Product Requirements Document

Component: SpecifAI Coding-Agent (Backend Automation)

Version: 0.3 -- July 2025

---

1 - Purpose & Vision

The **Coding-Agent** turns every accepted issue coming from the SpecifAI voice app into an **autonomous delivery pipeline**:

1. **Listen** for new or updated issues.
2. **Clarify** missing details via comments.
3. **Implement** the feature (code-gen, tests, PR).
4. **Notify** all stakeholders in real time and by e-mail.

Outcome: zero hand-offs, faster cycle-time, full traceability.

---

2 - Scope & Objectives (MVP)

Objective	KPI / Target
Detect ready issues within 1 min	Poll + webhook latency <= 60 s
Auto-label workflow states	100% of picked issues
Clarification turnaround	Comment posted <= 30 s after detection
Code-gen success rate	>= 80% PRs pass CI on first run
Status push to Supabase	<= 3 s end-to-end

---

3 - Functional Requirements

3.1 Issue Intake

- **Trigger 1: Cron Poll** -- every **10 min** call GitHub GraphQL for issues with label `ready-for-dev` AND assigned to `@specifai-agent`.
- **Trigger 2: Webhook** -- listen `issues.opened`, `issues.labeled`, `issues.assigned`.
- Store candidate issue IDs in Supabase table `agent_queue`.

3.2 Clarification Bot

- Validate presence of: *User Story*, *Acceptance Criteria*, *Edge-Cases*.
- If missing -> comment:

"Need more info on ___. Please update the issue; I will resume automatically."

- Apply label `needs-info` and pause processing.

3.3 Implementation Phase

1. **Branch Creation** -- `feat/issue-` on default branch.

2. **Code Generation** -- call internal code LLM or template library.
3. **Commit & PR** -- push branch, open Draft PR referencing issue (`Fixes #ID`).
4. **Label Flow**
- `in-progress` once branch pushed.
- `in-review` when PR CI passes.
- `done` when PR merged; auto-close issue.

3.4 Notifications

- **Realtime** -- publish JSON to Supabase channel `issue:`.
- **Email** -- SendGrid to PO & Dev Lead on state changes.

3.5 Error Handling

Error	Action
Code gen fails tests	Comment log excerpt; label `blocked`.
GitHub rate limit	Back-off; queue persists.
Clarification idle > 48 h	Unassign agent; email reminder.

---

4 - Non-Functional Requirements

Category	Requirement
Scalability	Parallel up to 10 issues (async workers)
Security	Token scope limited; secrets in Vault
Observability	Prometheus metrics incl. queue length
Reliability	99% polling success; retry 3x
Cost	Cloud Function (0.5 vCPU) + LLM; ~€100/mo

---

5 - Technical Architecture

```
graph TD
  subgraph GitHub
    ISS(Issues) --> WH(Events)
  end
  subgraph Supabase
    Q[agent_queue]
    RT[Realtime]
  end
  subgraph Cloud
    CRON(Cron 10 min)
    AGENT(Worker Pool\nPython/FastAPI)
    LLM(Code-Gen)
    CI(CI Pipeline)
    EMAIL(SendGrid)
  end
  CRON --> AGENT
  WH --> AGENT
  AGENT <-->|GraphQL| ISS
  AGENT -->|Insert| Q
  AGENT --> LLM
  AGENT -->|PR| CI
  AGENT -->|Publish| RT
  AGENT --> EMAIL
```

---

6 - Data Model (Supabase)

```
create table agent_queue (
  id bigserial primary key,
  github_issue bigint unique,
  status text check (status in ('queued','needs-info','in-progress','blocked','done')),
  last_check timestamptz default now(),
  error_log text
);
```

---

7 - Workflow State Machine

State	Trigger	Next State
queued	all mandatory fields present	in-progress
queued	missing data	needs-info
needs-info	issue edited & complete	queued
in-progress	PR opened & CI green	in-review
in-review	PR merged	done
blocked	manual intervention	queued

---

8 - Roadmap

Milestone	Content	ETA
A0	Poll + queue logging	Week 1
A1	Clarification bot	Week 2
A2	Code-gen POC + PR	Week 4
A3	Realtime + emails	Week 6
A4	Observability	Week 7

---

9 - Risks & Mitigations

Risk	Mitigation
LLM hallucinated code	Unit tests, strict templates
Queue flooding	Concurrency limit, priority order
GitHub API changes	SDK abstraction, version pin

---

10 - Open Questions

1. Auto-merge PRs vs human review?
2. Max runtime per job (30 min?)
3. Fine-tune LLM on repo data?

Prepared for SpecifAI Core Engineering - July 2025