LLM Design Patterns — Study Plan & User Story Template

Template User Stories LLM Projects

Contents

How to Use This Template

- **ID:** Stable, unique identifier (e.g., LLMDP-CH26-01).
- **Title:** Action-oriented, outcome-focused (e.g., "Build a minimal RAG pipeline").
- **Epic / Feature:** Chapter-level capability (e.g., "Ch26: RAG Patterns" → "RAG Quick-start").
- **Business Value:** Why this matters (e.g., reduces hallucinations on internal docs).
- **Priority / Estimate:** MoSCoW or P1–P4, and rough SP.
- **Persona:** Learner / Engineer / Reviewer / Sponsor.
- **Dependencies:** Pre-req chapters, tooling, data.
- **Assumptions / Risks:** Known constraints, open questions.
- **Acceptance Criteria:** Given/When/Then, objective and testable.
- **Evidence:** Links to artifacts (code, reports, dashboards).

User Story Template

Story Card Definition

Definition of Ready. **Definition of Ready:** Persona clear; Acceptance template drafted; Dependencies known; Estimate set.

Definition of Done. **Definition of Done:** Tests pass; Acceptance criteria met; Quality checks; Docs updated; Deployed or feature flagged.

Guidelines.

- **INVEST:** Independent, Negotiable, Valuable, Estimable, Small, Testable.
- **Format:** *As a [persona], I want [capability], so that [outcome].*
- **Acceptance (BDD):** Use 3–6 clear Given/When/Then criteria.
- **Evidence:** Prefer measurable artifacts (metrics, dashboards, traces).

User Story Template

ID: `LLMDP-XXX`
Title: *Short, imperative headline*
Epic / Feature: *Chapter / Capability*
Business Value: *Why this matters*
Priority / Estimate: *P1–P4, N story points*
Persona: *Learner / Engineer / Reviewer / Sponsor*
Dependencies: *Prereqs and tooling*
Assumptions / Risks: *Constraints, open questions*

User Story

☐ As a [**persona**], I want [**capability**] so that [**outcome**].

Acceptance Criteria (BDD)

☐ **Given** [precondition], **when** [action], **then** [result].
☐ **Given** [another precondition], **when** [action], **then** [result].

Evidence to Attach

Code/notebook link ; [metrics.json]; [dashboard/screenshot].
ks, notes, and follow-ups .

Example Cards

CH26 Example — Minimal RAG Quickstart

ID: LLMDP-CH26-01
Title: Build a minimal RAG pipeline
Epic / Feature: Chapter 26 — RAG Patterns / Quickstart
Business Value: Enable retrieval-grounded answers to reduce hallucination rate on internal docs.
Priority / Estimate: P1, 5 SP
Persona: Learner
Dependencies: Document corpus; embedding model; vector store.
Assumptions / Risks: Cost of embeddings; evaluation data availability.

User Story

☐ As a learner, I want to build a minimal RAG pipeline (ingest, index, retrieve, answer) so that I can baseline grounded answering on a small internal corpus.

Acceptance Criteria (BDD)

- ☐ **Given** a seed corpus and 10 evaluation questions, **when** I run the baseline pipeline, **then** I produce answers and store traces.
- ☐ **Given** the same questions, **when** I compare RAG vs. no-RAG on exact-match, **then** RAG improves accuracy by $\geq 20\%$.
- ☐ **Given** the index is rebuilt, **when** I rerun evaluation, **then** results are reproducible within 2% variance.

Evidence to Attach

- ☐ Notebook and CLI for ingest/index/query; config in repo.
- ☐ Evaluation report (`metrics.json`) with factuality, precision/recall, latency, cost.
- ☐ README shows setup, commands, and limitations.

CH17 Example — Chain-of-Thought and Self-Consistency

ID: LLMDP-CH17-02
Title: Evaluate CoT + self-consistency on a math/logic set
Epic / Feature: Chapter 17 — Prompting Patterns / Reasoning
Business Value: Quantify accuracy/cost trade-offs for CoT and sampling strategies.
Priority / Estimate: P2, 8 SP
Persona: QA reviewer
Dependencies: Prompting framework; evaluation set; sampling controls.
Assumptions / Risks: Longer latency and sampling cost; leakage risk if traces are exposed.

User Story

- ☐ As a QA reviewer, I want CoT and self-consistency variants in evaluation so that we can decide whether to enable them by default.

