HIGHTOWER.DIGITAL® · CREATE EXCELLENCE!!™

Customer Discovery — Use Case Report

IT REQUIREMENTS USE CASE · CD1 · VERSION R1.A

Author: LW Atkinson (CEO/CTO), HighTower Digital Inc.

Date: 06/25/2026 · **Classification:** HDI Confidential

Methods: Agile Scrum + Lean + XP · **Product:** IT Requirements As-A-Service (MSP SLA #1–5)

File: CD1_UCR1a

1 Customer Discovery Requirements (Design Spec Document)

Customer Discovery is the front end of requirements engineering. Following Steve Blank's Customer Development model and the Lean Startup loop, the goal is to **"get out of the building"** and validate the problem before committing engineering capacity. The deliverable is a **Design Spec** that converts unverified assumptions into evidence-backed problem, solution, and outcome statements — the StRS (Stakeholder Requirements Specification) layer of ISO/IEC/IEEE 29148.

1.1 Customer Discovery Use Case Statement

A discovery use-case statement frames the engagement using the **Jobs-To-Be-Done (JTBD)** lens — *"When [situation], I want to [motivation], so I can [expected outcome]"* — combined with the ISO 29148 problem framing. Below is the worked HighTower.Digital reference case used throughout Section 1: an enterprise client whose business requirements never survive the handoff to developers.

1.1.1 PROBLEM DESCRIPTION

Enterprise stakeholders express needs in business language; by the time those needs reach the development team they have been re-interpreted three or four times, producing rework, missed sprints, and "that's not what we asked for" at review. The measurable symptom: **38% of committed sprint stories are reopened** for clarification, and average lead time from request to merged spec is **9 business days**.

1.1.2 SOLUTION REQUIRED

A continuous, SLA-backed requirements triage service that ingests raw business needs and emits **developer-ready specifications** — API schemas, data models, acceptance criteria, and edge-case definitions — with traceability back to the originating stakeholder need, all within the client's existing sprint cadence.

1.1.3 BENEFITS NEEDED

- Reduce sprint story reopen rate from 38% to **< 10%**.
- Cut request-to-spec lead time from 9 days to **≤ 1 business day** (Partner tier) / **15 minutes acknowledgment** (Emergency tier).
- Achieve **100% developer-ready** specs (no open interpretation at sprint planning).

1.1.4 SYS/ECO CONSTRAINTS

- Must integrate with the client's existing tooling (Jira Cloud / Azure DevOps) — no rip-and-replace.
- Must operate **remotely and on-site** under the same SLA.
- Must comply with the client's data residency and security policy (e.g., ISO 27001, SOC 2).
- Real-time data systems impose a hard latency budget; specs must carry NFR latency targets.

1.1.5 REQUIRED OUTCOMES

A signed-off **Design Spec** containing: validated problem statement, prioritized feature/function lists, and a Technical Requirements index ready to feed Section 2 (Voice of the Customer).

> **Standard applied.** ISO/IEC/IEEE 29148 requires every requirement to be *necessary, unambiguous, complete, singular, feasible, verifiable, correct, and conforming*. The discovery statement above is written so each outcome is measurable and therefore *verifiable*.

1.2 Survey / Discovery Process — QA + Results

Discovery is run as five timestamped question gates, each built on the **5 W's + H** (Who, What, When, Where, Why, How). This mirrors the Lean *Build–Measure–Learn* validation cadence and "The Mom Test" principle of asking about past behavior rather than future hypotheticals.

1.2.1 `2026-06-01 09:00 EST` — PRE-VALIDATION QA

W	Question
Who	Who currently writes the requirement, and who consumes it downstream?
What	What artifact is produced today, and where does it break?
When	When in the sprint does the breakdown surface?
Where	Where (which tool/channel) does the requirement live?
Why	Why does the current handoff lose fidelity?
How	How is success measured today (if at all)?

1.2.2 `2026-06-04 10:30 EST` — VALIDATION QA

Confirms the problem with quantified evidence (reopen rate, lead time pulled from Jira). Each answer is logged against the originating stakeholder for traceability.

1.2.3 `2026-06-09 14:00 EST` — POST-VALIDATION QA

Validates the proposed solution shape with a low-fidelity spec sample; measures stakeholder comprehension ("can your developer build from this with zero questions?").

1.2.4 `2026-06-16 11:00 EST` — POST-VALIDATION-REVIEW 1 QA

Reviews the first real spec produced under the service against the success metrics; captures variance.

1.2.5 `2026-06-23 11:00 EST` — POST-VALIDATION-REVIEW 2 QA

Confirms sustained improvement across a full sprint; signs off the Design Spec for promotion to Development Spec.

> **Result of worked case:** reopen rate fell to **11%** after Review 1 and **7%** after Review 2; request-to-spec lead time dropped to **0.8 days** on the Partner tier — clearing the §1.1.3 targets.

1.3 Product / Service Features List

The discovery output is decomposed into candidate **features** (capabilities a user perceives). Representative set for the reference case:

- **F-01 Requirement Intake Portal** — single channel to submit raw business needs.
- **F-02 SLA Timer & Acknowledgment** — 15-min response clock with audit trail.
- **F-03 Spec Generator** — converts intake into a structured, developer-ready spec.
- **F-04 Traceability Matrix** — links each requirement to stakeholder, story, and test.
- **F-05 Tool Sync** — bidirectional Jira/Azure DevOps integration.

