International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

Enterprise Resource Planning(ERP) Implementation-Best Practices

Dwarkesh Ratnakar
Oracle Consultant
Highlands Ranch, Colorado, USA

***Abstract*— Organizations are increasingly transitioning from function-oriented to process-driven IT infrastructures. Enterprise Resource Planning (ERP) systems have emerged as among the most prevalent technology solutions deployed by major corporations. However, despite their widespread adoption, numerous ERP initiatives fail to achieve their objectives, frequently exceeding anticipated timelines and budgets. Considering the substantial capital investment required for ERP projects and the significant value they can deliver when properly executed, identifying the key factors that contribute to successful implementation becomes paramount. This paper examines the key factors essential for implementing Cloud/non-Cloud ERP systems from both managerial and technological standpoints. It represents a comprehensive framework designed to guide successful ERP deployment. The framework identifies critical factors across the complete implementation lifecycle.**

Keywords—ERP Implementation framework, Process Improvement, Best Practices.

I. INTRODUCTION

Organizations are becoming competitive day by day. Rapid delivery of products is crucial to survive. Traditional operational methods are ineffective in the global marketplace. The technology needs to be enhanced to improve operational efficiency and effectiveness. Many organizations are shifting from orthodox functional structures to process-oriented IT infrastructures. Enterprise Resource Planning (ERP) systems help to transform the processes & information. According to Davenport [1], ERP systems deliver multiple advantages, including accelerated decision-making capabilities, enhanced operational and cost control, reduced expenses, and improved organization-wide information flow. However, even with these advantages, many ERP implementations still fail to meet expectations. Research from Harvard Business School indicates that ERP projects frequently encounter budget overruns, schedule delays, organizational resistance to process redesign, skill shortages, and failure to realize anticipated returns on investment. There is a significant financial commitment to implementing ERP; therefore, successful implementation is crucial. This paper aims to establish a comprehensive framework to guide practitioners throughout the entire ERP implementation lifecycle.

II. CHALLENGES

A. Change Management and User Resistance

- Employees do not want to change as they are comfortable with existing processes.
- Employees fear that their jobs might be impacted.
- New Technology is to be learned by everyone in the organization.

B. Data Migration and Quality Issues

- Legacy data may have junk values in it.
- Different systems have different data formats.
- Usually, historical data has a different format, making it inconsistent with the current format.
- A decision needs to be made on which data should be migrated vs. archived.
- Multiple resources are involved in data cleansing.

C. Scope Creep and Requirement Changes

- Requirements change during the scope of implementation.
- In most cases, management may request additional features during the project.
- One phase usually overlaps with another, making it difficult to distinguish.
- Powerful stakeholders push for a certain functionality to be implemented.
- Scope changes for "Nice to have" features becoming "must-haves".

D. *Customization vs. Standardization Dilemma*

- Organizations stress the need to maintain "unique" processes.
- Every feature from legacy systems is expected to be implemented.
- It isn't easy to maintain customizations.
- It is a challenge to maintain a balance between business needs and best practices.

E. *Integration Complexity*

- It is a complex process to integrate ERP with existing third-party applications.
- Legacy systems are required to keep running in some of the Organizations.
- Some of the APIs may have limitations and compatibility issues.
- The larger the number of integration points, the higher the complexity.

F. *Resource Constraints and Skill Gaps*

- Experienced ERP consultants are difficult to find.
- Daily operations still need to be handled by Key business users.
- Internal teams may not be well-versed in the latest technical skills.
- Many people quit during long implementations.
- Knowledge is lost when people leave the team.

G. *Inadequate Executive Sponsorship*

- C-level commitment is important for the implementation.
- Big roadblocks need to be mitigated by the Executive sponsor.
- Difficult decisions require authority, which may not be vested with the team.
- Focus gets diluted if there are multiple competing priorities.
- When the budget exceeds, there is a limited authority to extend the budget.

H. *Unrealistic Timeline and Budget Expectations*

- Complete complexity may not be available during planning.
- It is difficult to deliver quality work quickly.
- Not all hidden costs have been considered.
- Resource availability constraints.
- Not all contingencies have been planned.

I. *Business Process Reengineering Difficulties*

- Organization does not want to change established workflows.
- There is coordination required with cross-functional processes.
- Different departments have political battles.
- Process may not be documented for all scenarios.
- Processes need to be simplified before they can be automated.