Acceptance Criteria (BDD)

- ☐ **Given** a 20-task set, **when** I enable CoT with $k = 1$, **then** accuracy improves by $\geq 8\%$ vs. baseline at $\leq 2\times$ latency.
- ☐ **Given** the same set, **when** I run single-shot vs. self-consistency with $k = 5$, **then** I report accuracy and cost-per-correct with plots.
- ☐ **Given** sensitive tasks, **when** I log traces, **then** PII is redacted and storage complies with policy.

Evidence to Attach

- ☐ Experiment configs checked in; results tracked in `metrics.json`.
- ☐ Plots in `reports/` and a short `cot_findings.md`.

Chapter User Stories

CH01 — Pattern Selection Rubric & Success Criteria

ID: LLMDP-CH01-01
Title: Create an LLM pattern selection rubric with KPIs
Epic / Feature: Chapter 1 — Introduction / Pattern Mindset
Business Value: Aligns solutions to constraints (cost, latency, safety) and reduces rework.
Priority / Estimate: P1, 3 SP
Persona: Architect / Tech Lead
Dependencies: None; access to product goals and constraints.
Assumptions / Risks: Stakeholder alignment required; scope creep risk.

User Story

☐ As an architect, I want a rubric mapping use cases to LLM patterns so that the team can consistently choose low-risk, high-value approaches.

Acceptance Criteria (BDD)

☐ **Given** top 5 use cases, **when** I score each against constraints (cost, latency, safety, accuracy), **then** the rubric recommends 1–2 patterns with rationale.

☐ **Given** a new use case, **when** I apply the rubric, **then** I can produce KPIs (quality, cost/request, p95 latency) and a validation plan.

☐ **Given** conflicting constraints, **when** I document trade-offs, **then** stakeholders sign off on chosen pattern.

Evidence to Attach

☐ `patterns_rubric.md`; `kpi_matrix.csv`; architecture decision record (ADR).

☐ Slide summary with decision tree.

CH02 — Data Cleaning Pipeline

ID: LLMDP-CH02-01
Title: Implement reproducible data cleaning (dedup, lang, PII)
Epic / Feature: Chapter 2 — Data Cleaning
Business Value: Improves training/eval quality; reduces safety risk and waste.
Priority / Estimate: P1, 5 SP
Persona: Data Engineer
Dependencies: Raw corpus; basic infra for ETL.
Assumptions / Risks: PII detection coverage; false positives.

User Story

☐ As a data engineer, I want a cleaning pipeline with deduplication, language detection, and PII scrubbing so that downstream training/eval is trustworthy.

Acceptance Criteria (BDD)

- ☐ **Given** an input corpus, **when** I run the pipeline, **then** duplicates are reduced by $\geq 95\%$ and a dedup report is generated.
- ☐ **Given** multilingual content, **when** language filtering is enabled, **then** non-target languages are removed with $\geq 98\%$ precision.
- ☐ **Given** PII patterns, **when** I scan and mask, **then** a PII audit log is produced with no unmasked samples in a 1k spot-check.

Evidence to Attach

- ☐ `cleaning_pipeline.ipynb`, configs; `cleaning_report.md`.
- ☐ Sample before/after stats; CI job log.

CH03 — Data Augmentation

ID: LLMDP-CH03-01
Title: Augment training data without semantic drift
Epic / Feature: Chapter 3 — Augmentation Patterns
Business Value: Improves coverage and robustness with minimal label cost.
Priority / Estimate: P2, 5 SP
Persona: ML Engineer
Dependencies: Clean dataset; augmentation tools.
Assumptions / Risks: Semantic drift; distribution shift.

User Story

- ☐ As an ML engineer, I want back-translation and paraphrase augmentation so that I increase coverage while avoiding meaning drift.

Acceptance Criteria (BDD)

- ☐ **Given** 5k samples, **when** I augment, **then** embedding-similarity medians stay ≥ 0.9 vs. originals.
- ☐ **Given** baseline performance, **when** I train with augmented data, **then** held-out accuracy improves by $\geq 3\%$.
- ☐ **Given** cost constraints, **when** I run augmentation, **then** I produce a cost/time report and data cards.

Evidence to Attach

- ☐ `augment_eval.md`, similarity histograms; MLflow runs.
- ☐ Data card JSON; PR with configs.

