



Flexibility in Software Development Methodologies: Needs and Benefits

Executive Summary

As a leading IT services organization, we have had several opportunities to design and develop software for Fortune 500 companies at the client site. Almost all the clients we worked with have used formal and mature procedures and tools for the development and deployment of software. These processes are normally oriented toward either sequential software development processes, such as the Waterfall model, or Agile software development processes, such as the Scrum model.

With rapid advances in communication and information technology, global development teams have become the norm, allowing organizations to deliver high-quality software and solutions at low cost. Leading software services companies have increasingly used a global software delivery model to provide custom information technology, consulting and business process outsourcing services to their various customers. For most software services companies, not having a global delivery strategy would put them at a serious competitive disadvantage.

This whitepaper conveys how companies can benefit from introducing flexibility into their software development methodologies. It provides an overview of our experience in executing a mid-size software development project for a large multinational corporation using a hybridized development framework. This framework incorporated the Waterfall and Scrum models for

different software modules of the same project, while utilizing geographically distributed teams. This allowed us to harness the benefits of both models while mitigating the risks associated with either of them.

Waterfall Model Overview

The Waterfall model is a highly structured, sequential software development process that progresses through various software design and development phases in linear order. An ordered list of the phases in the Waterfall process includes the following:

1. Requirements specification
2. Design
3. Construction (implementation or coding)
4. Integration
5. Testing and debugging (validation)
6. Installation
7. Maintenance

The model recommends total and correct completion and documentation of each phase before moving on to the next. As such, it emphasizes that requirements are frozen and the design phase is complete before proceeding to the coding phase. This saves time and effort in the construction, integration and testing phases of the software development process and reduces the risk of schedule slippage and cost overruns.

Progress on the project can be easily monitored by interested stakeholders because each phase has a defined start and end point. Significant customer involvement is required during the requirements specification, design and testing phases, while remaining phases do not require significant input from the customer. One major drawback associated with the Waterfall model is that it can be too late in the development cycle to discover business critical defects, as business testing only begins after the development phase.

Scrum Development Process Overview

Scrum has proved to be a highly successful framework for project management that takes an adaptive, iterative and incremental approach toward software and product development. Scrum is ideally suited for projects with changing requirements. In order to operate in such an environment, Scrum relies on a self-organizing, self-managing, cross-functional Scrum team that is supported by a Scrum Master and a product owner. The product owner represents the voice of the users and customers and is responsible for creating a prioritized wish list of product features, called a product backlog.

Scrum projects advance via a series of sprints, which are typically two to four weeks long. At the start of the sprint, team members commit to completing a portion of the product backlog.

This set of requirements in the backlog is frozen for the duration of the sprint. During each sprint, the Scrum team works on creating a potentially deliverable product increment. At the end of the sprint, the product owner and the customers review the deliverable and provide feedback on the build. This feedback could take the form of bugs, changes in functionality, etc., which could significantly alter the product backlog and the work done in the next sprint.

Although the Scrum methodology often requires significant customer involvement throughout the software development process, it can significantly increase productivity and reduce the time required to release quality software to the customer – when it is used to solve the right problems, by capable teams and with supportive management.

Project Description

Respirator Medical Analysis Software (RMAS) is an application that generates and maintains respirator evaluations for 40,000 workers around the world in a Fortune 100 product manufacturing conglomerate. It collects exhaustive health-related information from workers assigned to manufacturing facilities and stores this information in a database. Based on responses from the workers, the application uses a complex algorithm to approve the right respirator for the workers.

