

Case Study: Startup Care Services

Report By-Madhavi Kunte

Date : 14 -10 -2025

Organization Name: Startup Care Services

Type: Mid-sized, service-based startup (500–700 customers/month; support team 40–60 FTE) providing multi-channel customer support (email, web, phone, social).

Summary:

The current organization Startup Care Services is using a legacy CRM system built on a Delphi code base. This system has become outdated and is creating challenges in meeting business needs, particularly in improving the complaints management process.

Key Issues with current legacy CRM system:

Legacy/obsolete tech (Delphi): Hard to maintain, shrinking developer pool, slow to change.

Failed modernization attempt (C#): Prior conversion stalled due to **technical complexity** and **budget/resource** constraints—indicates high delivery risk if continued.

Poor fit for complaints handling: Limited workflow/SLA tracking, weak triage/routing, and no easy omnichannel intake (email/web) out-of-the-box.

Fragmented data & integrations: Difficulty integrating with other tools; weak single-customer view and reporting.

Inefficient Operations: Manual steps, Rework, and inconsistent data quality slow resolution and hurt customer experience.

Reporting/insight gaps: limited dashboards for SLA %, backlog age, time-to-resolution, and root-cause trends.

Scalability & cost concerns: upgrading the legacy stack demands heavy bespoke effort with uncertain timelines and costs.

Consideration to Move to Modern CRM or C#:

Modernise an outdated Delphi system: The current CRM is legacy Delphi and no longer meets business needs; a move to a modern stack (e.g., C#/.NET) or to a modern CRM is being considered to address maintainability and capability gaps.

Improve complaints handling & CX: The organisation's priority is faster complaint resolution, better customer experience, and stronger integration—outcomes that a modern platform (C# rewrite or COTS CRM) can enable.

Recover from a failed C# attempt—reassess options: A prior C# conversion failed due to technical complexity and budget/resource limits; leadership is now formally comparing **continue C#, procure a modern CRM, or custom build** to find a lower-risk path.

Talent and ecosystem advantages: C# offers a widely available skill pool and modern engineering tooling compared to Delphi; a modern CRM offers vendor-supported capabilities out of the box (cases, SLAs, analytics, integrations) to accelerate delivery.

Focus on efficiency, data, and scalability—not just “tech refresh”: The move is framed as platform modernisation to drive operational efficiency, better data handling/reporting, and scalable, cost-effective service operations.

Business Requirements to improve complaint Process:

Omnichannel Intake:

-Capture complaints from email/web/phone (auto-create Case with unique ID).

Auto-routing to queues:

-Route by product, severity, language, or customer tier.

SLA timers, escalations, and reminders

-Start/stop clocks; proactive alerts before breach.

Automatic acknowledgements & updates

Auto-acknowledge on intake; notify on status changes.

Knowledge & response templates

Agents use approved articles and email templates.

Single customer view

Link cases to customer, products, prior interactions.

Validation & duplicate prevention

Mandatory fields; soft/hard validation rules; de-dupe checks.

Operational dashboards

Live SLA %, backlog by queue, average time-to-resolution (TTR), reopen rate.

Root-cause & trend analysis

Structured resolution codes; monthly trend report.

Email/Web/Telephony integration

Email-to-Case, Web-to-Case; optional CTI later.

Part 2: Evaluation of Options

Option	Benefits	Risks	Impact / Suitability
V1 – Convert Delphi to C#	<ul style="list-style-type: none"> • Reuse existing domain logic & data model • Modern .NET stack with wider talent pool • Potentially lower license costs vs COTS 	<ul style="list-style-type: none"> • Prior attempt failed (complexity + budget/resources) • Hidden technical debt surfaces during rewrite • Long delivery timeline; high project risk • Still need to build workflows, SLAs, reports, integrations from scratch 	<p>Business Process Impact: Mostly like-for-like initially; slower time-to-value for complaints improvements.</p> <p>Suitability for Complaints: Medium–Low (capability accrues slowly; risk of overruns).</p>
V2 – Procure a Modern CRM (e.g., Salesforce/Dynamics)	<ul style="list-style-type: none"> • Out-of-the-box Case mgmt., SLAs, routing, Knowledge, dashboards • Faster time-to-value; proven patterns • Vendor support, ecosystem, connectors • Scales with growth; admin-friendly 	<ul style="list-style-type: none"> • License & subscription costs • Customisation boundaries; change management required • Vendor dependence 	<p>Business Process Impact: Rapid uplift—standardised intake, routing, SLA visibility, reporting.</p> <p>Suitability for Complaints: High (quick wins, measurable CX/SLA gains).</p>
V3 – Custom-Built Conversion (Greenfield)	<ul style="list-style-type: none"> • Exact fit to bespoke processes • Full control over roadmap & UX • No vendor lock-in 	<ul style="list-style-type: none"> • Highest cost/time to deliver & maintain • Requires strong product/engineering capacity • Feature parity (omnichannel, SLAs, analytics) is costly to recreate • Talent/continuity risk 	<p>Business Process Impact: Potentially excellent long-term fit, but slow initial benefits; high delivery risk.</p> <p>Long-Term Suitability for Growth: Medium (only justified if needs are truly unique and stable).</p>