CH04 — Handling Large Datasets

ID: LLMDP-CH04-01
Title: Streamable sharded dataset with throughput profiling
Epic / Feature: Chapter 4 — Data at Scale
Business Value: Enables fast training without memory bottlenecks.

Priority / Estimate: P2, 3 SP
Persona: ML Engineer
Dependencies: Object storage; parquet/arrow tooling.
Assumptions / Risks: I/O hotspots; shard imbalance.

User Story

☐ As an ML engineer, I want sharded parquet with a streaming loader so that I can train efficiently on large corpora.

Acceptance Criteria (BDD)

☐ **Given** a 200GB corpus, **when** I stream batches, **then** GPU utilization stays $\geq 80\%$ on average.

☐ **Given** multiple workers, **when** I read shards, **then** imbalance is $\leq 10\%$ and no worker starves.

☐ **Given** profiles, **when** I run a 10-minute soak, **then** I/O throughput and p95 latency are reported.

Evidence to Attach

☐ Loader script; profiler screenshots; throughput CSV.

☐ README with tuning tips.

CH05 — Data Versioning

ID: LLMDP-CH05-01
Title: Track dataset lineage with DVC and run linkage
Epic / Feature: Chapter 5 — Versioning Patterns
Business Value: Enables reproducibility and auditability.
Priority / Estimate: P1, 3 SP
Persona: MLOps Engineer
Dependencies: Git repo; object store; CI.
Assumptions / Risks: Storage costs; remote access.

User Story

☐ As an MLOps engineer, I want DVC-tracked datasets linked to experiment runs so that I can reproduce any model with exact data.

Acceptance Criteria (BDD)

☐ **Given** dataset v1, **when** I run training, **then** MLflow stores the DVC hash with the run.

☐ **Given** dataset v2, **when** I diff, **then** I see added/removed records and risk notes.

☐ **Given** an audit, **when** I checkout a past run, **then** I can reproduce metrics within 1%.

Evidence to Attach

☐ `dvc.yaml`; MLflow link; lineage diagram.

☐ ADR documenting retention policy.

CH06 — Annotation & Labeling

ID: LLMDP-CH06-01
Title: Set up labeling guidelines with QA and IAA
Epic / Feature: Chapter 6 — Annotation Patterns
Business Value: Higher label quality; less noise.
Priority / Estimate: P2, 5 SP
Persona: Data PM / Label Lead
Dependencies: Labeling tool; sampling strategy.
Assumptions / Risks: Annotator drift; budget.

User Story

☐ As a data PM, I want clear guidelines and QA sampling so that labels are consistent and reproducible.

Acceptance Criteria (BDD)

☐ **Given** two annotators, **when** I compute agreement, **then** Cohen's $\kappa \geq 0.75$ on a 200-sample set.

☐ **Given** QA rules, **when** I review 10% of labels, **then** correction rate is $< 5\%$ after iteration 2.

☐ **Given** guideline updates, **when** I re-run a small batch, **then** agreement improves vs. baseline.

Evidence to Attach

☐ `label_guidelines.md`; QA dashboard; IAA report.

☐ Example labeled JSONL before/after.

CH07 — Training Pipeline

ID: LLMDP-CH07-01
Title: Config-driven modular training with retries
Epic / Feature: Chapter 7 — Training Pipeline
Business Value: Reproducible training; fewer failed runs.
Priority / Estimate: P1, 5 SP
Persona: ML Engineer
Dependencies: Hydra/Lightning or equivalent; MLflow.
Assumptions / Risks: Preemptible nodes; spot failures.

User Story

☐ As an ML engineer, I want a modular pipeline with resume/retry so that long runs survive infra hiccups and are repeatable from config.

Acceptance Criteria (BDD)

- ☐ **Given** a training config, **when** I kill the job mid-epoch, **then** resume restarts within $\leq$ 2 minutes and continues deterministically.
- ☐ **Given** a failed run, **when** retry policy triggers, **then** the job restarts up to N times and logs structured reasons.
- ☐ **Given** a new seed, **when** I re-run, **then** metrics vary within expected CIs.

Evidence to Attach

- ☐ `config.yaml`; `pipeline/` code; MLflow runs.
- ☐ Failure-injection notes and logs.