Scrum Process Overview

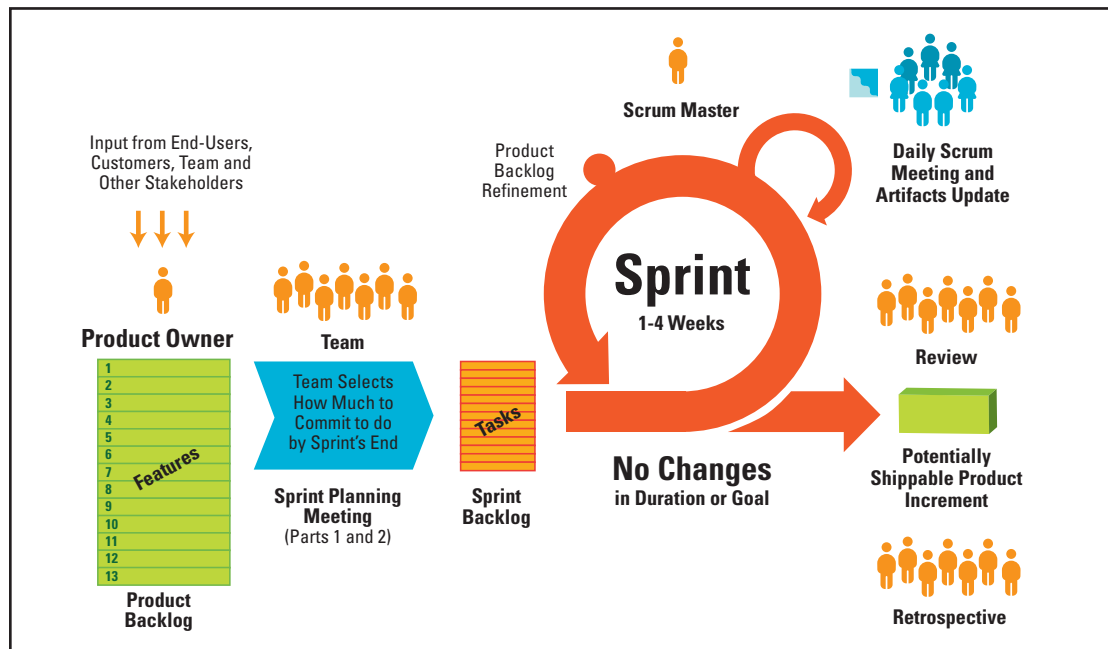


Figure 1

Functionality contained in the algorithm is based on government regulations and helps the client create a safe working environment in its manufacturing plants. It allows the client to manage risks associated with corporate liability, while embracing its role as a responsible corporate citizen.

The project has been developed as a three-tiered Web application on the Java Enterprise Edition platform, consisting of presentation, business and data layers. The application supports five different actors and contains 22 screens, with functionality that is based on requirements contained in 26 use cases.

The algorithm to evaluate the correct respirator for a worker formed the core of the business functionality and accounted for nine of the 26 test cases. Since the algorithm was based on regulations that had not changed for a number of years, requirements for the business tier were very well defined and fairly stable when the project kicked off.

The remaining 17 test cases dealt mostly with the presentation layer and covered complex functionalities involving the actors and user interface screens. The requirements for these 17 test cases were initially undefined by the client and were to be elaborated on as the project progressed.

Project Implementation

As project requirements and use cases were analyzed, we decided that software development of the business functionality would be best executed under a traditional Waterfall model, while using the Scrum model for the presentation layer would allow us to rapidly develop against emerging user interface requirements. A hybrid Waterfall and Scrum approach seemed to be a feasible and sensible option for a number of reasons:

- Requirements of the business and data functionality were stable and clearly defined.
- Clients saw greater business benefits in validating and testing the algorithm in its entirety, as opposed to validating incremental iterations of business functionality.
- Software design and development of the algorithm needed to be precisely documented and delivered to the client.
- It was possible to deliver the business functionality to the client independent of a produc-

tion-level user interface, thus allowing error detection and bug fixing before completing the development of the presentation layer. This flexibility somewhat mitigated the risk of delivering incomplete software at the end of development.

- Requirements pertaining to the presentation layer were going to be developed as the project progressed.
- Clients aimed to lessen the burden of data collection on workers and needed to see working prototypes of several user interface options before deciding on the most user-friendly and least time-consuming option.

Implementation of the Business and Data Layers

Besides the stability of client requirements, the success of a project or module executed by a global team using the Waterfall model depends on the offshore development team's understanding of the requirements. To ensure all stakeholders were on the same page, a Cognizant associate was assigned to work on-site at the client location as a business and IT analyst.

The primary responsibility of the analyst was to work closely with client business users to develop and refine use cases for the business module. This activity was scheduled to be completed in three weeks. The use cases were loaded into Rational RequisitePro to ease management and versioning. The relationships between the use cases and the high-level requirements were maintained through a requirements traceability matrix (RTM).

Once the high-level requirements were framed and the use cases were written, the use cases were sent to the development team for analysis, clarification and validation. Although it was estimated to be a weeklong activity, clarifications were sought until design and estimates were complete.

