What is the Solution Delivery Life Cycle (SDLC) Framework?

The SDLC provides a framework for implementing small to large, short-term to multi-year application development projects. This ensures that a consistent approach is used on all projects across the organization to deliver high quality solutions on time and on budget, meeting the client’s expectations.

Structured around the Project Management Institute (PMI) framework, our consulting project management methodology is suitable for all project types and sizes. While it can be tailored to suit a wide variety of needs, the benefits of this structure are always the same: repeatable, teachable and based on experience.

Iterative Development

We employ iterative development life cycles for components and/or configuration. This requires a rigorous architecture-driven approach, which is applicable to small and short term engagements as well as large and long term projects.

A Standard Language to Communicate

We use the Unified Modelling Language (UML), a standard language for specifying, visualizing, constructing and documenting the artefacts of software systems. It is also used for business modelling and other non-software systems.

Our use of UML as a standard allows project teams to easily communicate, explore potential designs and validate the architectural design of the software with users and stakeholders.

An Adaptive Process

Our iterative SDLC is an adaptive process. It accommodates risk-driven development as it recognizes that software development can be unpredictable due to changing requirements.

Our objective is to gain customer acceptance and approval at each phase of the process prior to proceeding to the next stage. This method minimizes the risk of a mismatch between requirements and deliverables and allows for the management and containment of project scope.
Inception Phase
Scope the solution as a basis for validating initial costing & budgets

Deliverables:
• Statement of Work (SoW) - includes business background, scope description, success factors & deliverables
• High level project schedule - includes activities, tasks & timelines
• Basic use case model - a high level overview of the solution
• Risk management plan - assessment of internal & external risks, along with risk mitigation strategy

These documents provide the basis for regular monitoring & project-based reporting during the life of the project.

Elaboration Phase
Problem analysis, establish a sound architectural foundation & eliminate high risk elements

The architecture foundation:
• Provides a design for the solution
• Identifies the relationships between solution entities
• Establishes the mechanisms required to ensure those relationships function cooperatively
• Details the interconnections of the solution components
• Establishes the standards to which these components should conform & how to achieve component interoperability

Transition Phase
Move the solution from development into production, making it available to & understood by the end user

This phase includes:
• Beta releases, bug-fixes & enhancement releases
• Developing user-oriented documentation
• Training & supporting users in their initial use
• Incorporating user feedback

Construction Phase
Feature & component development
This phase includes:
• Coding & development
• Data conversion & migration
• Source code control
• Release management
• Introduction to our Web-based defect tracking system
• Regular communication of progress & reviews with users & stakeholders