CH08 — Hyperparameter Tuning

ID: LLMDP-CH08-01
Title: Run budgeted HPO with early stopping
Epic / Feature: Chapter 8 — HPO Patterns
Business Value: Improves quality under cost constraints.
Priority / Estimate: P2, 5 SP
Persona: ML Engineer
Dependencies: Optuna/W&B; search space defined.
Assumptions / Risks: Overfitting to val set; budget limits.

User Story

- ☐ As an ML engineer, I want random+Bayesian HPO with early stopping so that I improve accuracy without exceeding budget.

Acceptance Criteria (BDD)

- ☐ **Given** a $X budget, **when** I run 50 trials with pruning, **then** best validation metric improves $\geq$ 3%.
- ☐ **Given** multi-objective metrics, **when** I analyze Pareto front, **then** I select a config with balanced cost/quality.
- ☐ **Given** a reproducibility check, **when** I re-run top-3, **then** results stay within 1% variance.

Evidence to Attach

- ☐ HPO report; plots; selected config.
- ☐ Cost breakdown and sweep artifacts.

CH09 — Regularization

ID: LLMDP-CH09-01
Title: Reduce overfit via dropout, weight decay, clipping
Epic / Feature: Chapter 9 — Regularization Patterns
Business Value: Improves generalization and stability.
Priority / Estimate: P3, 3 SP
Persona: ML Engineer

Dependencies: Baseline model; eval suite.
Assumptions / Risks: Underfitting if overly aggressive.

User Story

☐ As an ML engineer, I want to tune regularization knobs so that the model generalizes better without hurting accuracy.

Acceptance Criteria (BDD)

☐ **Given** a baseline, **when** I apply dropout/decay/clipping ablations, **then** held-out improves $\geq 2\%$ without latency penalty $> 5\%$.

☐ **Given** instability, **when** I enable gradient clipping, **then** loss spikes disappear (plots attached).

☐ **Given** 3 seeds, **when** I evaluate, **then** dispersion narrows vs. baseline.

Evidence to Attach

☐ Ablation notebook; metrics table; training curves.

CH10 — Checkpointing & Recovery

ID: LLMDP-CH10-01
Title: Design checkpoint schedule with retention policy
Epic / Feature: Chapter 10 — Reliability Patterns
Business Value: Reduces lost compute; enables audit.
Priority / Estimate: P1, 3 SP
Persona: MLOps Engineer
Dependencies: Storage; scheduler; pipeline hooks.
Assumptions / Risks: Storage quotas; restore failures.

User Story

☐ As an MLOps engineer, I want structured checkpoints and restore tests so that long runs can be resumed safely.

Acceptance Criteria (BDD)

☐ **Given** policy N_keep=3, **when** training runs, **then** only the last 3 checkpoints remain and older ones are GC'd.

☐ **Given** failure injection, **when** I restore, **then** training resumes within 2 minutes and metrics match within 0.5%.

☐ **Given** a compliance review, **when** I list artifacts, **then** hashes and metadata are present.

Evidence to Attach

☐ Checkpoint policy doc; restore logs; checksum table.

CH11 — Fine-Tuning

ID: LLMDP-CH11-01
Title: Domain adaptation via LoRA/QLoRA baseline
Epic / Feature: Chapter 11 — Fine-Tuning Patterns
Business Value: Improves domain performance at lower cost.
Priority / Estimate: P1, 5 SP
Persona: ML Engineer
Dependencies: Base model; domain corpus; eval set.
Assumptions / Risks: Catastrophic forgetting risk.

User Story

☐ As an ML engineer, I want a LoRA/QLoRA fine-tune so that domain metrics improve with modest hardware.

Acceptance Criteria (BDD)

☐ **Given** domain dataset, **when** I fine-tune, **then** task metric improves $\geq 5\%$ vs. SFT-free baseline.

☐ **Given** generic eval set, **when** I test, **then** no major regressions ($< 2\%$ absolute).

☐ **Given** cost limits, **when** I log tokens and time, **then** run stays within budget.

Evidence to Attach

☐ Fine-tune notebook; config; before/after scores.

CH12 — Pruning

ID: LLMDP-CH12-01
Title: Structured pruning for smaller, faster inference
Epic / Feature: Chapter 12 — Model Compression
Business Value: Cuts latency and memory without large quality loss.
Priority / Estimate: P3, 3 SP
Persona: ML Engineer
Dependencies: Baseline model; pruning toolkit.
Assumptions / Risks: Accuracy drop; hardware quirks.

