr>
<th>Stage</th>
<th>Details</th>
<th>Vendor</th>
<th>Client</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint scope and duration.</td>
<td>Sprint duration should be no more than three to four weeks.</td>
<td>Decide the schedule (three or four weeks) depending upon the requirement.</td>
<td>To confirm the scope at least one week before the start of the Sprint.</td>
<td>Sprint backlog.</td>
</tr>
<tr>
<td>Sprint requirement gathering.</td>
<td>For Sprint execution, it is mandatory to have clear requirements prior to the start of the Sprint.</td>
<td>Confirm the requirement’s clarity after requirement analysis.</td>
<td>To decide the start of Sprint and clarify all requirements.</td>
<td>Product backlog user stories.</td>
</tr>
<tr>
<td>Sprint execution.</td>
<td>Product backlog will be updated with new requirements.</td>
<td>Execute the Sprint.</td>
<td>Perform user testing.</td>
<td>UAT logs.</td>
</tr>
<tr>
<td>Sprint user testing.</td>
<td>To be done in the last week of the Sprint.</td>
<td>To plan bug fixing and changes suggested.</td>
<td>Perform user testing.</td>
<td>UAT logs.</td>
</tr>
<tr>
<td>Sprint daily stand-up meeting.</td>
<td>15 minute stand-up meeting.</td>
<td>Present quick status and plan for the day.</td>
<td>Clarification queries and discuss what to take up next if required.</td>
<td></td>
</tr>
<tr>
<td>Sprint review.</td>
<td>Sprint review meeting to be held daily to give status and receive clarifications if any.</td>
<td>Share learnings and issues.</td>
<td>Share learnings and issues.</td>
<td>Steps needed to modify the process.</td>
</tr>
<tr>
<td>Sprint sign-off.</td>
<td>Sprint sign-off as the exit criteria.</td>
<td>Provide sign-off on successful execution of each Sprint.</td>
<td>Sign-off e-mail.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2

Figure 3
Ready, Set, Sprint

**Methodology**

Project delivery is divided into small functional releases or increments to manage risk and to obtain early feedback from customers and end users (see Figure 4). These small releases are delivered on a schedule using iterations that typically last between one and four weeks each. Iterations are all fixed at the same length to maximize feedback and regularly force the trade-offs necessary to deliver. Moreover, their fixed scope helps retain stability. Plans, requirements, design, code and tests are created initially and updated incrementally as needed to adapt to project changes.

**Sprint Schedule**

The various stages of a representative Sprint are depicted in Figure 5.

- **Sprint 0**: This Sprint is the initial step to begin the Agile process. It consists of gathering requirements necessary to begin the first Sprint. A product backlog list is created to identify all the potential user stories. Sprint 1 will begin only after the scope is confirmed and all requirements are clarified.

- **Sprint 1**: In the example shown in Figure 5, each Sprint consists of three weeks of development followed by two weeks of user acceptance testing (UAT) and change requests (CR). Change requests raised by the client are classified into major and minor CRs. Minor CRs are incorporated into the current Sprint during the UAT week and major CRs are added to the product backlog. The product backlog is reviewed during the iteration planning process, and the scope for the upcoming Sprint is...
finalized based on the priority of the items in the product backlog.

- **Sprint 2**: Requirement gathering for Sprint 2 will begin during the development stage of Sprint 1. Development of Sprint 2 will begin in the second week of UAT of Sprint 1.

The same process will be repeated for subsequent Sprints.

The roles associated with each stage in the iterative process are illustrated in Figure 6.

### Testing

Maximizing the benefit of global Agile software development requires that testing and coding begin concurrently. Requirements are translated into user stories and then software features, against which test cases are built. Acceptance testing occurs in tandem with development; no development Sprint is considered complete until the software features pass QA.

Test-driven development (TDD) is a rapid cycle of testing, coding and refactoring. When adding a feature, the developer and tester may perform dozens of these cycles, implementing and refining the software in baby steps until there is nothing left to add or take away. A Microsoft research study shows that TDD substantially reduces the incidence of defects. When used properly, it also helps improve design, documents public interfaces and guards against future mistakes.

