Empowering Learners to Build Production - Ready AI Agents

Applied Agentic AI for EMs

Capstone Project 1

Multi-Agent BRD-to-Engineering System Generator

Problem Statement

Empowering Engineering Managers with Agentic Tools for Efficient Planning and Technical Design

1. Business Use Case

Engineering Managers (EMs) often face a bottleneck in translating complex Business Requirement Documents (BRDs) into structured technical plans and architecture. This gap results in delays, misalignment, and inconsistent project scoping.

The Opportunity:

- **Faster Turnaround from BRD to Execution:** An AI-driven system that reduces time spent on manual parsing, clarifying assumptions, and drafting artifacts.
- **Standardized Planning and Design:** Consistent, AI-generated outputs that align with org-level engineering templates.
- **Interactive Engagement:** Conversational agents that gather required inputs (resourcing, tech stack, milestones) from EMs directly.
- **Decision Intelligence:** Helps EMs quickly weigh trade-offs (e.g., tech stack options, PoC requirements, resourcing impact).

---

2. Technical Architecture

Component	Technology	Purpose
Multi-Agent System	**n8n** (agent orchestration)	Manages modular agents through low-code workflows
LLM Backend	OpenAI GPT / Claude	Natural language understanding and content generation
BRD Input Parser	PDF-to-Text, Markdown	Initial intake and classification of requirement sections
Prompt Engineering	Prompt templates with few-shot examples	Guides agents with task-specific formatting, tone, and structure cues
State Management	n8n Context	Maintain session info, responses, and dependencies
Frontend Interface	Streamlit / Gradio	Review and edit outputs from planning/design agents
Guardrails	Custom validation + Regex filters / LangChain Guardrails	Enforces structure, quality checks, and security constraints on output

3. Agent Architecture

A. Planning Agent

- **a) Eng Plan Generator:** Conversational bot to identify delivery phases, team composition, interdependencies, etc.
- **b) Schedule Estimator:** Generates high-level effort estimate, project timeline, and resource allocation matrix using heuristics (e.g., ideal team size, velocity).

B. Design Agent

- **a) Solution Architect:** Outputs high-level system design mapped to functional requirements.
- **b) PoC Planner:** Outlines scope of a testable Proof-of-Concept for feasibility, including components and goals.
- **c) Tech Stack Recommender:** Suggests 2–3 stack configurations based on scalability, internal capability, and integration needs.

---

4. Data Flow

Input: BRD Upload
→ Section Parsing (Functional, Non-functional, Dependencies)
→ Trigger Relevant Agents
→ RAG for referencing templates
→ Output Review & Finalization via UI
→ Export (PDF, Confluence, CSV)

---

5. Deliverables/Objectives

- **Functional System in n8n** with agent routing and fallback handling
- **Planning Outputs:**
    - Structured Engineering Plan (phases, risks, team needs)
    - Project schedule (high-level estimates with justifications)
- **Design Outputs:**
    - High-Level Architecture (logical & system boundaries)
    - PoC document (objectives, criteria, modular scope)
    - Tech stack matrix (trade-offs, compatibility, risk factors)
- **User Interface for EMs:** Review, modify, approve, and export artifacts
- **Test Coverage:** 80%+ coverage using mock BRDs

---

6. Learning Goals for Engineering Managers

System Design & Decomposition

- Learn to break down BRDs into discrete planning and design components
- Apply AI workflows to streamline documentation tasks

AI-Driven Engineering Ops

- Understand conversational agents for scoped data gathering
- Use LLM + RAG to surface org-specific templates and guides

Leadership Enablement

- Produce stakeholder-ready artifacts in minutes
- Run cost-time-effort analysis scenarios with agentic tools

---

7. Submission Guidelines

Working Prototype Must Include:

- n8n flow (modular, easy to debug)
- All 5 agent functions working with sample BRD
- Live editable UI with approval flow

Demo Video (5–10 min):

- Upload BRD → Agent outputs
- Walkthrough of each agent's response
- UI review and export feature

Agent Workflow Based Structure:

```
brd_agent_em/
├── n8n_flows/
│   ├── planning/
│   │   ├── eng_plan_generator.json
│   │   └── schedule_estimator.json
│   ├── design/
│   │   ├── solution_architect.json
│   │   ├── poc_planner.json
│   │   └── tech_stack_recommender.json
│
├── brd_parser/
│   └── brd_input_cleaner.json
│
├── shared_nodes/
│   ├── context_handler.json
│   └── fallback_handler.json
│
├── frontend/
│   └── web_ui_integration.md
│
├── sample_inputs/
```

```
│   └── sample_brd.md
│
├── tests/
│   └── test_plan_flow.md
│
├── config.yaml
└── README.md
```

Documentation Includes:

- Setup instructions
- Architecture diagram
- Sample inputs/outputs
- Agent protocols and decision logic
- Known limitations

---

8. Evaluation Criteria

Area	Weight
Multi-Agent Architecture	25%
Relevance & Detail of Plans	15%
Schedule Estimation Accuracy	10%
Architecture & PoC Outputs	15%
Tech Stack Trade-offs	10%
UI/UX Experience for EMs	10%
Code Modularity & Docs	15%

---

9. Future Directions

- **Jira/ClickUp Export:** Convert output directly into user stories/tasks
- **Slack Integration:** Agent responses via command-line prompts
- **Cross-team Collaboration:** Multi-EM workflows and shared artifact editing
- **Infra Planning Agent:** Add CI/CD, security, and monitoring setup templates
- **Governance Layer:** Add role-based approvals and history tracking

---