



Implementing EnterpriseWeb

EnterpriseWeb supports rapid iterative development to accelerate time-to-value for its customers.

The key to successful projects is clear-headed logic, both for eliciting and documenting business requirements, as well as for scope, timeline and budget management. Technology aside, there are no substitutes for tight, open and honest collaboration.

Projects are based on hybrid Agile-Waterfall project management. A Project is typically comprised of multiple Phases covering discrete, though typically interrelated Scopes of Work.

- A Phase is comprised of Sprints (generally 4-6)
- Each Sprint is 6 weeks
- Sprint have 3 two-week legs:
 - Requirements Review: results in a single authoritative scope document
 - Solution Delivery: enables focused ‘heads-down’ development execution
 - Approval: review, report and approve

Each Sprint has a clear, reasonably achievable and mutually agreed to Scope. This is important as EnterpriseWeb guarantees quality, timeliness – as long as Scope is fixed and unambiguous.

Requirements should be **MUTUALLY EXCLUSIVE COLLECTIVELY EXHAUSTIVE (MECE)**. The goal is to tease-apart business logic to understand the pieces and how they fit together, avoiding unspoken assumptions.



The Sprints give an organization regular opportunities to measure progress against project scope and allow for iterative course-correction from Sprint to Sprint where requirements evolve/expand/change.

The first Phase of a Project is called Phase 0. It starts with the post-contract kick-off with team introductions and review of plans, methods and assignments, as well as the deployment of code and establishment of core Enterprise Tables and Integrations (Security, People, Organizational Hierarchy).

Subsequent Phases support development of ‘smart’ data-driven processes and applications delivered in Sprints.

Sprint activity includes: Modeling Entities, Workflows, Views/Functions, Reports and Security

- App Domain Entity and Reference Data – Metadata that enables automation, integration and/or reporting
- Workflow – Loosely-coupled Tasks for dynamic data-driven / rules-based processes - Conditions must reference Enterprise Entities, App Domain Entities or Reference Data
 - Pre-Conditions = assignments (“who can perform this Task”)
 - Task Logic = personalization (“what information and capabilities support input and decisions)
 - Post-Conditions = next-best-actions (“what Task can follow”)
- ‘LiveList’ – which AppEntity metadata is available for quick search in LiveList (default view, and others)
- Functions – Any existing functions (look ups, etc.) that need modification/enhancement? Any new functions required? In either case, what metadata is involved and what does the function produce or do?
- Views / Reports – other than LiveList – what metadata is involved in what format
- ‘Object’ Security – Any limitation on who can ‘view’ the Create New Button. If they can see Create New, then default assumption is they can ‘view’ LiveList records for that Case (at least their own), what if any Unit/Position policies constrain the views.



Project Partnership

Implementing enterprise software is hard work – it requires focused execution. Customers share responsibility for achieving project objectives. EnterpriseWeb guides good design choices, based on its understanding of Customer requirements, but customers must position EnterpriseWeb to succeed.

Projects need qualified domain and technical resources to support requirements development and other customer deliverables required for the project. It's also important customers have a Project Manager to coordinate their Resources and keep their deliverables on schedule.

Beyond the kick-off, most all project work can be performed remotely – saving customers travel costs and optimizing professional service dollars.