J. *Testing Inadequacies*

- Time allotted for testing is inadequate.
- Test scenarios may not be comprehensive.
- Testing should involve End-users.
- Emphasis on integration testing.
- Performance testing under realistic loads needs to be conducted.
- Exception scenarios need to be covered.

K. *Training and Knowledge Transfer*

- Training scheduled too early (users forget before go-live)
- Generic training, which is not role-specific.
- Insufficient hands-on practice time.
- No post-implementation refresher training.
- Documentation is not user-friendly.
- Super-users are not properly trained.

L. *Go-Live and Post-Implementation Support*

- Inadequate hyper care support for post-go-live.

- Key project team members leaving immediately after launch.
- Business continuity concerns during cutover.
- Users panic when issues arise in production.
- Balancing production support with continued enhancements.

M. Measuring and Realizing ROI

- Benefits are often intangible or difficult to quantify.
- Long payback periods.
- Baseline metrics were not established as a pre-implementation.
- Expected benefits not tracked post-go live.

III. BEST PRACTICES

A. Governance And Leadership.

- Identify a C-level executive sponsor with authority and influence.
- Ensure sponsor is visible, engaged, and accessible throughout the project.
- Executives should chair steering committee meetings.
- Empower sponsor to make binding decisions and remove obstacles.
- Schedule regular executive briefings (weekly during critical phases).

B. Create an Effective Governance Structure.

- Clear decision-making authority at each level.
- Escalation paths defined and documented.
- Regular cadence of meetings (daily standups, weekly work stream reviews, monthly steering committee).
- RACI matrix (Responsible, Accountable, Consulted, Informed) for all major decisions.
- Issue and risk registers with owners and due dates.
- Eg: Executive Steering Committee -> Program Management Office (PMO) -> Project Manager -> Functional Work Streams (Finance, HR, Supply Chain, etc.) -> Cross-functional Integration Team.

C. Project Planning and Management

- Discovery/Assessment (10-15% of timeline)
  - Current state analysis.
  - Gap identification.
  - Requirements gathering.
- Design (20-25%)
  - Future state process design.
  - Configuration decisions.
  - Integration architecture.
- Build/Configure (25-30%)
  - System configuration.
  - Data migration development.
  - Integration development.
- Test(20-25%)
  - Unit testing.
  - Integration testing.
  - User acceptance testing.
  - Performance testing.
- Deploy (10-15%)
  - Training delivery.
  - Go-live preparation.
  - Cutover execution.
- Stabilize/Optimize (Post go-live support)
  - Hypercare support
  - Issue resolution
  - Continuous improvement
- Implement Robust Project Management
  - Daily standup meetings (15 minutes max).
  - Track progress against milestones visibly.
  - Maintain a comprehensive issue log with severity ratings.
  - Weekly status reports to stakeholders.
  - Monthly steering committee presentations.
  - Use project management software (MS Project, Jira, Smartsheet, etc.).
  - Implement earned value management for large projects.
  - Visual dashboards showing RAG (Red-Amber-Green) status.
  - Burndown charts for agile work streams.
  - Dependency mapping and critical path analysis.

D. Team Composition and skills

- Project Leadership:

  - Executive Sponsor (C-level)
  - Program Manager
  - Project Manager
  - Change Manager
- Business Process Owners:
  - Finance lead
  - HR lead
  - Supply chain lead
  - Sales/customer lead
  - Manufacturing lead (if applicable).
- Technical Team:
  - Solution architect
  - Integration architect
  - Data migration lead
  - Security/compliance specialist
  - Infrastructure/network specialist
- Implementation Partners:
  - Lead consultant (experienced in your industry).
  - Functional consultants by module.
  - Technical developers.
  - Data migration specialists.
- Super Users:
  - 2-3 power users per functional area
  - Available 50-100% during implementation
  - Will become internal experts post go-live.

E. Scope Management

- Document in-scope and out-of-scope items explicitly.
- Create phased approach: MVP (Minimum Viable Product) first, enhancements later.
- Use MoSCoW prioritization:
  - Must have - Core requirements for go-live
  - Should have - Important but not critical
  - Could have - Nice to have if time/budget permits
  - Won't have - Explicitly deferred to future phases.
- Change Control Process
  - All change requests should be submitted through formal process.
  - Impact analysis (time, cost, risk) for each request.
  - Steering committee approval required.
  - Updated project plan and budget.
  - Communication of approved changes.
- Manage Customization Requests:
  - Challenge every customization with "why can't we use standard?"
  - Require executive approval for any customization.
  - Document total cost of ownership (TCO) for customizations.
  - Show impact on future upgrades.
  - Implement "vanilla first" approach.