As a next step, the effort to develop the core business functionality was analyzed and the estimates were created using the Use Case Point Estimation methodology. Both the development team and the IT analyst were involved at this

We decided the business functionality would be best executed under the Waterfall model, while the Scrum model was best for the presentation layer.

stage. The estimation formulation and review were completed in two weeks. The estimates were reviewed with clients, and client approval was obtained. Once the estimates were approved, the requirements were frozen.

Following approvals, the analyst then worked with the offshore team lead and senior developers to develop the software design. The design phase included both the database design, as well as the architecture design. The design specifications and design diagrams were documented in a Software Architecture Document (SAD) template and were reviewed by the IT analyst. It was then reviewed by a client architect, and the design was approved. The design process was completed in three weeks.

Based on the final approved design, the development team started to develop the business module. The application development was completed in 2.5 months, including unit testing of different pieces of the functionality and peer testing within the team.

The business module was then delivered for quality assurance to the testing team. Testing, including bug fixing, was completed in two weeks, and the final build was delivered to the client for user acceptance testing and business scenario testing. Any changes to the functionality or requirements were logged and implemented, using the complete Waterfall model template in a new, smaller iteration.

Implementation of Scrum

For a Scrum project to succeed, it is crucial to foster a collaborative, self-organizing and self-managing environment within the team. Development of trust between team members is a primary ingredient for the creation of such an environment. It is, therefore, essential for teammates to communicate as openly and regularly as required.

In a geographically distributed team, it is often necessary to spend time collaborating with team members outside of core work hours in order to bridge the time difference between onsite and offshore locations. We, therefore, set team member expectations accordingly before the start of the project and were fortunate to have a highly skilled and enthusiastic team willing to dedicate

a portion of its non-work hours to brainstorming and consulting with each other as needed. We also provided team members with virtual private network (VPN) and remote desktop access so they could connect to their work machines without needing to be in the office. Phone and videoconferencing facilities were provided to enable interactive dialogue and the perception of nonverbal communication signals.

For the purposes of UI development, the onsite analyst assumed the role of product owner. Given his proximity to client management and his involvement in use case and requirements development, this was a logical choice. The offshore team was made up of the Scrum Master and the actual development team.

The requirements that were generated in the backlog were developed across different sprints. Each sprint had a timeline of one month, with two sprints executed by two different Scrum teams in parallel. The sprints executed in parallel consisted of requirements that were independent modules and hence did not require integration at the end of the month. The UI module was completed in a two-month timeframe, utilizing four different sprints.

Each sprint was executed in the following manner:

- To ensure that project business requirements were clearly communicated and understood by the offshore team, the product owner and the offshore team worked together on converting the business requirements of the prioritized product backlog into IT requirements.
- The first four days after the sprint planning meeting within every month-long sprint was devoted to this process, enabling the offshore team and the onsite analyst to be on the same page.
- In addition to brainstorming sessions, daily 20-minute standup video/teleconference calls were scheduled to discuss what work was completed, what remained and to discuss any potential problems, as well as provide clarifications on requirements and requested features. An important tool was the creation of meeting minutes.
- A shared Excel spreadsheet was used to exchange queries and clarifications on requirements in the backlog between the Scrum team and Scrum product owner. While

For a Scrum project to succeed, it is crucial to foster a collaborative, self-organizing and self-managing environment within the team.

