TrustFort IT Consulting

trustfort.net

ERP Implementation Checklist

1. Project Planning and Initiation

- **Define Objectives and Scope**
- **Identify Primary Goals**
- Streamline processes, improve data visibility, enhance compliance, etc.
- **Determine Implementation Approach**
- **Decision Point: Lift-and-Shift vs. Clean Start**
- **Lift-and-Shift:** Migrating existing processes and data as-is.
- **Clean Start:** Re-engineering processes for optimization.
- **Considerations for Decision**
- Evaluate current process efficiency.
- Assess data quality and relevance.
- Consult stakeholders on preferences.
- **Assemble Project Team**
- Assign a dedicated **Product Owner.**
- Form cross-functional Agile teams with representatives from key departments.
- **Develop Project Plan Using Agile Methodology**
- Break down the project into **sprints** with defined deliverables.
- Establish a **Product Backlog** of features and tasks.
- Set up regular **Sprint Planning, Daily Stand-ups, and Sprint Reviews.**
- **Select Implementation Partner**
- Choose a certified Microsoft Partner experienced with Business Central and Agile projects.
- Assess their ability to adapt to your specific processes.

2. Business Process Analysis and Decision on Lift-and-Shift vs. Clean Start

- **Conduct a Detailed Analysis**
- **Current State Assessment**
- Document existing processes and workflows.
- Identify inefficiencies and pain points.
- **Future State Vision**
- Define desired process improvements.
- Align with industry best practices.
- **Decision: Lift-and-Shift vs. Clean Start**
- **Lift-and-Shift**
- Pros: Faster implementation, less change management.
- Cons: May carry over inefficiencies and outdated practices.
- **Clean Start**
- Pros: Opportunity to optimize processes, adopt best practices.
- Cons: Requires more time and change management efforts.
- **Engage Stakeholders**
- Facilitate workshops to gather input.
- Make an informed decision collaboratively.
- **Document Decision and Rationale**
- Clearly outline reasons for the chosen approach.
- Communicate the decision to all stakeholders.

3. Agile System Design and Configuration

- **Establish Agile Framework**
- Adopt Scrum or another Agile framework suitable for your organization.
- Define roles: Scrum Master, Product Owner, Development Team.
- **Product Backlog Creation**
- List all features, customizations, and integrations required.
- Prioritize items based on business value.
- **Sprint Planning**
- Plan short sprints (2-4 weeks) focusing on specific modules or functionalities.
- **Configure Core System Settings in Iterations**
- **Sprint 1:** Set up company information, core financial settings.
- **Sprint 2:** Configure sales and purchasing modules.
- **Sprint 3:** Implement inventory and manufacturing modules.
- **User Story Development**
- Write user stories capturing requirements from the end-user perspective.
- Example: "As a sales manager, I want to generate real-time sales reports to make informed decisions."
- **Continuous Integration and Testing**
- Integrate and test configurations at the end of each sprint.
- Adjust based on feedback.

4. Data Migration Strategy

- **Determine Data Migration Approach**
- **Lift-and-Shift Scenario**
- Migrate existing data with minimal changes.
- **Clean Start Scenario**
- Migrate only essential data, archive or cleanse old data.
- **Data Migration Planning**
- Identify data sources and types.
- Decide on data to be migrated vs. archived.
- **Data Cleansing and Preparation**
- Clean data to ensure accuracy.
- Standardize data formats.
- **Iterative Data Migration**
- Migrate data in phases aligned with sprints.
- Validate data integrity after each migration.

5. Integration and Customization in Agile

- **Identify Integration Requirements Early**
- Include integration tasks in the Product Backlog.
- **Develop Integrations Iteratively**
- Integrate systems incrementally during relevant sprints.
- **Customization Development**
- Prioritize customizations based on business value.
- Develop custom features in sprints, ensuring they align with standard processes where possible.

6. Testing and Feedback Loops

- **Continuous Testing**
- Perform testing at the end of each sprint.
- Types of Testing:
- **Unit Testing:** By developers during sprints.
- Functional Testing: Verify user stories meet acceptance criteria.
- **Regression Testing:** Ensure new changes don't affect existing functionality.
- **User Acceptance Testing (UAT)**
- Involve users in testing during sprints.
- Gather feedback for immediate adjustments.
- **Review and Retrospectives**
- Conduct Sprint Reviews to demonstrate functionality.
- Hold Sprint Retrospectives to discuss improvements.

7. Training and Change Management

- **Iterative Training Approach**
- Provide training on features as they are developed.
- Use a "train-the-trainer" model for scalability.
- **Change Management Strategy**
- Communicate changes regularly.
- Address concerns promptly.
- **Engage Users Early**
- Encourage user involvement in sprints to increase buy-in.
- **Support Materials**
- Develop user guides and FAQs incrementally.

8. Go-Live Preparation in Agile

- **Final Sprint(s) for Go-Live Readiness**
- Focus on system stabilization and final preparations.
- **Conduct Final System Testing**
- Perform end-to-end testing.
- **Dress Rehearsal**
- Simulate go-live in a controlled environment.
- **Communication Plan**
- Inform all users of go-live dates and expectations.
- **Risk Assessment**
- Review potential risks and mitigation strategies.

9. Go-Live and Post-Implementation Support

- **Go-Live Execution**
- Execute cutover plan as per schedule.
- Monitor system performance closely.
- **Provide Immediate Support**
- Have Agile team members available for rapid issue resolution.
- **Gather Feedback**
- Encourage users to report issues.
- Log and prioritize post-go-live enhancements.

10. Continuous Improvement and Adaptation

- **Post-Implementation Sprints**
- Plan additional sprints for enhancements and new features.
- **Regular System Audits**
- Schedule reviews to ensure the system evolves with business needs.
- **Adopt DevOps Practices**
- Streamline deployment of updates and new functionalities.
- **Encourage Innovation**
- Create channels for users to suggest improvements.
- **Align with Organizational Processes**
- Continuously adapt the ERP system to support changes in business processes proposed to stakeholders

Additional Considerations

- **Decision Documentation**
- Keep detailed records of decisions made, especially regarding Lift-and-Shift vs. Clean Start.
- **Change Control Board (CCB)**
- Establish a CCB to manage changes during the project.
- **Knowledge Transfer**
- Plan for ongoing knowledge sharing between the implementation team and internal staff.
- **Performance Metrics**
- Define KPIs to measure the success of the implementation.
- **Budget and Time Management**
- Monitor budget and timelines, adjusting as needed.

Tips for a Successful Agile ERP Implementation

- **Embrace Flexibility**
- Be prepared to adapt plans based on feedback and changing requirements.
- **Prioritize Communication**
- Maintain open lines of communication within the team and with stakeholders.
- **Focus on Delivering Value**
- Prioritize features that deliver the highest business value early.
- **Empower the Team**
- Encourage team members to take ownership and make decisions.
- **Celebrate Milestones**
- Acknowledge and celebrate successes at the end of each sprint to maintain morale.