Part 3: Business Analyst Approach — Requirements Gathering

Objectives and outcomes

Our objective is to capture **clear, testable** requirements to modernise the complaints process so that we reduce time-to-resolution (TTR), improve SLA compliance and customer experience (CX), and ensure the platform integrates reliably with email/web and reporting tools. By the end of discovery, we will have a **validated requirements pack**, a **prioritised MVP backlog**, baseline **KPIs**, and **TO-BE** process maps that guide configuration and testing.

1. Stakeholder engagement plan

Stakeholder interviews (1:1)

Purpose. Elicit goals, pain points, constraints, and success measures directly from decision-makers and frontline leaders.

Who. Sponsor, Operations/Service Lead, Queue Managers, Compliance/QA, IT & Data Leads.

Sample questions.

- “What would ‘improved complaints handling’ mean in 90 days? Which metrics must move?”
- “Where do complaints stall today? What rework or duplicate effort occurs?”
- “Which integrations are non-negotiable for Day 1 (email, web forms, telephony, reporting)?”
- “What compliance or audit rules (e.g., GDPR, retention, field history) must be enforced?”
- “What is the maximum acceptable SLA breach rate for the MVP?”

Outputs. Interview notes, draft success metrics initial risk/assumptions log.

2. Discovery workshop (cross-functional)

Purpose. Align on scope, outcomes, constraints, and assumptions across functions to avoid divergent interpretations.

Agenda (90 minutes).

1. Problem statement and desired outcomes (sponsor)
2. Current pain points (ops/agents): top five with impact (e.g., queue backlog age)
3. Constraints (budget, timelines, licences, policies)
4. Assumptions & risks (e.g., email-to-case is in scope; telephony phase 2)
5. Success metrics (agree 3–5 KPIs to track)

Outputs. Agreed scope boundaries, assumptions/risks list, initial KPI set and owners.

3. Process analysis

AS-IS mapping (swimlanes)

Purpose. Visualise the real, current complaint journey and identify waste and bottlenecks.

Method. Map lanes for **Customer, Agent, Queue Manager, System**. Walk through: **Intake → Triage → Route → Work/Escalate → Resolve → Close → Report**. Capture hand-offs, tools used, duplicate steps, and delays (e.g., manual triage emails, missing data causing rework).

Outputs. AS-IS diagram, pain-point log (e.g., “No pre-breach alert; escalations are reactive”).

TO-BE design

Purpose. Define the target journey that the new solution must enable.

Key design elements.

- **Intake & auto-creation:** Email/Web forms create Cases with unique IDs and auto-acknowledgements.
- **Standardisation:** Mandatory **category/severity/channel**; validation rules to prevent incomplete records.
- **Routing & SLA:** Rules route cases to correct queues; SLA timers start/stop; **pre-breach** alerts to team leads.
- **Knowledge & templates:** Agents use approved articles to speed resolution and ensure consistency.
- **Closure & analytics:** Resolution codes and root-cause tagging feed dashboards and trend analysis.

Outputs. TO-BE diagram with SLA checkpoints, queue ownership, and escalation paths.

Data, integrations, and non-functional needs

Data and systems review

Purpose. Ground requirements in facts (volumes, peaks, breach rates) and identify integration points.

Activities.

- Review 3–6 months of volume and SLA reports (where available).
- Catalogue systems: email, web forms, telephony, identity/CRM data sources, BI/reporting.
- Define **minimum dataset** for a Case (customer, product, category, severity, timestamps).

Outputs. Volume baselines, integration map, data quality issues (e.g., duplicate contacts).

Non-functional requirements (NFRs)

Areas. Security (role-based access), audit (field history for severity/resolution), performance GDPR scalability (handle peak volumes), admin/configurability (low-code).

Outputs. NFR checklist with testable acceptance criteria and owners.

Requirements elaboration

Requirements workshop

Purpose. Convert needs into **functional** and **non-functional** requirements with traceability to process steps and KPIs.

Functional examples (for MVP).

