

---

---

# **Gateway's Capabilities in Executing Projects using Agile Methodology and Concepts**

---

---

Copyright © 2006-2007 **Gateway TechnoLabs Pvt. Ltd.**

This document and any files with it are for the sole use of the intended recipient(s) and may contain confidential and privileged information. If you are not the intended recipient, please destroy all copies of the document. Any unauthorized review, use, disclosure, dissemination, forwarding, printing or copying of this document or any action taken in reliance on this document is strictly prohibited and may be unlawful.

Visit us at [www.gatewaytechnolabs.com](http://www.gatewaytechnolabs.com)

**TABLE OF CONTENTS**

- 1. DEVELOPMENT METHODOLOGIES AT GATEWAY..... 3**
- 2. GATEWAY’S AGILE DEVELOPMENT METHODOLOGY (GAD) ..... 4**
  - 2.1. KEY FEATURES OF GATEWAY AGILE DEVELOPMENT METHODOLOGY .....5
    - 2.1.1. Requirements Gathering with user stories and high level Use Cases ..5**
    - 2.1.2. Project Management, Tasks/Iterations planning with business users 5**
    - 2.1.3. Test Driven Development .....5**
    - 2.1.4. Continuous Integration.....6**
    - 2.1.5. Continuous Refactoring.....6**
    - 2.1.6. Shorter Iterations and Frequent User Feedbacks .....6**
    - 2.1.7. Process Adaptation .....7**
  - 2.2. KEY BENEFITS OF AGILE METHODOLOGY .....7
- 3. PROJECT CASE STUDY ..... 8**
  - 3.1. DEVELOPMENT FOR A LARGE MANAGED NETWORK SERVICES COMPANY.....8
    - 3.1.1. Client Overview .....8**
    - 3.1.2. Brief description of work done .....8**
    - 3.1.3. Development Process.....8**
- 4. OTHER PROJECTS EXECUTED IN AGILE METHODOLOGY ..... 12**
  - 4.1. DEVELOPMENT OF AN ENVIRONMENTAL DATA MANAGEMENT APPLICATION ..... 12
  - 4.2. DEDICATED PORTAL DEVELOPMENT FOR A COMPANY IN AUSTRALIA..... 13
  - 4.3. DEVELOPMENT OF A POCKET PC RETAILING APPLICATION FOR A US CLIENT..... 14

## **1. DEVELOPMENT METHODOLOGIES AT GATEWAY**

At Gateway, the project development methodology and approach followed depends on various factors and project inputs. We follow iterative development methodologies, RUP or agile development approach depending on the project type and clients requirements.

As far as iterative methodologies are concerned, we have a set of well-defined software engineering processes for iterative development life cycle. Each of these processes ranging from requirements to release is based on industry-standard software development practices. Our processes have evolved over a period of time, to incorporate best practices in requirements management, project management and oversight, design methods and testing methodologies and tools.

The development methodology to be applied on a project is decided after analyzing client's requirements and project type. We follow adaptable process model where our teams work closely with our clients to define a development methodology.

