

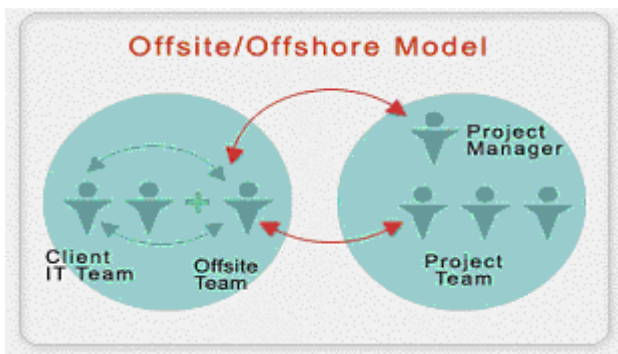
Development Models

DITL combine effective business solutions with a unique Offshore Development Model to deliver custom software solutions to help you realize your true business potential. Our Offshore development methodology uses two commonly followed approaches:

Offsite/Offshore Model

The first tier includes a Project Services Center based in Bangladesh, which is the point-of-contact for the customer. The Project Services Team, manages:

- Analysis
- Planning
- Technical Architecture
- High-Level Design
- Delivery
- Coordination of all communication between the customer and the Project Execution Center.



The second-tier Project Execution Center is stationed Offshore at the client side. The Offshore center includes a development team also headed by a Project Manager. The Project Execution team is involved in:

- Detailed design
- Construction
- Testing
- Documentation

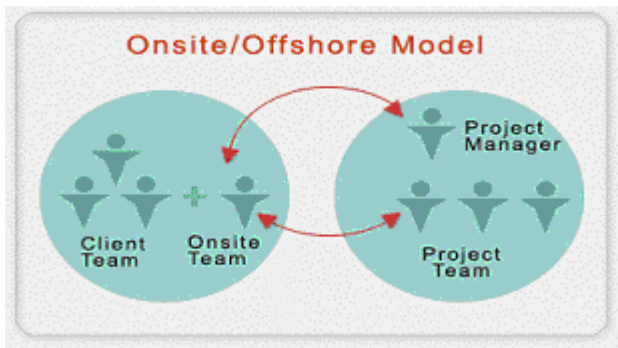
The offsite team does most of the design and deployment while the Offshore team does development & testing.

Onsite/Offshore Models

Where the nature of the project involves intense collaboration or interaction on a continual basis or there is a customer specific requirement, an onsite presence is maintained.

The onsite manager works from the customer site and coordinates project related aspects with the Offshore Development Center (ODC) and the customer. This onsite presence may involve a single resource or an entire project services team, depending on the nature, size, and complexity of the development initiative. The important features of this approach are:

- The approach requires an onsite coordinator or an onsite project manager.
- The onsite component may include one or more resources depending upon the size and complexity of the project.
- A considerable amount of the work effort can be transitioned to the Offshore facility. That transition can result in a significant cost benefit to the customer.



Processes

Software projects are inherently complex, risky and require careful planning. Planning software includes development, estimates, staged development, requirements capture, and risk and change management control procedures, business case studies, user interface prototypes and overall project control.

Proper planning ensures that the project doesn't move away from its targeted goals while the customer gets a clear definition of the project and is in the know of the project status and has ready access to project deliverables at any point of time.

To ensure the timely and quality delivery for 100% client satisfaction we implement the following:

Prototypes & Staged Deliveries

- Projects that define deliverables in terms of prototype, phases or stages identify risks and constraints early on and are able to plan and implement counter measures with greater success.
- Customers who receive small releases regularly during the project gain trust more easily than those who are in the 'dark' and are awaiting the end of the project for a whole set of deliverables that may be delivered in whatsoever form.
- The functionality is developed and delivered in stages thus ensuring that the most important function is delivered at the earliest.
- Reduces the possibility of estimation errors
- Project visibility to customers and upper management increases

Weekly Status Reports

After the end of every week, we send a Status Report to our clients telling them about:

- The activities originally planned for the last week
- The activities actually performed in the last week
- The activities planned for the coming week
- Issues, if any
- Changes to Stage Schedule, and reasons for the same
- Overall Project Status

Comprehensive Bug Management

Bugs are part of every product development process. How do you track the bugs you find during product development and after? Bugs that are found but not properly tracked might slip away. DITL uses the leading bug and defect-tracking tool Bugzilla for bug management, which leads to:

- Streamlined software development and maintenance
- Clear, updated information about the status of each bug or defect
- Full control over the work flow
- History of work done on each bug, defect or enhancement being always a click away
- A fully integrated solution that enables users to input and track new bugs, defects, and problems in general

Version Controlling

DITL uses the most popular version controlling tool CVS. On specific client requirements, we can also use Visual SourceSafe (VSS).

Knowledge Management System

DITL understands the power of Knowledge Management, and has in place a KMS in all the technical departments to ensure that **we solve a problem just once**. This drastically reduces the time of solving some of the most common, but difficult problems.