- **Omnichannel intake:** Email-to-Case & Web-to-Case with auto-acknowledgement within 60 seconds.
- **Standardised triage:** Mandatory category/severity; cannot move to “In Progress” without completion.
- **Routing & escalations:** Rule-based assignment; pre-breach alert 2 hours before SLA.
- **Single customer view:** Prior cases and interactions visible on the Case.
- **Dashboards:** SLA %, TTR, backlog age, reopen rate; filterable by queue/product.
- **Knowledge/templates:** Linked articles reduce handling time.

Outputs. Requirements catalogue with unique IDs, mapped to TO-BE steps and KPIs.

User stories and acceptance criteria

Purpose. Make the work buildable and testable.

Examples.

- **Story:** *As an Agent, I want email complaints to auto-create a Case with a unique ID so that customers receive immediate acknowledgement.*
- **Story:** *As a Queue Manager, I want P1 cases routed to the Priority queue so that high-severity items are actioned first.*
- **Story:** *As a Service Lead, I want pre-breach alerts so that we prevent SLA misses.*

Outputs. INVEST-compliant stories with AC, plus Definition of Ready/Done.

Validation, prioritisation, and sign-off

Prototyping / screen walkthrough

Purpose. Validate early and reduce rework. Show a low-fidelity screen or sandbox configuration (case layout, required fields, routing demo) and collect feedback.

Outputs. Feedback log; adjustments to fields, rules, or alerts.

Prioritisation (MVP cut)

Method. MoSCoW (Must/Should/Could/Won't) with a light value/effort check. "Must" items must directly improve SLA/TTR/CX and be feasible within MVP capacity; "Should/Could" move to Phase 2+.

Outputs. Prioritised backlog with dependencies and an MVP scope statement.

Playback and sign-off

Purpose. Confirm shared understanding before build/config and UAT planning.

Outputs. Approved requirements pack, signed MVP backlog, and baseline KPIs for post-go-live measurement.

Governance, change control, and traceability

- **Traceability matrix:** Each requirement links to a user story, a TO-BE step, and at least one KPI.
- **Change control:** Any new/changed requirement records rationale, impact on scope/cost/timeline, and approval.
- **Quality gates:** Only stories with clear AC and data fields defined will enter build; only completed acceptance tests will move to "Done".

Three-week discovery timeline (to MVP backlog)

Week 1: Understand & baseline

- 1:1 interviews; discovery workshop
- AS-IS mapping; initial data snapshot (volumes, breaches)
- Draft problem statement, KPIs, risks/assumptions

Week 2: Design & define

- TO-BE process with SLA checkpoints and routing
- Requirements workshop → functional + NFR catalogue
- Draft user stories with AC; prepare low-fi screens

Week 3: Validate & commit

- Prototype/screen walkthrough; capture feedback
 - Prioritisation (MoSCoW) → MVP backlog + dependencies
 - Playback & sign-off; finalise KPIs and measurement plan
-

Deliverables summary

1. **Requirements Pack:** Functional + NFRs, mapped to TO-BE and KPIs
2. **Process Pack:** AS-IS and TO-BE swimlanes with SLA and escalation points
3. **User Stories & AC:** INVEST stories for MVP with Definition of Ready/Done
4. **Prioritised Backlog:** MVP cut, dependencies, and high-level plan
5. **KPI Baseline & Targets:** SLA %, TTR, backlog age, reopen %, CSAT

This approach is purposefully lean: it gives stakeholders clarity quickly, protects timelines through a disciplined MVP cut, and ensures what we build directly improves SLA compliance, resolution speed, and customer experience.

Part 3: Business Analyst Approach

Stakeholder Workshops

Overview

I will run a short, structured series of workshops to align scope, design the target complaints journey, define requirements, and commit to a measurable MVP. Each session has a clear objective, agenda, attendees, and tangible outputs.

1) Discovery & Outcomes Workshop (Kick-off)

Objective: Establish problem statement, desired outcomes, scope boundaries, and success metrics.

Attendees: Sponsor/Exec, Operations Lead, Service/Queue Managers, CX Lead, IT/Data Lead, Compliance.

Agenda (60–90 min):

- Context & goals (why change; what “good” looks like)
- Current pain points & impact (SLA breaches, TTR, CX)
- Scope boundaries & constraints (budget, timelines, licences, compliance)
- Assumptions, risks, dependencies; next steps

Outputs: Agreed problem statement; scope/constraints list; KPI shortlist with owners; initial risks/assumptions log.

2) AS-IS Complaints Journey Workshop

Objective: Visualise the current end-to-end complaints process and identify bottlenecks/waste.