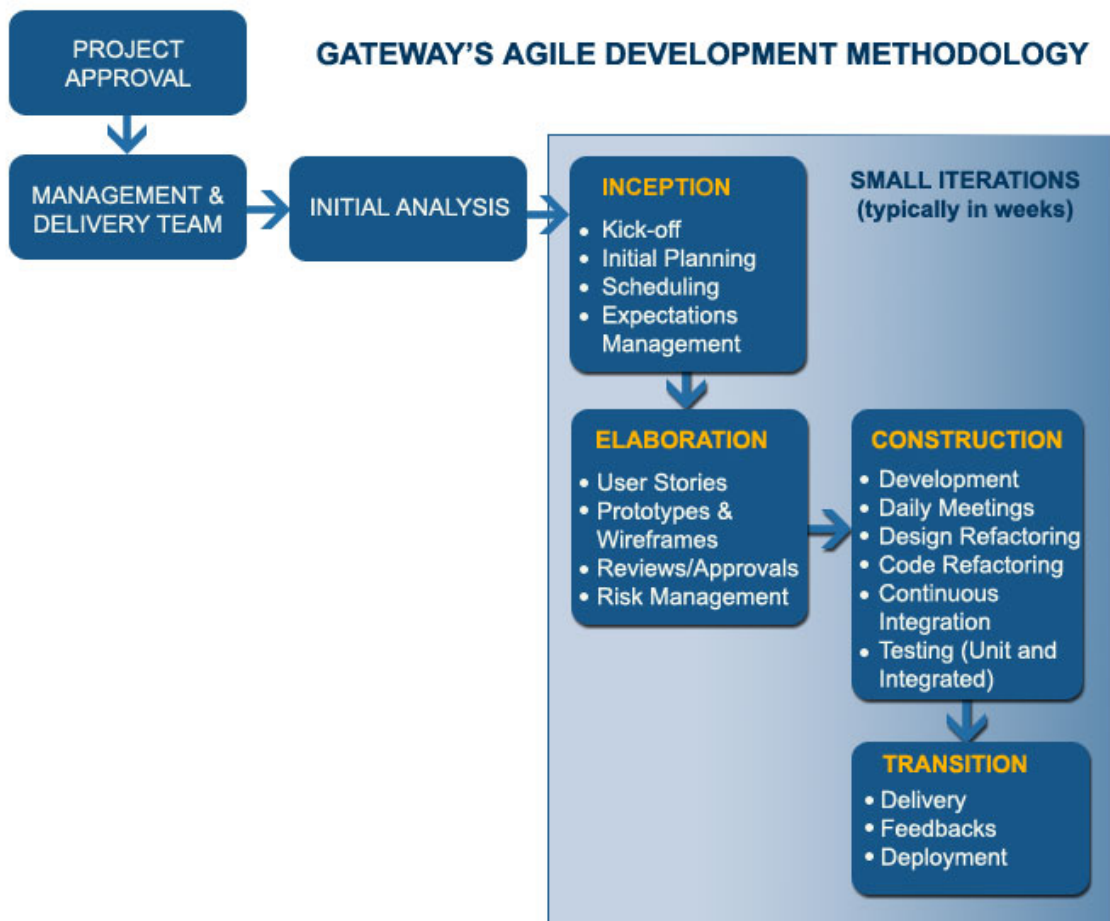
We have wide and rich experience in implementing following methodologies for all the projects executed so far:

- Waterfall Model executed in iterative Development Cycle
- V-Model and Spiral development cycles
- Rational Unified Process
- Agile Methodologies

## 2. GATEWAY'S AGILE DEVELOPMENT METHODOLOGY (GAD)

Gateway Agile Development Methodology (GAD) brings together the power of iterative methodologies with the flexibility of agile principles. Our Agile Methodology is a disciplined development philosophy which combines our development experience with the current industry standards.

The principles of agile methodologies resonate very well with our business model where in we extend our development and management services to our clients worldwide. This process allows our development teams to effectively respond to the changing needs of businesses and requirements. Gateway Agile Development Methodology combines iterative development model (RUP) with the extreme programming methods and is optimized for offshore development model.



## **2.1. KEY FEATURES OF GATEWAY AGILE DEVELOPMENT METHODOLOGY**

### **2.1.1. Requirements Gathering with user stories and high level Use Cases**

Gateway has executed multiple projects where our teams work on a given set of functionalities and the features of the system are evolving. In such cases, we extensively use voice, instant messaging and other collaborative tools like Wiki's to gather requirements. Particularly, in cases of voice communication over phone calls, we record the discussion as an important reference. We then document our understanding in form of user stories or high level use cases and get a signoff from the client. At times, the user stories we write also covers implementation details (technical story) for functionalities covered in a particular iteration (release cycle).

### **2.1.2. Project Management, Tasks/Iterations planning with business users**

The dedicated project manager keeps close communication with the client side project co-ordinator for defining, implementing and delivering the set of features.

Apart from our daily meetings with clients (through voice and instant messaging), we also have special planning sessions (Iteration Planning Meeting) with clients where we jointly work on a plan for every iteration and tasks associated with it. This imparts complete visibility to the client on the tasks that are being carried by the offshore development team. We also have client visiting us frequently and our development teams going onsite to foster face to face human interactions.