F. Process Design and Standardization

- Challenge Current State:
  - Don't replicate broken processes.
  - Ask "Why do we do it this way?" repeatedly
  - Look for workarounds - they indicate process problems.
  - Question "we've always done it this way" mindset.
- Follow the 80/20 Rule:
  - 80% of standard processes from ERP best practices.
  - 20% configured to meet specific needs.
  - Resist customization unless it is truly differentiating.
- Simplify Before Automating:
  - Remove unnecessary steps.
  - Eliminate redundant approvals.
  - Streamline handoffs between departments.
  - Reduce manual touchpoints.
- Design for Scalability:
  - Plan for future growth.
  - Build flexible structures (chart of accounts, org hierarchy).
  - Consider global expansion requirements.
  - Design with reporting needs in mind.
- End-to-End Process View:
  - Map processes across functional boundaries.
  - Identify integration points.
  - Design for data flow, not just transactions.
  - Consider upstream and downstream impacts.

G. Data Management

- Phase 1: Assessment
  - Inventory all data sources.
  - Assess data quality using profiling tools.
  - Identify master data vs. transactional data.
  - Determine historical data requirements (typically 2-3 years).
  - Define data retention policies.
- Phase 2: Data Cleansing
  - Start 6-9 months before go-live
  - Fix at source before migration
  - Standardize formats (addresses, phone numbers, product codes)
  - Eliminate duplicates
  - Complete missing critical fields
  - Validate against business rules
  - Assign data stewards for ongoing quality
- Phase 3: Data Mapping
  - Map source fields to target ERP fields
  - Define transformation rules
  - Document business logic
  - Identify data gaps requiring manual intervention
  - Create data dictionary.
- Phase 4: Migration Approach
  - ETL Tools: Informatica, Talend, Oracle Data Integrator
  - Iterative Testing: Multiple mock migrations before go-live
  - Validation Rules: Automated reconciliation reports
  - Cutover Plan: Detailed timeline with freeze periods.
- Migration Testing Cycles:
  - Mock 1 - Technical validation (6-8 weeks before go-live)
  - Mock 2 - Business validation (4-6 weeks before)
  - Mock 3 - Dress rehearsal (2-3 weeks before)
  - Final Migration - Go-live weekend.

H. Integration Architecture

- Minimize integrations where possible (use native ERP functionality).
- Consolidate integration platforms (use middleware/ESB).
- Real-time vs. batch - choose based on business need.
- Leverage pre-built connectors wherever available.
- Design for scalability and performance.
- Build monitoring and alerting.

I. Test Strategy

Testing phases:

- Unit Testing (Build Phase)
  - Each configuration tested individually, performed by consultants, validate technical build.
- Integration Testing (System Testing)
  - End-to-end process flows, Cross-module scenarios, Interface testing.
- User Acceptance Testing (UAT)
  - Business users test real scenarios, Validates business requirements.
- Performance Testing
  - Volume testing with realistic data loads, stress testing, concurrent user testing.
- Regression Testing
  - Retest after fixing or any changes, ensure it does not break any other functionality.

Testing Best Practices:

- Prepare Test Environment
  - Replicate production environment.
  - Load realistic data volumes.
  - Include actual users in testing.
- Create Test Scripts
  - Document step-by-step test cases.
  - Include expected results.
  - Cover positive and negative scenarios.
  - Test exception handling.
  - Include integration scenarios.
  - Test month-end/year-end processes.
- Defect Management
  - Categorize by severity (Critical, High, Medium, Low).
  - Define go-live criteria (e.g., zero critical, less than 5 high defects).
  - Track defects in formal system (Jira, Azure DevOps).
  - Daily defect triage meetings.
  - Retest all fixes.
- Test Data

- Create realistic test scenarios.
- Use anonymized customer names/scenarios.
- Include edge cases and exceptions.
- Document test data for repeatability.