Attendees: Frontline Agents, Team/Queue Managers, Ops Lead, BA, QA, IT observer.

Agenda :

- Map swim lanes (Customer, Agent, Queue Manager, System)
- Walk through: Intake → Triage → Route → Work/Escalate → Resolve → Close → Report
- Identify handoffs, rework, data gaps, failure demand
- Capture volume hotspots and edge cases

Outputs: AS-IS swimlane diagram; pain-point backlog (prioritised); initial field list (mandatory vs optional).

3) TO-BE Design & SLA/Routing Workshop

Objective: Define the target process, service levels, routing rules, and escalation paths.

Attendees: Ops Lead, Queue Managers, CX Lead, BA, IT Designer/Architect.

Agenda (90 min):

- Target journey overlay on AS-IS (what changes, where value is created)
- SLA policy (per category/severity), start/stop rules, pre-breach alerts
- Assignment & routing rules (queues, skills, language, tier)
- Roles & responsibilities (RACI), operational controls

Outputs: TO-BE process map with SLA checkpoints; routing rule catalogue; RACI; operational control points.

4) Requirements & Non-Functional Workshop

Objective: Convert goals into functional and non-functional requirements tied to the TO-BE map and KPIs.

Attendees: Product Owner, Ops Lead, Compliance, IT/Data, BA, QA.

Agenda (60–90 min):

- Group needs by domain: Intake, Triage/Rules, SLA/Escalations, Communications, Reporting/Analytics, Data Quality, Security/Compliance, Integrations
- Write/validate requirement statements (ID, description, rationale, acceptance criteria at high level)
- Non-functional checklist: security, audit, performance, availability, GDPR, scalability, admin/config

Outputs: Requirements catalogue (functional + NFR) with IDs, owner, and TO-BE/KPI traceability.

5) User Stories & Acceptance Criteria Workshop

Objective: Produce a buildable, testable MVP backlog.

Attendees: Product Owner, BA, QA/Test Lead, Tech Lead, Ops Representative.

Agenda (60–90 min):

- Translate top requirements into INVEST user stories
- Define acceptance criteria and Definition of Ready/Done
- Identify data needs, validations, and edge cases per story

Outputs: Draft MVP user stories with AC; DoR/DoD; dependency notes for sequencing.

6) Prioritisation & MVP Commitment Workshop

Objective: Agree the MVP cut and sequencing based on value vs effort and KPI impact.

Attendees: Sponsor, Product Owner, Ops Lead, Tech Lead, BA.

Agenda (60 min):

- Review story set against KPIs (SLA, TTR, CSAT) and constraints
- Confirm MVP scope, dependencies, and acceptance checkpoints

Outputs: Prioritised backlog with an explicit MVP scope statement, target dates, and ownership.

7) Prototype / Screen Walkthrough & Playback (Validation)

Objective: Validate understanding and reduce rework before build/config.

Attendees: Agents, Queue Managers, PO, BA, Tech Lead, QA.

Agenda (60–90 min):

- Walk through low-fi wireframes or sandbox screens (case layout, required fields, routing demo, SLA timers)
- Collect structured feedback (what works, what's missing, what to change)
- Confirm measurement plan (how we'll evidence KPI movement post-go-live)

Outputs: Feedback log with resolutions; updated stories/AC; sign-off to proceed.

Governance & Cadence

- **Facilitation:** BA owns agendas, timeboxes, and artefacts; PO signs off on scope/stories; Sponsor signs off on MVP.
- **Documentation:** All outputs stored in a shared workspace (e.g., Confluence) with versioning and traceability.
- **Rituals:** Stand-ups (as needed), and a weekly steering checkpoint to unblock risks and confirm decisions.

This workshop plan ensures we move from problem to **agreed design**, to **testable requirements**, and finally to a **committed MVP**—all tied to measurable improvements in SLA, time-to-resolution, and customer experience.

5 Key Requirements for the CRM system.

	Requirement	Rationale	Acceptance Check
1	Omnichannel Complaint Intake (Email/Web) with Auto-Acknowledgement	Reduce manual work and confirm receipt to customers.	≥95% of emails to support@ and 100% of web forms auto-create a Case within 60 seconds and send an acknowledgement with Case ID.
2	Standardised Categorisation & Severity	Enable routing, SLA rules, and consistent reporting.	Category, Subcategory, Severity, and Channel are mandatory before moving to “In Progress.”
3	Rule-Based Routing to Queues/Teams	Ensure the right team sees the right cases quickly.	P1 cases appear in the correct queue within 30 seconds; audit shows which routing rule fired.
4	SLA Timers with Pre-Breach Alerts & Escalations	Prevent SLA misses and improve time-to-resolution.	SLA clocks start/stop on every Case; pre-breach alert triggers ≥2 hours before breach for P1; breach rate drops ≥20% vs baseline.
5	Operational Dashboards & Reports	Give real-time visibility for action and oversight.	Live dashboards for SLA%, Avg Time-to-Resolution, backlog by queue, and reopen rate; refresh ≤5 minutes and export weekly.