We have exposure to usage of Wiki's where the users put their stories, design details, progress notes etc. Apart from that, we have a wide exposure in using collaborative tools like Mantis, JIRA, DotProject and other proprietary project management applications for collaborative issue and task management.

### **2.1.3. Test Driven Development**

We have done test driven development for one of the large scale financial applications where we wrote the functional test cases before the coding activity and functional and unit level test cases were executed early and often. We also have relevant expertise in writing automated test scripts using

xUnit tools for PHP, Java and .NET. We have used tools like xUnit and Selenium to automate the functional tests that focus on requirement of code. After these tests are executed, the code is refactored (cleaned up/fixed).

#### **2.1.4. Continuous Integration**

We rebuild and test the application frequently through a serialized build process. At the end of each change/feature development, the developer builds the application, runs the test and after the tests pass, updates the code in the source repository.

We also have worked in models where we build frequently on development environment (sometimes, every few hours) and then release them frequently on the test and/or production servers (sometimes every alternate day).

#### **2.1.5. Continuous Refactoring**

As the features are added to the system, refactoring is done on an ongoing basis to ensure that we tune up the design and code to meet the requirements of the system. Each change, feature or module development is preceded by a refactoring exercise to keep the code clean and design simple/scalable. Refactoring is done on code and design without changing the behavior of the application.

#### **2.1.6. Shorter Iterations and Frequent User Feedbacks**

At Gateway, the typical iteration length is one to two weeks with multiple intermediate builds and demos in case of agile projects.

With frequent demos/releases, we ensure that we involve the business users into the development process and gather their early feedbacks and re-prioritize the tasks. The best time to incorporate a change is while developing, and we do this with frequent releases of working application in every iteration.

Gateway has widely adopted the practice of showcasing the in-development work to the clients using remote desktop software and collaboration tools like GotoMeeting and LiveMeeting. This is not only an opportunity for us to build links between onshore and offshore team but also get the business users to

see the application and suggest changes/feedbacks that can be incorporated early in the development cycle.

In many cases, we have simulated the exact environment as that of the clients to weed out the possibilities of differences in application behavior because of different environments.

### **2.1.7. Process Adaptation**

We focus on adapting our processes, documents and templates to that of our clients. We define and design the documentation, formats that suit both the teams, hence adapting our agile development process with that of our clients. One of the key advantages being that when we start working as an offshore development team, the client still works with the same set of processes they are used to.

## **2.2. KEY BENEFITS OF AGILE METHODOLOGY**

- Complete team work based Adaptable Model
- Client has working software at the end of each iteration (which is typically in weeks)
- Progress is measured based on working software
- Easy integration of late changes
- Close communication with the business people on daily basis
- Project thrives on close collaboration and trust between the onsite and offshore teams
- Continuous attention to better design and code through on-going refactoring
- Smaller iterations are easy to manage
- Ideally suited for offshore development model and distributed development model.
- Change Driven or Feature Driven Development
- Quickly deliver what business needs. Development teams can move at the same speed with the business needs.
- Faster Deliveries on time
- Transparency in development processes

### 3. Project Case Study

#### 3.1. DEVELOPMENT FOR A LARGE MANAGED NETWORK SERVICES COMPANY

##### 3.1.1. Client Overview

The client is a large managed network Services Company based in US offering comprehensive networking services and OES support to clients like IBM, Ingram Micro, British Telecom, Hughes amongst many others

##### 3.1.2. Brief description of work done

Gateway's offshore team is supporting the client for last 3 years and we have worked across three major versions of their management portal which integrates more than 54 different applications. Apart from this, our teams provide support for their CRM and OSS portal across two different versions. We also provide day to day support for their production servers by doing data fixes and updates.

##### 3.1.3. Development Process

###### Requirements Gathering

- We heavily rely on one-to-one communication with the business users via dedicated phone lines and VOIP phone calls. Phone calls are recorded for reference. All recorded communications and requirements are archived in VSS as a reference.
- We write understanding documents (a variant of user stories) to elicit the requirements and send them via email to the client for review/approval.
- The document is also uploaded on collaborative issue tracking tool for reference.
- We maintain a requirements traceability matrix to track each requirement through the development cycle.
- The requirements are categorized into 6 main types, namely New Development Request, Feature Enhancement, Modifications, Bug Fixes, Data Updates and Data Fixes.
- We also design quick wireframes and/or non-functional prototypes for extending feel of the application to the business users.