A Hybrid Development Framework

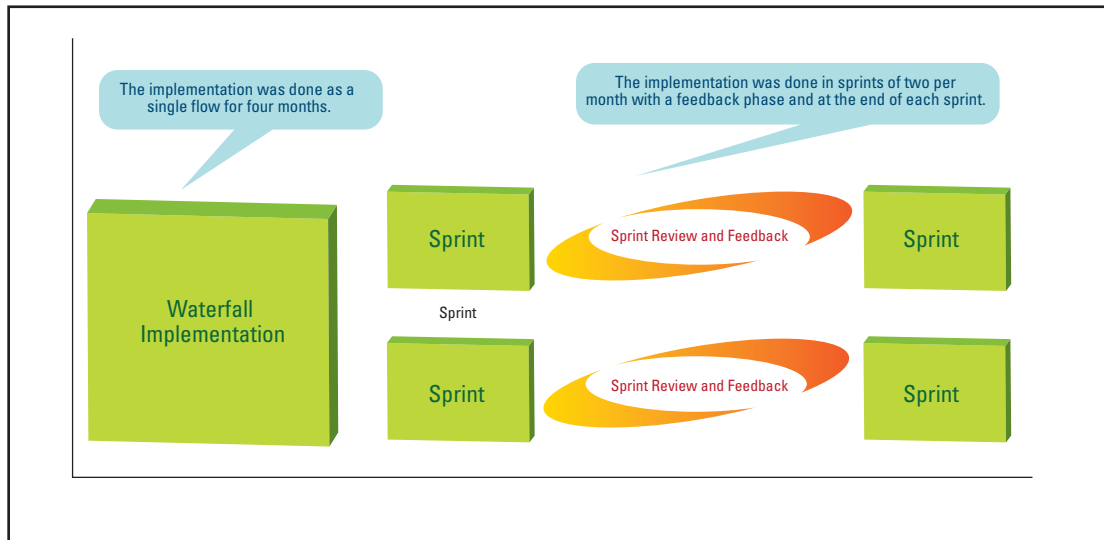


Figure 2

it can be argued that Agile development is less about creating documentation and more about development of working software, it is important in a global context to document and validate the understanding of the entire team. During development, pair programming was used to effectively share responsibilities in the project. Instead of assigning delivery and unit testing of an entire requirement to one team member, the responsibility is shared by two team members, with development and unit testing occurring in tandem.

- The last week of each sprint was devoted to peer testing and bug fixing.
- A sprint review meeting was held after each sprint to gather feedback from the product owner and the clients. Any changes suggested during this meeting were added to the product backlog and prioritized by the product owner for the next sprint.

Conclusion

We successfully developed and tested the presentation and business modules within six months, as per client requirements, and delivered an integrated application for user acceptance testing. Deployment was scheduled after completion of UAT and a few final bug fixes. The client considered the deployment a great success, and we heard a lot of positive feedback from the RMAS user

base and the client project manager. Thus, the hybridized software development approach was a novel and highly effective solution, used to guide a challenging project to an extremely favorable outcome.

Having a focused and expert IT team that was comfortable with different software development models – as well as taking an open and flexible approach toward the software development process – were the key ingredients to speeding up development time and minimizing risks. At the same time, the global software development team allowed us to keep costs low and helped our client meet budget targets.

While a hybrid software development process may not be a silver bullet for all software development problems, it could be detrimental for large organizations to be biased toward a single software development methodology to execute all projects. The choice between Agile, Waterfall or any other software development methodology should be based on which model best reduces risk, increases productivity and improves quality while achieving the goals of the project.

We believe that the freedom afforded to software architects, analysts or developers to tailor the software development process according to business needs and project characteristics is a crucial factor in successful project completion.

References

<http://shannonxj.blogspot.com/2008/01/waterfall-model.html>

<http://idimension.wordpress.com/tag/the-scrum-master/>

<http://java.sun.com/javaee/>

<http://www-01.ibm.com/software/awdtools/reqpro/>

www.mountaingoatsoftware.com/topics/scrums

About the Author

Siddharth Sharad Chandak is a Project Manager at Cognizant. He manages a number of critical projects that Cognizant is currently executing for a key customer in the manufacturing and logistics vertical. He holds a Master of Science degree in Computer Science from Kansas State University. His interests include architecting and designing enterprise software applications, software performance tuning and project management. He can be reached at siddharth.chandak@cognizant.com.

Vishnu Rangarajan is a Senior Associate at Cognizant. He manages and supports multiple projects for key clients for Cognizant in the manufacturing and logistics vertical. He holds a Bachelor of Engineering degree in Electrical and Electronics from SRM Engineering College in India. His interests include database administration and project management. He can be reached at varadhavishnu.rangarajan@cognizant.com.

About Cognizant

Cognizant (NASDAQ: CTSI) is a leading provider of information technology, consulting, and business process outsourcing services, dedicated to helping the world's leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 50 delivery centers worldwide and approximately 130,000 employees as of September 30, 2011, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at www.cognizant.com or follow us on Twitter: Cognizant.



World Headquarters

500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277
Email: inquiry@cognizant.com

European Headquarters

1 Kingdom Street
Paddington Central
London W2 6BD
Phone: +44 (0) 20 7297 7600
Fax: +44 (0) 20 7121 0102
Email: infouk@cognizant.com

India Operations Headquarters

#5/535, Old Mahabalipuram Road
Okkiyam Pettai, Thoraiyakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060
Email: inquiryindia@cognizant.com