### Business Benefits

The benefits that were realized as part of our implementation of a multimillion-dollar project for a global consumer goods company delivered via a global Agile delivery model are as follows:

#### Financial

- **Market opportunity**: Building functional products with limited features and then continuing to add features results in accelerated time-to-market.

- **Faster return-on-investment**: By using a global Agile development model, products/applications can be built and released earlier than by using traditional Agile or waterfall models, resulting in faster ROI.

- **Higher productivity and reduced risk**: Clients are able to obtain higher productivity than with traditional Agile and waterfall models due to elimination of irrelevant paperwork, and they require fewer artifacts by leveraging the Agile global delivery model. The iterative process ensures that risks are minimized due to constant feedback on deliverables provided by the client organization.

---

**Figure 6**
Process

- **Continuous improvement**: Agile development accelerates the delivery of initial business value, and through a process of continuous planning and feedback is able to ensure that value is continually maximized throughout the development process.

- **Visibility**: By measuring and evaluating status, based on testing software, much more accurate visibility into the actual progress of projects is available for end user and development teams.

- **Collaboration**: Through daily communication, global Agile teams can best understand and discuss customer requirements. Team collaboration reinforces best practices and makes it easier to introduce new tools and development techniques.

Application

- **Lower defect density**: Customers are assured of getting defect-free applications due to the iterative nature of releases. Critical bugs are caught early in the project.

- **Adaptability**: As a result of iterative planning and feedback loops, global teams are able to continuously align the delivered software with desired business needs, easily adapting to changing requirements throughout the process.

- **Flexibility**: An Agile global delivery model is far more accommodating to change requests at any time of the project when compared to traditional Agile and waterfall models. Frequent communication eliminates gaps in requirement gathering.

The Best of Both Worlds

To succeed, global Agile software development model projects require the following:

- **An ideal project resource split**: 20% customer resources, 30% IT onsite resources and 50% IT global resources.

- **The teams should have overlapping work hours** and the entire team should start together.

- **An offshore representative** should be part of the daily Scrum.

- **Retention of business knowledge** by the client organization is key.

- **A variety of communication methods** should be used (rather than just one): videoconferencing, IM, LiveMeeting, Web collaboration tools, project portal repository, Wiki forums and face-to-face sessions.

- **Preplanning** for the next Sprint (three-week Scrum) should occur in parallel with the running Sprint.

Guidelines to consider include:

- **Well-defined acceptance criteria with production-like boundaries**: While unplanned work sometimes occurs, it lowers efficiency of completing planned work. Try, if at all possible, to defer the unplanned work to the next Sprint (a three-week delay).

- **Centralize documentation** so it is accessible for all and changeable by all.

- **Maintain daily stand-up meetings**.

- **Maintain metrics** of planned tasks vs. unplanned tasks completed per Sprint.

- **Ensure the requisite communication bandwidth**. This is often a challenge for global teams. Remember to schedule overlapping work hours.

- **Provide full visibility into project status**.

- **Work with mature clients only**: clients need to understand how Agile works.

- **Provide Agile mindset and method training**. Examples of training approaches include:
  > One-day workshops.
  > Computer-based training (CBT) of best practices.
  > Mentoring.

Footnotes

About the Author

Jithin Josey PMP®, CSCP is a Senior Consultant in Cognizant Business Consulting’s Engineering and Manufacturing Services Practice. He has seven-plus years of experience in project management, Agile software development methodology, business requirement gathering, business analysis and intelligence and business consulting around IT strategy, supply chain management and manufacturing and logistics across multiple geographies in the U.S., Canada, Singapore, UAE, Philippines, Malaysia and India for leading Fortune 500 clients. Jithin holds an M.B.A. from SP Jain Center of Management-Dubai/Singapore and a bachelor’s of engineering degree from Bangalore University-India. He can be reached at Jithin.Josey@cognizant.com.