



# Salesforce Best Practices eBook

## Background

In an attempt to streamline the planning, design, implementation, and deployment of Salesforce applications, Vertex Computer Systems identified and documented recommended Salesforce solution Guidelines and 'Best Practices' for use by Agencies, and Lines of Business ("LOB").

## Strategic Salesforce Framework

Vertex's Strategic Salesforce Framework deliverables consist of high-level Governance and Guideline recommendations along with various CRM – Customer Relationship Management – 'Best Practices' only and not a set of detailed 'recipes' for all Salesforce solution options.

- Defining and documenting recommended Governance standards
  - Tools
  - Templates
  - Best Practices
- Documenting recommended Salesforce Guidelines
  - Project/Business Case Preparation
  - Planning
  - Define
  - Design
  - Build
  - Validate
  - Deployment
  - Post Go-Live Operations
  - Support
- Documenting consolidated Salesforce vision, as available
- Defining and documenting Master Systems of Record Best Practices
- Defining initial Salesforce Implementation Roadmap(s)

## Salesforce Solution Planning

Vertex's Strategic Salesforce Framework deliverables consist of high-level Governance and Guideline recommendations along with various CRM – Customer Relationship Management – 'Best Practices' only and not a set of detailed 'recipes' for all Salesforce solution options.

- Defining and documenting recommended Governance standards
  - Tools
  - Templates
  - Best Practices

- Documenting recommended Salesforce Guidelines
  - Project/Business Case Preparation
  - Planning
  - Define
  - Design
  - Build
  - Validate
  - Deployment
  - Post Go-Live Operations
  - Support
- Documenting consolidated Salesforce vision, as available
- Defining and documenting Master Systems of Record Best Practices
- Defining initial Salesforce Implementation Roadmap(s)

## Salesforce Solution Planning Process Considerations

### Project Charter

Each Salesforce project should be a by-product of a clearly defined Project Charter – one that fully supports the Customer Mission Statement. The typical components of a typical Project Charter include:

- Purpose Statement
  - Why is the project being undertaken?
  - How does this project support the
- State's Mission Statement?
  - Project Objective(s)
  - Qualitative
  - Quantitative – i.e., stated in measurable terms/metrics
  - Critical Success Factors
- Project Sponsor(s)
  - Including a personal statement from the Sponsor(s) as to why this project is important to the Customer
- Implementation Roadmap, if applicable

- Project Scope
  - In Scope functionality
  - Data Migration/Integration details
  - Scope exclusions
  - Assumptions
- Miscellaneous
  - Constraints/Dependencies
  - Critical Assumptions

### Map-Gap Analysis

Planning a Salesforce solution implementation requires a complete analysis of Salesforce solution offerings relative to Business needs and required functionality – commonly referred to as a ‘Map-Gap’ analysis. Consequently, in order to conduct at least an initial Map-Gap analysis, it’s critical that a list of required Business and Technical require-

ments be mapped to specific Salesforce license functionality, as warranted, depending upon license type.

Additionally, Salesforce suitability and selection – including license types and numbers – will be likely based upon both current and future Business requirements and functional needs. Consequently, a functional, implementation Roadmap is always recommended to support both license selection and implementation decisions.

A sample, Salesforce functional Roadmap is provided below, and each customer Salesforce Org should have a suitable Roadmap defined to justify its strategy, plans, budgets, and approvals.

Agency/LOB Business Need	Salesforce Solution	Phase 1	Phase 2	Phase 3	Comments/Assumptions
Role Hierarchies/Security	Standard Salesforce	✓			Data visibility limited by Role/Profile/BU
Data Quality	Naming Conventions	✓			Streamlining data entry, searching, and reporting
Data Migration	Data Loader	✓			Accounts, Contacts
Accounts/Person Accounts	Accounts	✓			Four (4) Record Types
Contacts	Contacts	✓			Single Record Type
Cases	Opportunities	✓			Leveraging Agency-specific needs
Activities	Tasks/Call Reports	✓			Recording Tasks and scheduling Follow-Ups
Grants Management	Custom Object	✓			TBD
MS Outlook Integration	Salesforce for Outlook	✓			w/side Panel; Windows machines only
Mobile	Standard Salesforce1	✓			iPhones, iPads, Android phones/tablets
Document Storage	Standard Salesforce		✓		Include
Contracts/SharePoint Integration	Standard Salesforce		✓		Files Connect
Legacy Integration	Oracle RDBMS		✓		Integration method TBD
Projects/Programs	Custom Object		✓		TBD
CTI Integration	Standard Salesforce			✓	Open CTI Adapter
Customer Communities	Standard Salesforce			✓	License determination TBD
Reports & Dashboards	Standard SFDC	✓	✓	✓	Key Metrics reporting and Dashboards
Curriculum Development	Agency/OB-specific	✓	✓	✓	Training Curriculum and ‘Leave Behind’
End-User Training	Agency/LOB-specific	✓	✓	✓	On-site Training Class

