T & C

From Conversation to Working Software

A real project. Real AI agents. Real outputs.
Every step documented. **Nothing hidden.**

Case Study: Grant Intelligence Platform

THE BRIEF

The client needed a SaaS platform. Fast.

THE CHALLENGE

Grant Discovery Is Broken

Organizations spend hours manually searching, filtering, and evaluating grant opportunities. Researchers review dozens of irrelevant results before finding one worth pursuing. Time-to-shortlist: ~2 hours per search session.

THE ASK

AI-Powered Grant Intelligence

Build a multi-tenant SaaS platform that ingests grants from federal sources, scores them against each organization's profile, and surfaces the best opportunities instantly. Self-serve onboarding, no account manager needed.

INTAKE METHOD

Video + Conversations + Docs

Client requirements came through video calls, Slack conversations, voice notes, and shared documents. No formal RFP. No 40-page spec. Just natural human communication captured by AI.

WHAT HAPPENED NEXT

AI Agents Took Over

From that raw input, our AI pipeline produced: product analysis, estimation, portfolio approval, technical architecture, sprint planning, and implementation — with humans approving at every gate.

STAGE 1 — INTAKE

Raw conversations became structured requirements.

Everything the client said was captured, transcribed, and structured by AI.

VIDEO

Stakeholder interviews — Client described the grant discovery problem, target users (grant researchers), and the vision for a self-serve SaaS platform. AI transcribed and extracted 18+ distinct requirements.

CONVERSATION

Slack discussions — Ongoing refinements: scoring dimensions, eligibility engine rules, set-aside detection for NHO/HUBZone, multi-tenant data isolation. Each decision captured and traced to source.

DOCUMENT

Business Requirements — Self-serve onboarding wizard (5 steps, <60 min), grant ingestion from Grants.gov + SAM.gov + SBIR.gov, scoring engine with configurable weights, verdict system (GO/SHAPE/MONITOR/NO-GO).

DECISION

Human clarification — "Search relevance scoring (#31) is a separate capability from evaluation pipeline scoring (#25). Different user journey, different triggers." — Victor, 2026-03-18. Captured and fed into analysis.

STAGE 2 — AI PRODUCT ANALYSIS

Agent decomposed the XL request into 8 capabilities.

The AIPMO Product Manager agent analyzed the full brief, identified overlaps, mapped dependencies, and produced independent product capabilities.

AIPMO PRODUCT MANAGER — ACTUAL OUTPUT

Complexity: XL — 18+ deliverables, 8 systems, 5 external dependencies
Decomposition: 1 XL issue decomposed into 8 independently estimable capabilities (6 Medium + 2 Small)
Critical path identified: C1 (Account Setup) → C2 (Grant Ingestion) → C3 (Eligibility) → C4 (Scoring)
Parallel after critical path: C5 (Pipeline View), C6 (Analysis Cards), C7 (Application Tracking), C8 (Notifications)

#	CAPABILITY	COMPLEXITY	STAGE	DEPENDENCIES
#22	Account creation and workspace setup	M	analyzed	None (foundational)
#23	Grant opportunity discovery and ingestion	M	analyzed	#22
#24	Eligibility analysis	M	analyzed	#22, #23
#25	Opportunity scoring and verdicts	M	analyzed	#22, #24
#26	Opportunity pipeline view	M	analyzed	#25
#27	Opportunity analysis cards	S	analyzed	#24, #25
#28	Application tracking	M	analyzed	#25, #26
#29	Email notifications	S	analyzed	#22, #25

Each capability is a GitHub issue with full requirements, acceptance criteria, and dependency map. View live: `github.com/t-and-c/client-grant-intelligence/issues`

STAGE 3 — AI ESTIMATION & EFFICACY ANALYSIS

Agent estimated effort and applied the Efficacy Pareto.

Not gut-feel estimates. Data-driven analysis with confidence scoring, gap assessment, and decomposition into Promise + Stretch scope.

AIPMO ESTIMATOR — ACTUAL OUTPUT (ISSUE #31)

Upstream Confidence Review: 6 claims analyzed — 4 HIGH, 1 MEDIUM (no scope impact), 1 MEDIUM (scope-affecting, 1.3x drag applied)
Gap Assessment: Greenfield — no search, scoring, or explanation capability exists. Three upstream dependencies (#22, #23, #24) required.
Decomposition: 3 capabilities — C1: Scored Results (6 pts), C2: Explanations (3 pts), C3: Configurable Weights (4 pts, Stretch)
Voice of Reason: "If C1 takes 2x, Promise total goes from 9 to 15 points. Still achievable." — Passed.

CAPABILITY	EFFORT	SCOPE	INTENT IMPACT	LEVERAGE
C1 — Scored search results	6 pts	Promise	9/10	1.35
C2 — Score explanations	3 pts	Promise	7/10	2.10
C3 — Configurable weights	4 pts	Stretch	5/10	0.96

Promise (committed): 9 points — scored results + explanations. This is what we guarantee.