Part 4: Recommendation & Roadmap

Recommendation

Choosing V2 – Procure a Modern CRM (e.g., Salesforce Service Cloud).

Why this is most suitable

- **Fastest time-to-value:** Out-of-the-box **Cases, SLAs/Milestones, queues/routing, Knowledge, dashboards**—no need to rebuild core complaints features (unlike V1/V3).
- **Lower delivery risk:** Prior **C# conversion failed**; V2 avoids a risky rewrite and leverages proven patterns and vendor support.
- **Measurable CX gains:** Built-in SLA tracking, pre-breach alerts, and reporting directly target **resolution speed** and **customer experience** goals.
- **Scalable & admin-friendly:** Click/low-code configuration, AppExchange connectors, and clear upgrade path suit a growing (startup) service organisation.
- **Total cost clarity:** Licences + configuration are more predictable than a custom build's open-ended engineering effort.

Key risks & mitigations (brief)

- *Licensing cost* → Phase seats, start with MVP scope, negotiate term/discounts.
- *Change management* → Early prototyping, champions, training, and playbooks.
- *Integration complexity* → Phase integrations (email/web first; telephony/chat next) and use standard connectors.

Roadmap (high level)

Phase 1 — MVP (0–8 weeks)

Objective: Go live with a usable complaints backbone that moves SLA and TTR.

- **Discovery & design:** Finalise TO-BE process, SLAs, routing, and data fields.
- **Configure core:** **Email-to-Case, Web-to-Case, Case layout, mandatory fields, queues, routing rules, SLA timers/pre-breach alerts.**
- **Dashboards:** Live **SLA%, Avg Time-to-Resolution (TTR), Backlog by queue, Reopen rate.**
- **Data:** Light migration (recent open cases), contact de-duplication guardrails.
- **Pilot:** 1 squad for 2 weeks; iterate quickly, then org-wide enablement.

Phase 2 — Scale (8–16 weeks)

Objective: Deepen capability and coverage.

- **Knowledge & templates,** advanced routing (skills/language/tier), canned responses.
- **Integrations:** Telephony/CTI or chat (as needed), enrich reporting (BI feed).
- **Governance:** Role-based access, field history for severity/resolution, GDPR processes.
- **Training:** Playbooks, QA checks, floor-walking, hyper care.

Phase 3 — Optimise (Quarter 3+)

Objective: Continuous improvement & insight.

- **Root-cause analytics & trend reporting,** backlog aging reduction initiatives.
- **Automation:** Targeted flows (auto-categorise, smart case updates), knowledge deflection.
- **Service quality:** CSAT capture & closed-loop improvement.

Summary :

V2 gives you the quickest, safest route to visible improvements in the complaints process while setting a scalable foundation for future growth.

V2 — Modern CRM (e.g., Salesforce Service Cloud)

- **Business:** Fastest time-to-value with built-in Cases, SLAs, routing, dashboards—so SLA compliance and time-to-resolution improve within weeks. Lower change/adoption risk thanks to proven patterns.
- **Technical:** Avoids the high risk of another C# rewrite. Mature platform with vendor-managed upgrades, security, and reliable APIs/connectors (Email/Web-to-Case) → less custom code, easier maintenance, faster iterations.
- **Cost:** More predictable TCO (licenses + configuration) and phased scaling. Reduced build/regression effort and lower ongoing maintenance vs. bespoke V1/V3. Faster ROI from earlier measurable CX/SLA gains.

1) Short term — MVP (0–8 weeks)

- Finalise TO-BE process, SLAs, routing rules; configure **Email-to-Case/Web-to-Case**, Case layout, mandatory fields, queues, **pre-breach alerts**, and **dashboards** (SLA%, TTR, backlog).
- Light data migration (open cases), pilot with one team, iterate.

2) Mid term — Scale (8–16 weeks)

- Roll out to all teams; add **Knowledge & templates**, advanced routing (skills/language/tier), initial integrations (e.g., CTI/chat), role-based access + audit.

- Training, playbooks, hyper care; feed BI for reporting.

3) Long term — Optimise (Quarter 3+)

- Root-cause analytics and continuous improvement backlog; targeted automations (auto-categorise, smart escalations), CSAT capture & closed-loop actions.