An application rollout Roadmap can be a combination of Standard and Custom objects with high level scope items clearly articulated. Vertex suggests a similar, customer-specific chronological and sequential (prioritized) Roadmap be developed and revised by each customer on an on-going basis to satisfy its Business needs.

## Use Cases

Similarly, documented Use Cases should be developed to support each implementation 'Phase' to support various design considerations including Org vs App, Record Types, Page Layouts, Data migration and integration, etc. Use Cases must consider the following

- Users – Internal, External, CRM-based vs Community-based
- Roles
- Business Processes
- Custom requirements
- Data Access/Visibility
  - CRUD – Create, Read, Update, Delete
  - Organizational Wide Defaults (OWD)
  - Profiles, Page Layouts, Sharing Rules, etc.

## Build versus Buy

Build versus Buy' refers to a decision on whether to base Salesforce solutions on standard Sales and/or Service Cloud functionality and configuration -- "click-not-code" using built-in configuration tools -- or whether to select more basic -- and much less expensive -- Force.com licenses which offer standard Salesforce functionality but whose standard Objects and functionality are limited to:

- Standard Salesforce OWD (Organization Wide Defaults)/Security
- User Permissions/Role Based Sharing
- Standard Salesforce1 Mobile
- Accounts
- Contacts
- Activities
- Chatter
- Identity
- Content
- Documents
- Ideas
- Reports & Dashboards
- Visual Workflow

Along with additional capabilities, including:

- Custom Object creation
- Mobile SDK custom solutions
- Integration capabilities
- AppExchange
- Data migration tools

## Critical Success Factors

Like with any Business solution implementation, any Salesforce implementation effort must address a number of critical success factors including:

- Active and on-going Executive Sponsorship
- Sufficiently detailed requirements – i.e., project scope
- Adherence to Salesforce implementation 'Best Practices'
- Leveraging standard Salesforce functionality to the maximum extent possible
- Adherence to a clearly defined scope
- Data migration and integration points clearly defined

Individual Agencies may have internal groups, committees, etc. to support its initiatives, and Vertex recommends each customer follow its own individual, proven practices to support its Salesforce initiatives.

### Salesforce License Types Available

- Force.com Unlimited App license (Force.com)
- Sales Unlimited License (Sales Cloud)
- Service Unlimited License (Service Cloud)
- Salesforce Communities

## Implementation Methodology

No single Implementation Methodology is right for all Salesforce solution design and development initiatives, but, generally speaking, the following guidelines may help:

### PoC

- Useful for confirming technical or functional suitability for defined Business needs – e.g., testing an AppExchange offering, API, etc. – based upon a very defined set of Business and Technical requirements

### Pilot

- Often an extension of a PoC to confirm Business or Technical suitability
- Often used to check Salesforce1 and Mobile SDK solutions
- Recommended duration of  $\leq 30$  days to coincide with possible software offerings' 'free trials'

### Agile

Agile methodologies vary from formal to informal, but all are based upon short 'sprints' based upon that sprint's requirements or objectives.