Methodology

Lifecycle Methodologies and Tools

The DITL development team adopts project methodologies based on the client's project specifications and requirements. DITL technologies has extensive expertise on the following methodologies:

Waterfall Model

This life-cycle model demands a systematic, sequential approach to software development that begins at the Customer's software requirements and progresses through analysis, design, coding, testing and post development warranty and is considered an ideal choice when the user's software requirements are clearly stated at the inception of the project.

Object Oriented Model

Each Object Oriented Development Project that is taken up by DITL may go through all or some of the phases of the Software Development Life Cycle (SDLC) defined by DITL's QMS procedures. This methodology is used to define the activities and work products for each phase and in projects where the development tasks arrive as work packets. The phases of execution, the associated work products, verification and validation criteria for each of the relevant phases shall be at par with this methodology.

Prototyping Model

This methodology defines a mechanism to handle concept building and / or prototyping projects and is used by DITL in complex projects in order to understand requirements better, to reduce design risks and to share the user interface with the Customer. Concept building projects are typically of an R&D type, where the goal is to arrive at an optimal solution based on a short description of requirements by the Customer. 'Throwaway' or 'Evolutionary' prototyping (Spiral Model) are used depending on whether the model would be discarded after use or would be adapted after use until it eventually evolves into the product.

Incremental Model

The Incremental model of development is an evolutionary model that combines the elements of the linear sequential model (Waterfall model) and the iterative philosophy of Prototyping and is considered ideal for a project that is complex by nature having large business components and interfaces with third party business applications, requiring high availability, and tight security. It also helps in managing the technology risks by spreading the risk across successive increments. This unique methodology has the distinct advantage of getting developed, quality assured and demonstrable functionality at the end of each iteration, which can be improved upon with successive iterations to get the desired functionality. In other words, early increments are "stripped down" versions of the final product, but they do provide capability that serves the user and also provide a platform for evaluation by the user.

In the Project lifecycle, we use tools which facilitate or effectively document the following activities,:

- Project Management and Planning (PMP)
- Configuration Management (CM) & Version Controlling
- System Architecture Design
- Automated Testing
- Bug Management

Quality

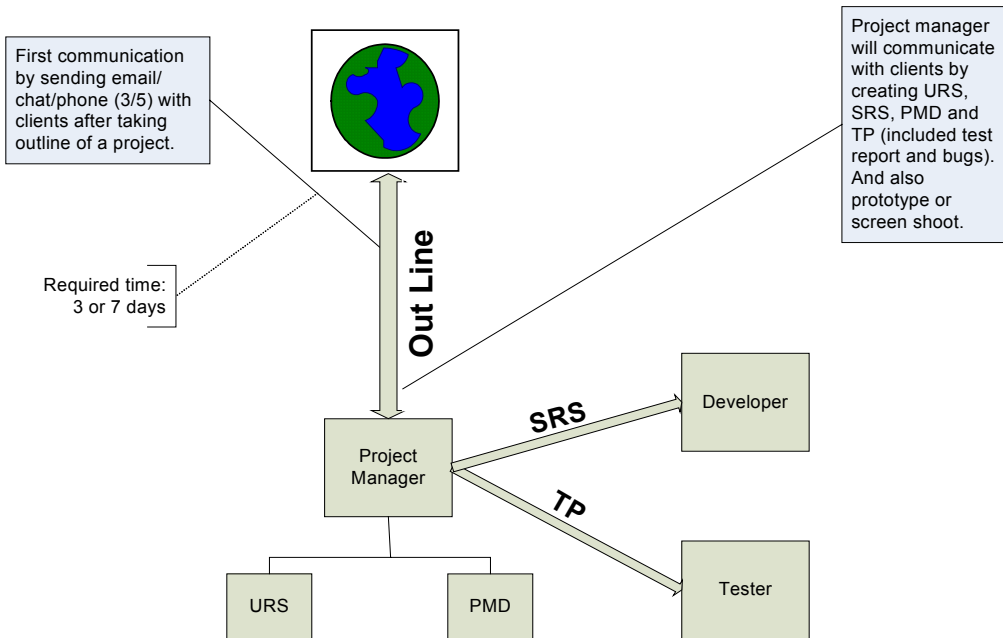
Quality is an ever-extending goal - the better you are, the better you need to be. The management of the quality process is infinite, and marked only by milestones, never by completion! DITL recognizes that, to fulfill our goal of self-evident quality, we need to constantly improve our deliverables to match the increasing expectations of our customers. With standardization being the key to all growth - professional, personal and financial. DITL reviews all its processes periodically and enhances them regularly.

We implement the Software Quality Assurance (SQA) process that addresses the quality assurance needs at every phase of the development cycle. Our QA team has developed a focused quality control checklist. In addition, we also have a comprehensive Quality Testing Checklist which ensures that every solution delivered by DITL measures up to the highest possible international standards.

The QA Division and Testing Laboratory includes:

- Test Planning
- Test Cases and Test Scripts creation
- Automated and Manual Functional Testing
- Regression Testing
- Test Results Reporting
- Performance Testing
- Bottlenecks Analysis

Flow diagram for working with overseas client



Flow diagram for managing overseas project