Stretch (aim for): +4 points — configurable weights. If execution goes clean.

STAGE 4 — HUMAN GATE: PORTFOLIO APPROVAL

AI recommended. Humans decided.

The Portfolio Manager agent produced a recommendation. The human operator reviewed and approved with full visibility.

AIPMO PORTFOLIO MANAGER — RECOMMENDATION

Budget: 6 points needed, ~47 points available (~13% utilization)
Schedule: Schedulable with constraints — Phase 1 dependency chain
Scenario A (recommended): Full 3-factor scoring with Promise scope
Leverage: C1 = 1.35, C2 = 2.10 (highest value per effort point)
Decision: APPROVE — Scenario A

HUMAN DECISION

Approved by Joana (Product Owner)

Full commitment package reviewed:
- Scope: 2 Promise capabilities (9 pts)
- Dependencies: acknowledged and accepted
- Timeline: Sprint 1 starts immediately
- Risk: upstream chain managed separately

This is a mandatory human gate. No code is written until a human approves.

`approved` labels applied to #32 and #33 → Pipeline continues autonomously

STAGE 5 — AI ARCHITECTURE PLANNING

Agent produced a 475-line architecture document.

Not a sketch. A complete technical blueprint with data contracts, component structure, technology decisions, and dependency maps.

ARCHITECTURE AGENT — KEY DECISIONS

1. Scoring as separate workspace package (`packages/scoring/`) — pure TypeScript, no framework dependency, independently testable

2. TF-IDF for keyword matching — lightweight, deterministic, no external infra. Swappable via same interface.

3. Template-based explanations — deterministic, consistent, testable. No LLM dependency.

4. Default equal weights (33.3% each) — least biased without user research. Configurable weights are Stretch.

5. Separate from #25 scoring — different factor models (3 vs 5 dimensions), clean interfaces for future consolidation.

COMPONENT STRUCTURE

```
packages/scoring/src/
  types.ts
  factors/
    keyword-match.ts (TF-IDF)
    eligibility-alignment.ts
    deadline-proximity.ts
  composite-scorer.ts
  explanation-generator.ts
  utils/
    text-processing.ts
    date-utils.ts
```

TECHNICAL SUB-ISSUES CREATED

#35 — Scoring package (6h)
#36 — API integration (8h)
#37 — Search results UI (9h)

Full architecture document committed to repo: `docs/architecture-search-scoring.md` (475 lines)

STAGE 6 — AI SPRINT PLANNING

Agent planned sprints with back-pressure analysis.

23 agent-hours of work fitted into 2 sprints. The system flagged that 74% of work is blocked by upstream dependencies.

SPRINT 1 — ACTIONABLE NOW

TASK	EFFORT	STATUS
#35 — Scoring package	6h	in-dev

Goal: Build the scoring engine as a reusable workspace package. Unblocked — can start immediately.

SPRINT 2 — BLOCKED BY UPSTREAM

TASK	EFFORT	STATUS
#36 — API integration	8h	blocked
#37 — Search UI	9h	blocked

Requires: #22, #23, #24 (upstream capabilities still in 'analyzed' state).

BACK-PRESSURE SIGNAL (AGENT TO PRODUCT)

Significant tension detected: 74% of committed work (17 of 23 agent-hours) cannot begin until upstream capabilities #22, #23, #24 progress past 'analyzed' state. Sprint planning recommends prioritizing upstream capability advancement to unblock the dependency chain.

STAGE 7 — AUTONOMOUS IMPLEMENTATION

Coding agents build. 24/7. With full audit trail.

Once Sprint 1 was approved, the implementation agent launched a coding session. Every action logged in GitHub.

IMPLEMENTATION DISPATCH — ACTUAL OUTPUT

Planning artifacts reviewed:

- Architecture document (475 lines)
- Technical task #35 (acceptance criteria)
- Parent epic #34
- Product capabilities #32, #33

Implementation agent launched:

Session ID: 966c9acf

Scope: `packages/scoring/`

Types, 3 factor calculators, composite scorer, explanation generator, full test suite

Supervisor monitoring:

Agent progress evaluated after each session. Relaunch with focused instructions if needed.

WHAT THE AGENT BUILDS

- Package scaffolding (Turborepo workspace)
- TypeScript types and interfaces
- TF-IDF keyword match calculator
- Eligibility alignment calculator
- Deadline proximity calculator
- Weighted composite scorer
- Template-based explanation generator
- Complete unit test suite
- All monorepo checks must pass

DEFINITION OF DONE

- All acceptance criteria met
- `pnpm lint` + `type-check` + `test` pass
- Branch pushed, draft PR opened
- Human reviews working code, not slides

THE EVIDENCE

Everything is in GitHub. Open and auditable.

Not slides. Not promises. A live repository with real code, real issues, real agent outputs.

37	4	6	7	50+
Issues Created	PRs Merged	Architecture Docs	CI Workflows	Agent Outputs

PRODUCT BACKLOG

`docs/product-backlog.md` — Living document with priority, effort, leverage scores, sprint assignments, dependency chains. Updated by agents at every stage.