### Waterfall

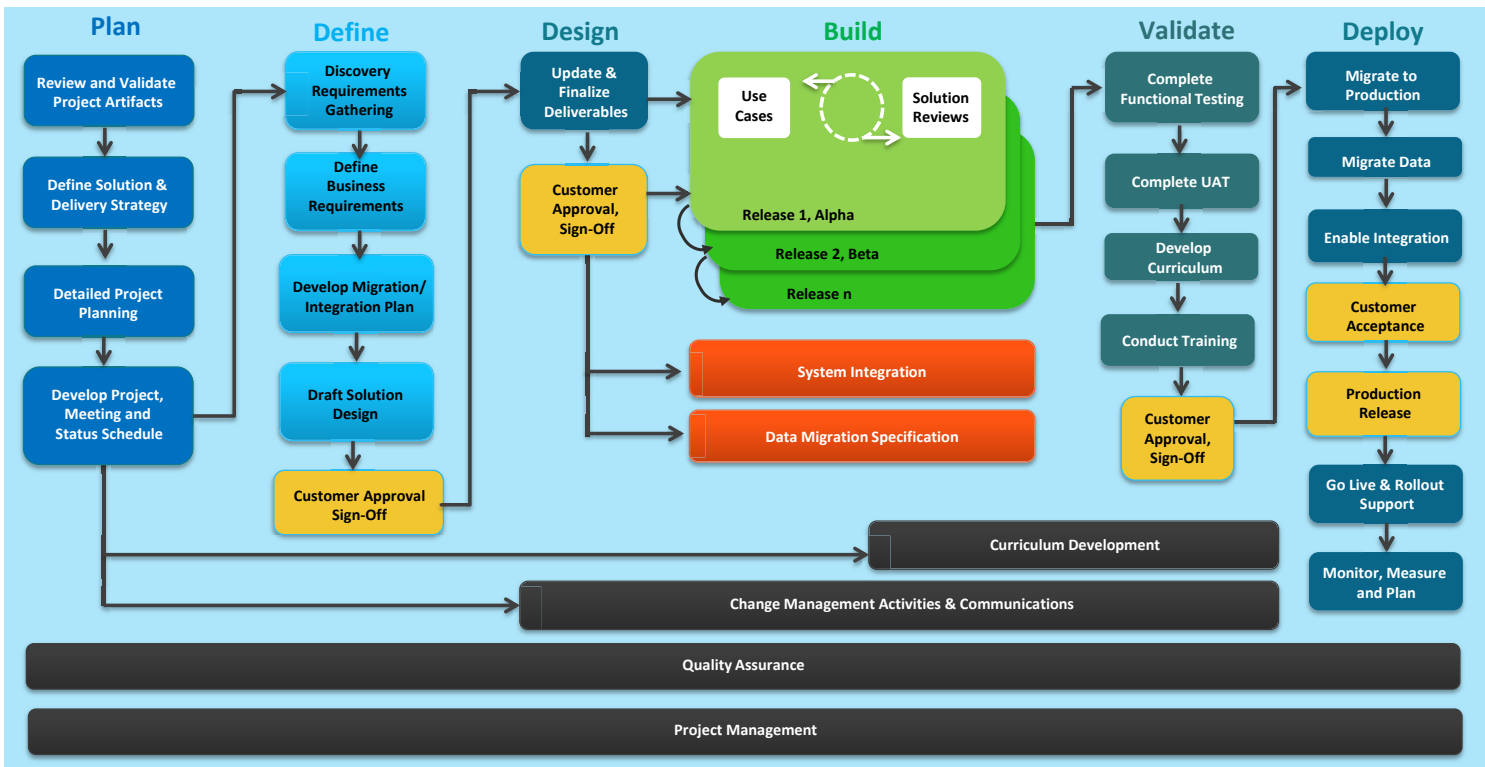
- Waterfall is often considered the typical SDLC – Software Development Life Cycle -- Methodology
- Typically beginning with gathering a full range of Business and Technical requirements and documenting those requirements in detail
- Waterfall generally includes the following – or similar -- methodology stages:
  - Plan
  - Define
  - Design
  - Build
  - Validate
  - Deploy

## Hybrid

- Except for Proofs of Concepts and, possibly, some Pilots, Vertex recommends a Hybrid methodology that combines the gathering and documenting of short-term, near-term, and long-term Business and Technical requirements.
- Minimizing implementation risks by understanding immediate and long-term scope and requirements
- Minimizing implementation risks by understanding immediate and long-term scope and requirements
- Supporting the planning of what functionality and Objects will be implemented, and when
- If the Business Use Case requires custom application development, a high-level design phase is strongly recommended. Security, Object interactions (custom, standard), data migration strategy, and integration design should be all addressed and documented
- Faster design and implementation cycles are possible if documentation requirements are less stringent
- Agile-like Build cycles are typical in a Hybrid implementation methodology (see below). A simple reference table for selecting the right Implementation Methodology is provided below.

Implementation Scenario	PoC	Pilot	Waterfall	Agile	Hybrid
Technical viability needed	✓				
Business Functional viability needed		✓		✓	
Fixed, limited implementation budget			✓		✓
Flexible implementation budget available		✓		✓	
Implementation w/defined requirements			✓		✓
Implementation w/'loose' requirements				✓	✓

Vertex's proven Salesforce 'Hybrid' Implementation Methodology is provided below for reference.