### Iteration Planning Meeting

- As soon as the understanding document is delivered to the client, we do an iteration planning meeting with the client.
- We identify the estimated time based on the Requirements Traceability Matrix and agree on the timelines and define iteration.
- Based on this, we agree upon a timeline and allocate necessary resources from the team depending on priority of each task.

### Test Planning and Test Design

- The requirements gathering phase is followed by creation of test scenarios for the given work (manual or automated scenarios). We write manual test scenarios wherein we also write cases to verify the database updates. For specific requirements, as per the clients needs, we also write automated test scripts in NUnit before the development begins.
- Tests are executed after each piece of work is accomplished.
- Testing pattern employed depends on the client requirements. We do manual testing based on the test scenarios or we also write NUnit based scripts for full integrated testing, whenever applicable.
- Test findings are reported in collaborative issue tracking tool.
- After a change is implemented, we update the build on staging server where they are tested by the test team and then demonstrated to the client.
- All finished iterations are first moved to staging server where integration and regression test is performed by a dedicated test team member to verify RTM and breaks due to changes.
- Peer code reviews were done on a periodic basis to ensure that the code is reviewed and optimized. We used Microsoft FxCop to identify major design related findings in source after which the code and design were again refactored.

### Development and Continuous Integration

- All development is done of client servers over remote desktop connection after secured connection to their network via VPN (Customized Open VPN).
- Development was carried out based on the test cases/scenarios and the requirements document (user stories) referring to the coding guidelines.
- The source, database code (procedures, views, triggers etc.) are all kept under version control system (Microsoft VSS) at clients end.
- Minor iterations are moved to production on completion every 3-4 days and major iterations are moved once in every few months.
- The development team works during the day time (IST) and few team members are available during the PST day time to provide production support in case of issues being found in the new iterations on production servers.
- Constant Refactoring of design and code to ensure performance, stability and optimized design.
- Continuous integration becomes easy since we work from client's environment and provide builds in client environment.
- We have a dedicated bandwidth with backup internet lines to ensure 24X7 connectivity with our client's network.
- A development tracking document is maintained and updated to reflect the current version in development, staging and production environments for each module of the application.
- After each change is done, we update the build at the staging server where it is tested by the testing team and demonstrated to the client.
- After approval from client and our internal testing team, the build is moved to production environment.
- The deployment code is isolated from the staging and updated for move to production. All production moves are made at fixed off peak times with care being taken to ensure least down time.
- If down time is going to be more than a few seconds then proper notification is sent out well in advance. A log is maintained of all changes made with backups handy for roll back in case of failure.

- Records of all changes and integration are maintained in a tracking sheet (object modification list) mentioning versions of each object on development, staging and production servers. This checklist comes in handy when moving the builds to staging and production environments.

#### Project Management

- A dedicated project manager works closely with the client on requirements, iterations and deliveries.
- Requirements are tracked using traceability matrix and estimations are done jointly with the client.
- The team members update the project status on a daily basis via emails and also using collaborative tracking tool.
- Multiple methodologies are being used to do an effective version control as there are rapid revisions in the application with upgrade of baselines once every few months. We use VSS for requirements baselines, traceability matrices, .NET source code and the database source code. We use XML based versioning for web services request tracking.
- Daily calls with client on project status and updates
- Daily meetings with the teams to have project status, re-prioritize and align
- Gateway's process adapted to suit client's requirements and overall project efficiency.