ARCHITECTURE DOCS

`docs/architecture-search-scoring.md` — 475-line technical blueprint. Component structure, data contracts, type definitions, technology rationale.

SPRINT BACKLOG

`docs/sprint-backlog.md` — Committed deliverables, Definition of Done, ground rules, queued items, blockers. Full transparency.

Live repo: `github.com/t-and-c/client-grant-intelligence`

THE TIMELINE

5 days. From intake to implementation.

Mar 14	**Repository created** from T&C template. CI/CD, code review, security scanning active from minute one. AUTOMATED
Mar 14	**Product intake processed.** XL brief decomposed into 8 product capabilities. Dependencies mapped. Backlog created. AIPMO PRODUCT MANAGER
Mar 14	**Old technical tasks superseded.** Pipeline rework triggered re-analysis through product-first pipeline. 11 old tech tasks replaced with 8 product capabilities. PIPELINE
Mar 18	**New capability intake (#31).** Near-duplicate detected with #25 — flagged for human decision. Victor clarified: separate capability. AIPMO + HUMAN GATE
Mar 18	**Full analysis + estimation.** #31 analyzed, decomposed (3 capabilities), estimated (13 pts), Promise/Stretch scoped. Efficacy Pareto applied. AIPMO ESTIMATOR
Mar 18	**Portfolio approval.** Budget validated, schedule assessed, commitment package reviewed and approved by Joana. AIPMO + HUMAN GATE
Mar 18	**Architecture planned.** 475-line architecture document. 3 technical sub-issues created with full acceptance criteria. ARCHITECTURE AGENT
Mar 19	**Sprint planning + implementation dispatch.** 2 sprints planned. Back-pressure flagged. Coding agent launched on #35. SPRINT PLANNER + CODING AGENT

INDUSTRY BENCHMARKS

How does T&C compare? The research is clear.

STANDISH GROUP CHAOS REPORT (2020)

Only 31% of software projects succeed.

50% are "challenged" (late, over budget, or under scope). 19% fail outright.

Source: Standish Group, CHAOS Report 2020. Sample: 50,000+ projects.

MCKINSEY & OXFORD UNIVERSITY (2012)

66% of large IT projects overrun budget.

Average: 45% over budget, 7% over time. 17% are "black swans" that threaten the company.

Source: McKinsey Digital / HBR, "Delivering Large-Scale IT Projects." n=5,400 projects.

STRIPE DEVELOPER COEFFICIENT (2018)

Developers spend only 32% of time on new features.

42% on technical debt and maintenance. 13.5 hours/week on bad code. $300B wasted globally per year.

Source: Stripe, "The Developer Coefficient." Survey of 1,000+ C-level executives.

IEEE / REQUIREMENTS ENGINEERING RESEARCH

Requirements errors cause 40-50% of all defects.

Requirements engineering consumes 15-20% of total project effort. Fixing a requirements error in production costs 100x vs. at requirements stage.

Source: IEEE, Boehm & Papaccio (1988); Jones, "Applied Software Measurement" (2008).

These are the baselines. The industry average. **This is what "normal" looks like.**

T&C VS. INDUSTRY STANDARD

Grant Intelligence: measured against the benchmarks.

METRIC	INDUSTRY AVERAGE	T&C (GRANT INTELLIGENCE, ACTUAL)
Intake to structured requirements	2-6 weeks (discovery workshops, stakeholder interviews, requirements documents)	**< 1 day. AI captured, transcribed, structured, and decomposed all inputs.**
Requirements to architecture	2-4 weeks (architecture committee reviews, design sessions, documentation)	**Same day. 475-line architecture document with rationale for every decision.**
Architecture to sprint plan	1-2 weeks (estimation poker, sprint planning meetings, team alignment)	**Same day. AI estimated, applied Efficacy Pareto, planned 2 sprints with back-pressure.**
Requirements traceability	Rare. Most projects lose the link between requirement and code by sprint 3.	**100%. Every line traces to a conversation, decision, or acceptance criterion.**
Blocker detection	Discovered mid-sprint or at delivery. "We didn't know X was blocked."	**Proactive. System flagged 74% of work blocked by upstream BEFORE sprint started.**
Developer time on new features	32% (Stripe, 2018) — rest is debt, maintenance, meetings, context switching	**AI agents: ~100% on implementation. No meetings, no context switching, no politics.**
Audit trail	Jira tickets, Confluence pages, Slack threads — scattered, incomplete, after-the-fact.	**Every agent output, every human decision, every approval — in GitHub, real-time.**

10-20x
Faster: intake to code

100%
Requirements traceability

0
Hidden blockers

24/7
Implementation capacity

This Is How We Build.

Not a concept. Not a demo. A real project, running right now, with every step traceable from conversation to code.

37	50+	475	5 days
Issues Managed	Agent Outputs	Lines of Architecture	Intake to Implementation

T&C — Outlier-Architected AI Development