### Salesforce Define Process Considerations

### Sandbox Considerations

At the very least, all implementations should include a DEV (Developer) Sandbox; however, in most cases, a 'three-tier', configuration and development infrastructure is recommended:

1. DEV (Developer)
2. QA (i.e., Full-Copy)
3. PROD (Production)

At the very least, all implementations should include a DEV (Developer) Sandbox; however, in most cases, a 'three-tier', configuration and development infrastructure is recommended:

- What functions do you need to perform on each sandbox?
- How often do you need to refresh the sandbox?
- Do you need to test against data, and if so how much?
- Do you have to test against integrations

- Who will need access to each Sandbox (developers, testers, power users)?
- What migration tool will you use?  
Partial data and Full copy sandboxes are usually the most powerful/capable versions of Sandboxes.

## Salesforce Design Process Considerations

- Leverage Vertex's BRD Workbook to document solution requirements (and maintain all on-going enhancements with Revision History and documentation, as warranted)
- Sandbox Considerations
  - DEV
  - QA
  - Full-Copy
  - Migrations
  - Best Practices
    - Hide all Objects that are not being leveraged
    - Maintain 'Home' Object at far left of Object/Menu bar
    - Maintain a sequential flow of Objects – left-to-right
    - Typically, Reports, Dashboards will be at the far right of the Object/Menu bar  
Avoid Field Name abbreviations
    - Typically, Pick Lists should be arranged either in sequential or 'alpha' order, with the last option being "Other"
    - Avoid creating unnecessary Reports (i.e., given Salesforce's ability to filter Reports and Dashboards)
  - Activities – i.e., leveraging Tasks, Events vs 'Next Steps'
  - Object Layout – i.e., consistency and flow

- Leverage Pick Lists versus Text fields to maximize data quality and simplify Mobile data entry
- Check Boxes to streamline data entry and maximize data quality
- Hide Unused Objects/Fields
- Leverage Social Media to add Account logos (e.g., Twitter), Contacts' pictures (e.g., LinkedIn)

## Salesforce Build Process Considerations

- Sandbox Environment(s)
- Methodology
- Subject Matter Experts
- Quality Assurance
  - Unit Testing
  - Systems Testing
  - Sandbox Considerations
- Initial Data Load (Partial)
- Build Best Practices
  - Use a spreadsheet – e.g., Vertex Configuration Workbook -- to track all changes including Security, validation rules, workflows and approvals
  - Set up a maintenance window and have a strategy to test changes out on a Full Copy sandbox before migrating to Production
  - Have a proper Developer environment (depending on the complexity of your Org)
    - Continuous integration server (Jenkins, et al)
    - Github (code management repository) – very important once you have custom code in your Org
    - Leverage Force.com IDE
- Align your deployments with Salesforce release cycle

## Salesforce Deployment

- Initial Data Load(s)
  - Iterations
  - Final
- DEV-to-QA Migration
- QA-to-PROD Migration
- DEV-to-PROD Migration
- Rollout considerations
- Go-Live
- Deployment Best Practices
  - No code development/changes on Production environment
  - Tiered environment – DEV, QA, PROD
  - Limit the number of System Administrators
  - Track configuration changes in a systematic way – e.g., Excel template, Google drive, etc.
  - Prioritize your projects and enhancements
  - Use test driven deployments

- Use a Source repository (if you have more than one Admin, you need a source repository)
- Provide audits and accountability for your deployments
- Use a continuous integration server like Jenkins when using multiple Sandboxes

## Salesforce Operations' Considerations

- Data Quality
  - Taxonomy/Naming Conventions
  - Duplicate Management
- User Adoption
- Maintenance (e.g., Pick Lists)
- Enhancements
- Roadmap Updates
- Change Management
- Data (Cloud) Back-Up Recommendations
  - Data Back-Up through Salesforce data tools
  - Code Back-Up using Continuous integration tools

**Please email all your questions and your available time to discuss this cost saving method to [jim.szafranski@vertexcs.com](mailto:jim.szafranski@vertexcs.com)**



## About Us

Vertex is a CMMi Level-3 IT consulting organization that engages with its customers at a strategic level and provides 'thought leadership'. Vertex's team of Solution Scientists craft innovative solutions, with a holistic view, that make businesses smarter. Vertex acts as an advisory partner, aligning its offerings with the business goals and objectives of its customers.

For more information, reach us at [info@vertexcs.com](mailto:info@vertexcs.com)