Sample Implementation Plan

Phased build, QA, launch, and change control

Example artifact, not client specific

Purpose

Ship a production system with predictable scope, test coverage, and measurable outcomes.

Success Metrics

Set before build

Speed-to-lead

Target under X minutes from inquiry to first response

Lead to booked

Target Y percent conversion from qualified lead to scheduled meeting

Failure rate

Target below Z percent for system errors and missed handoffs

Cycle time

Reduce by N percent from initial contact to close

Metrics define scope, scope defines build.

Phase 0: Prep

Duration: 1 to 3 days

Deliverables

- Confirm access, permissions, and environment availability
- Lock scope documentation, workflow list, and success metrics
- Confirm required fields and dedupe rules

Dependencies

- Admin access to CRM
- Phone system integration access
- Email and SMS platform credentials
- Calendar and forms system permissions

This phase establishes the foundation for everything that follows. Without proper access and locked scope, downstream phases will encounter delays and rework. Treat this as a gate—no build starts until prep is complete.

Phase 1: Foundation

Duration: 3 to 7 days

Data hygiene

Establish required fields and validation rules to ensure clean data entry from day one. Configure field-level validation to prevent incomplete or malformed records from entering the system.

Routing and assignment

Build intelligent routing logic with fallback rules to ensure no lead falls through the cracks. Define assignment criteria based on territory, product expertise, or workload balancing.

Logging and alerting

Create baseline monitoring and notification systems to surface errors immediately. Configure alerts for failed automations, unassigned leads, or breached SLAs.

Foundation work is invisible to end users but critical to system reliability. Invest time here to avoid firefighting later.

Phase 2: Workflow Build

Duration: 1 to 2 weeks

1   **Intake to first response**

Automate lead capture, qualification, and immediate acknowledgment. Build routing logic to deliver leads to the right person or queue within target timeframes.

2   **Follow-up sequences**

Design nurture workflows with stop conditions to prevent over-communication. Include personalization tokens, A/B testing capability, and engagement tracking.

3   **Booking and handoff**

Configure scheduling automation with calendar integration and automated reminders. Build handoff protocols between SDRs, AEs, and customer success teams.

This phase brings the system to life. Each workflow should map directly back to success metrics defined in Phase 0.

Phase 3: QA and Hardening

Duration: 3 to 7 days

01

Run test cases

Execute comprehensive testing including edge cases, error conditions, and boundary scenarios. Document all test cases with expected vs. actual results.

02

Fix failures

Address identified issues systematically, adding fallbacks and retry logic where needed. Prioritize fixes based on severity and user impact.

03

Verify reporting

Confirm all dashboards and reports reflect accurate, real-time data. Validate metric calculations against success criteria from Phase 0.

Gate requirement: No launch until all critical test cases pass and reporting accuracy is verified.

Phase 4: Launch and Training

Duration: 1 to 3 days

Soft launch

Begin with limited volume to validate system behavior under real conditions. Monitor closely for unexpected issues.

Stabilization

Maintain enhanced monitoring and support availability during the initial ramp period. Address issues immediately.

1　　　**2**　　　**3**

Team training

Conduct hands-on training sessions and deliver standard operating procedure documentation. Include troubleshooting guides.

A controlled launch minimizes risk while building team confidence. The stabilization window is not optional—plan for dedicated support coverage during this critical period.

Risks and Change Control

Risk register

Risk	Mitigation
Messy data	Required fields + manual review queue
No ownership	SLAs + escalation protocols
Too many tools	Simplify stack per blueprint
Scope creep	Enforce change control

Change control rule

> **No production change without a test case and approval.**

Changes requested after launch must follow the same rigor as initial build: document the requirement, write test cases, execute QA, and obtain sign-off before deployment.

This discipline prevents regression and maintains system integrity over time.

Version 0.1 (Sample) · Example only, not client specific