J. Training and Enablement

- Instructor-led training (most effective for complex topics)
- Virtual/webinar training (for distributed teams)
- Self-paced e-learning (for basic concepts)
- Video tutorials (for reference)
- Quick reference guides (job aids)
- Sandbox environment for practice.
- Tiered strategy- Tier 1: Super Users (Train-the-Trainer), Tier 2: End Users, Tier 3: Executives.

K. Go-Live Preparation and Execution.

- Create Detailed Cutover Plan:
    - Hour-by-hour schedule.
    - Task assignments with backups.
    - Dependencies mapped.
    - Go/no-go decision points.
    - Rollback procedures.
    - Communication protocols.
- Pre-Cutover Activities (Final Week):
    - Final data migration rehearsal
    - Freeze legacy systems.
    - User access provisioning
    - Final configuration validation
    - War room established
    - Communication sent to all stakeholders
- Post Go-Live Support:
    - War room staffed 24/7 (if global) or extended hours.
    - All hands-on deck.
    - 15-minute response time for critical issues.
    - Daily standup meetings.
    - Executive briefings twice daily.

IV. Common Pitfalls to Avoid

- "Lift and shift" mentality - Replicating old processes exactly.
- Over-customization - Customizing before trying standard functionality.
- Inadequate testing - Rushing through UAT to meet deadline.
- Poor data quality - Migrating dirty data.
- Weak change management - Assuming users will adapt automatically.
- Unrealistic timelines - Aggressive schedules with no buffer.
- Consultant dependency - Not building internal capability.
- Scope creep - Saying yes to every enhancement request.
- Going dark - Not communicating regularly with stakeholders.

CONCLUSION

ERP implementation success is 20% technology and 80% people, process, and change management. Organizations that invest heavily in change management, maintain disciplined scope control, build strong internal capability, and embrace industry best practices achieve significantly higher success rates.

The key is treating ERP implementation not as an IT project, but as a business transformation initiative that requires executive commitment, organizational alignment, and sustained focus on adoption and value realization.

REFERENCES

[1] **Davenport, T.H.** (1998). "Putting the Enterprise into the Enterprise System." *Harvard Business Review*, 76(4), 121-131.

[2] **Markus, M.L., & Tanis, C.** (2000). "The Enterprise Systems Experience—From Adoption to Success." In *Framing the Domains of IT Research: Glimpsing the Future Through the Past*, Pinnaflex Educational Resources.

[3] **Holland, C.P., & Light, B.** (1999). "A Critical Success Factors Model for ERP Implementation." *IEEE Software*, 16(3), 30-36.

[4] **Somers, T.M., & Nelson, K.G.** (2004). "A Taxonomy of Players and Activities Across the ERP Project Life Cycle." *Information & Management*, 41(3), 257-278.

[5] **Nah, F.F.-H., Lau, J.L.-S., & Kuang, J.** (2001). "Critical Factors for Successful Implementation of Enterprise Systems." *Business Process Management Journal*, 7(3), 285-296.

[6] **Aladwani, A.M.** (2001). "Change Management Strategies for Successful ERP Implementation." *Business Process Management Journal*, 7(3), 266-275.

[7] **Al-Mashari, M., Al-Mudimigh, A., & Zairi, M.** (2003). "Enterprise Resource Planning: A Taxonomy of Critical Factors." *European Journal of Operational Research*, 146(2), 352-364.

[8] **Xu, H., Nord, J.H., Brown, N., & Nord, G.D.** (2002). "Data Quality Issues in Implementing an ERP." *Industrial Management & Data Systems*, 102(1), 47-58.

[9] **Umble, E.J., Haft, R.R., & Umble, M.M.** (2003). "Enterprise Resource Planning: Implementation Procedures and Critical Success Factors." *European Journal of Operational Research*, 146(2), 241-257.

[10] **Bingi, P., Sharma, M.K., & Godla, J.K.** (1999). "Critical Issues Affecting an ERP Implementation." *Information Systems Management*, 16(3), 7-14.

[11] **Finney, S., & Corbett, M.** (2007). "ERP Implementation: A Compilation and Analysis of Critical Success Factors." *Business Process Management Journal*, 13(3), 329-347 .

[12] **Ruivo, P., Oliveira, T., & Neto, M.** (2015). "Using Resource-Based View Theory to Assess the Value of ERP Commercial-Packages in SMEs." *Computers in Industry*, 73, 105-116.