1.4 Product / Service Functional List

Functions are the system behaviors that realize the features (ISO 29148 functional requirements):

ID	Function	Realizes
FN-01	Submit requirement with attachments and stakeholder tag	F-01
FN-02	Start/stop SLA timer; emit acknowledgment within 15 min	F-02
FN-03	Parse intake into INVEST-compliant user stories	F-03
FN-04	Generate Gherkin acceptance criteria	F-03
FN-05	Create/maintain Requirements Traceability Matrix (RTM)	F-04
FN-06	Push approved story to Jira backlog via REST API	F-05

1.5 Technical Requirements

1.5.1 BUSINESS CASES

The service must demonstrably reduce rework cost. Business case BC-01: *"Reducing sprint reopen rate from 38% to <10% recovers ~3.4 developer-days per sprint per team."* This is the ROI hook that justifies the retainer.

1.5.2 EDGE CASES

- Intake submitted with **no measurable outcome** → system flags as "unverifiable" and blocks promotion (enforces ISO 29148 *verifiable* criterion).
- Stakeholder retracts/changes the need mid-sprint → RTM versioning records the change and impacted stories.
- SLA timer crosses a non-business-hours boundary (Advisory tier) → clock pauses per contract.

1.5.3 USE CASES

UC-01 Submit & Triage: Client analyst submits a need → SLA timer starts → HighTower architect acknowledges within 15 min → triages into discovery gate 1.2.1.

1.5.4 'TECHNICAL REQUIREMENTS' CARD

TECH REQ CARD · TR-CD-01

Title: Unverifiable-requirement guardrail **Type:** Functional + Quality (Verifiability) **Source:** §1.1.5, §1.5.2 **Statement:** *The system shall reject promotion of any requirement that lacks a measurable, testable acceptance criterion.* **Acceptance (Gherkin):**

```
Feature: Verifiability guardrail
  Scenario: Block unverifiable requirement
    Given an intake item with no measurable acceptance criterion
    When an analyst attempts to promote it to the backlog
    Then the system rejects promotion
    And displays "Add a measurable acceptance criterion to continue"
```

Priority: MUST · **Verification:** Automated test + review gate

1.5.5 TECH REQUIREMENTS INDEX + PRIORITY

ID	Requirement	Priority	Method
TR-CD-01	Unverifiable-requirement guardrail	MUST	Scrum/XP test-first
TR-CD-02	15-min SLA acknowledgment	MUST	Lean flow
TR-CD-03	RTM auto-generation	SHOULD	Scrum
TR-CD-04	Jira bidirectional sync	SHOULD	XP CI
TR-CD-05	Discovery analytics dashboard	COULD	Lean metrics

Appendix D — Glossary (Ubiquitous Language)

Term	Definition
Trouble-Ticket	Canonical term for a submitted requirement/incident entering the SLA pipeline.
Developer-Ready Spec	A specification with API schema, data model, acceptance criteria, and edge cases — zero open interpretation.
SLA Tier	Advisory (4 h), Partner (1 h), Emergency Owner (15 min) response thresholds.
RTM	Requirements Traceability Matrix linking need → story → test.
NFR	Non-Functional Requirement (performance, scalability, reliability, security, etc.).
ADR	Architecture Decision Record — context, decision, consequences.
Bounded Context	DDD boundary within which a domain model and its terms are consistent.
INVEST	Independent, Negotiable, Valuable, Estimable, Small, Testable.
WSJF	Weighted Shortest Job First — Cost of Delay ÷ Job Size.

Appendix E — References

1. ISO/IEC/IEEE 29148:2018 — *Systems and software engineering — Life cycle processes — Requirements engineering.* IEEE SA. https://standards.ieee.org/standard/29148-2018.html
2. ReqView — *ISO/IEC/IEEE 29148 Requirements Specification Templates (StRS/SyRS/SRS/BRS/OpsCon).* https://www.reqview.com/doc/iso-iec-ieee-29148-templates
3. Schwaber, K. & Sutherland, J. — *The 2020 Scrum Guide.* https://scrumguides.org/scrum-guide.html
4. IIBA — *A Guide to the Business Analysis Body of Knowledge (BABOK® Guide v3).* https://www.iiba.org/
5. Blank, S. — *Jobs-To-Be-Done as the Front End of Customer Discovery.* https://steveblank.com/2021/11/04/market-definition-its-the-front-end-of-customer-discovery/
6. Ulwick, T. — *Jobs to Be Done (JTBD): The Original Framework.* Strategyn. https://strategyn.com/jobs-to-be-done/
7. Fitzpatrick, R. — *The Mom Test* (customer discovery interview method).
8. AltexSoft — *Nonfunctional Requirements: Examples, Types and Best Practices.* https://www.altexsoft.com/blog/non-functional-requirements/

9. TestQuality — *Gherkin User Stories & Acceptance Criteria Guide (2026).* https://testquality.com/gherkin-user-stories-acceptance-criteria-guide
10. Boost / PlatinumEdge — *INVEST Criteria for Agile User Stories.* https://www.boost.co.nz/blog/2021/10/invest-criteria
11. Architecture Decision Records. https://adr.github.io/
12. KanbanZone — *Agile Frameworks Compared: Scrum, Kanban, Lean, XP (2024).* https://kanbanzone.com/2024/agile-frameworks-compared/
13. PPM Express — *From RICE to WSJF: Prioritization Techniques.* https://www.ppm.express/blog/13-prioritization-techniques
14. GetProductPeople — *Prioritization Techniques: RICE, MoSCoW, ICE & Kano.* https://www.getproductpeople.com/blog/prioritization-techniques-rice-moscow-ice-kano
15. HighTower.Digital — *MSP SLA #1+5 — IT Requirements As-A-Service.* https://www.hightower.digital/msp-sla-15

Document control. HDI Confidential · Author: LW Atkinson (CEO/CTO) · Version R1.A · 06/25/2026.