## 4. Other Projects Executed in Agile Methodology

### 4.1. Development of an Environmental Data Management Application

Client Overview and Work Description	Key Process Points
<p>The client is an environmental engineering company in US and was looking for offshore partner to develop a tool to collect and report Environmental Liability data.</p> <p>Gateway team was deployed on a dedicated development model and we were expected to follow an agile process with test driven development using NUnit for development of this tool.</p>	<ul style="list-style-type: none"> <li>• Requirements from client came in form of module wise descriptions via email.</li> <li>• The client also provided us with the details about the workflow, UI Design and Database Design.</li> <li>• Client also mentioned the validations and business rules underlying each module.</li> <li>• Gateway team archived these details in versioning tool and studied them.</li> <li>• Doubts and clarifications therein were sought on twice-a-week conference calls.</li> <li>• We used Source Gear Vault for online source version control to manage concurrency between the client and our teams.</li> <li>• During development, we do all the database changes locally and then update them on the Source Gear Vault for synchronization.</li> <li>• All the development was integrated daily on the Source Gear Vault after relevant testing.</li> <li>• This required our team to maintain a very close communication with the client teams.</li> <li>• We used NUnit - a unit-testing framework tool for all .Net languages. We tested modules quickly by running the NUnit Test Suite.</li> <li>• NUnit Script contained tests for all classes of modules and we ran the test on selected functions to test. We are doing exhaustive regression testing using NUnit test scripts.</li> <li>• The coding standards were given to us by the client and we adapted our development process to suit client's business requirements.</li> </ul>

#### 4.2. Dedicated portal development for a company in Australia

Client Overview and Work Description	Key Process Points
<p>The client is a large scale human resources company from Australia. They wanted to develop an interactive job portal which allows user to submit their resumes and provide different services to job-aspirants through an interactive online application.</p> <p>Gateway was selected as an offshore development partner for development of advanced version of their portal with enhanced features in agile development model. We have so far delivered two major versions spread across many smaller iterations.</p>	<ul style="list-style-type: none"> <li>• Requirements gathering through requirements specifications per iteration.</li> <li>• Discussion of doubts/queries in daily meetings with client over voice.</li> <li>• Development of Wireframes and non-functional prototypes</li> <li>• Iterations planning by the client</li> <li>• Manual Test Scenarios were documented after the requirements gathering/understanding</li> <li>• Test Scenarios and non-functional prototypes were sent to client for approval</li> <li>• Coding was done according to our coding guidelines for features on high priority</li> <li>• Early and often testing was performed both by the developers and by the dedicated testing engineer.</li> <li>• Twice a week tested releases to the client for on-going testing and early feedbacks</li> <li>• Continuous Integration and releases thereof on the staging environment for client review and feedbacks.</li> <li>• We used collaborative issue tracking tool for managing and tracking issues</li> <li>• Project Management through daily stand up team meetings for setting priorities</li> <li>• Daily meetings and collaboration with client over dedicated phone lines, Skype and instant messengers.</li> <li>• Shorter development iterations of one and half weeks with multiple intermediate releases/demos to the client.</li> </ul>

### 4.3. Development of a Pocket PC Retailing Application for a US client

<b>Client Overview and Work Description</b>	<b>Key Process Points</b>
<p>The client is a product company specializing in development of retail products for handheld devices and Pocket PC's.</p> <p>Gateway was selected to do a pilot project where we implemented agile methodology.</p> <p>Subsequent development has been done on agile principles and the product is now a suite of three feature-rich sub-products which are being developed and supported by Gateway since last 2 years.</p>	<ul style="list-style-type: none"> <li>• The project started as a pilot project of 3 weeks with limited set of features.</li> <li>• Gateway team followed agile practices to develop the features.</li> <li>• After achieving zero-defect pilot release and knowing the fact that the requirements are evolving - Agile Development methodology was suggested.</li> <li>• Twice a week meetings with client to discuss requirement of the phase. Phase typically was not more than one week initially.</li> <li>• We created module based use cases based on our understanding and sent across to client for approval.</li> <li>• A dedicated test wrote the test cases/scenarios for the approved features.</li> <li>• We also submitted wireframes wherever the understanding needed validation from client.</li> <li>• Each iteration of a week covered development, design, refactoring, unit testing and testing.</li> <li>• Releases to client at the end of each iteration for testing. UAT points were sent in email and Gateway team solved issues to seek closure of the iteration after which next iteration would begin.</li> <li>• Daily team meetings (some times more than one meeting in a day) to discuss the progress and set the priorities.</li> <li>• Very close communication with the client on requirements, design, UAT and project management.</li> <li>• We used online collaborative issue and task tracking tool to manage tasks and defects.</li> </ul>