User Story

☐ As an ML engineer, I want to prune 10–30% of parameters so that I improve throughput with minimal loss.

Acceptance Criteria (BDD)

☐ **Given** a pruned model, **when** I benchmark, **then** latency improves $\geq 20\%$ at p95 with $< 2\%$ accuracy drop.

☐ **Given** traffic mix, **when** I load-test, **then** throughput increases proportionally and remains stable for 30 minutes.

☐ **Given** compatibility tests, **when** I deploy, **then** no unsupported ops surface.

Evidence to Attach

☐ Benchmark CSV; plots; pruning config.

CH13 — Quantization

ID: LLMDP-CH13-01
Title: Compare 8-bit vs 4-bit quantization
Epic / Feature: Chapter 13 — Quantization Patterns
Business Value: Reduces memory; enables edge or CPU inference.
Priority / Estimate: P2, 4 SP
Persona: ML Engineer
Dependencies: AWQ/GPTQ/AutoGPTQ; eval harness.
Assumptions / Risks: Quality loss on long-form tasks.

User Story

☐ As an ML engineer, I want to evaluate 8-bit vs 4-bit to pick the best memory/quality trade-off for our serving stack.

Acceptance Criteria (BDD)

☐ **Given** baseline FP16, **when** I quantize, **then** memory drops $\geq 50\%$ (8b) and $\geq 70\%$ (4b).

☐ **Given** task set, **when** I evaluate, **then** 8b quality loss $\leq 1\%$ and 4b $\leq 3\%$.

☐ **Given** latency targets, **when** I measure p95, **then** latency improves $\geq 20\%$.

Evidence to Attach

☐ Metrics table; hardware notes; deployment checklist.

CH14 — Evaluation Metrics

ID: LLMDP-CH14-01
Title: Build task-aligned metrics and eval suite
Epic / Feature: Chapter 14 — Evaluation Patterns
Business Value: Honest progress measurement; regression detection.
Priority / Estimate: P1, 4 SP
Persona: QA Engineer
Dependencies: Datasets; metric scripts; dashboarding.
Assumptions / Risks: Metric gaming risk.

User Story

☐ As a QA engineer, I want a standard eval suite so that we can compare models fairly across tasks and time.

Acceptance Criteria (BDD)

- ☐ **Given** 3 core tasks, **when** I run the suite, **then** metrics are logged with CI status and trend charts.
- ☐ **Given** a new model, **when** I evaluate, **then** baseline deltas are computed with significance where possible.
- ☐ **Given** a regression, **when** thresholds trip, **then** CI fails with a link to diffs.

Evidence to Attach

- ☐ Eval scripts; dashboard URL; threshold policy doc.

CH15 — Cross-Validation

ID: LLMDP-CH15-01
Title: K-fold protocol across domain slices
Epic / Feature: Chapter 15 — Generalization Patterns
Business Value: Reduces overfitting to narrow distributions.
Priority / Estimate: P2, 3 SP
Persona: QA Engineer
Dependencies: Dataset partitions; eval harness.
Assumptions / Risks: Leakage risk if split poorly.

User Story

- ☐ As a QA engineer, I want k-fold CV by domain so that we estimate true generalization and spot overfit.

Acceptance Criteria (BDD)

- ☐ **Given** four domains, **when** I split, **then** no sample appears in both train and test folds.
- ☐ **Given** CV runs, **when** I summarize, **then** mean±std is reported with confidence intervals.
- ☐ **Given** variability, **when** I analyze, **then** action items are filed for high-variance domains.

Evidence to Attach

- ☐ Split manifest; metrics report; leakage checks.

CH16 — Interpretability

ID: LLMDP-CH16-01
Title: Run attribution/probing on selected tasks
Epic / Feature: Chapter 16 — Interpretability Patterns
Business Value: Increases trust and debugging speed.
Priority / Estimate: P3, 4 SP
Persona: Research Engineer
Dependencies: Probing tools; datasets.
Assumptions / Risks: Misinterpretation of probes.

User Story

☐ As a research engineer, I want attribution and probes so that we can explain behaviors and target fixes.

Acceptance Criteria (BDD)

☐ **Given** failing items, **when** I run attribution, **then** salient tokens and layers are identified with visuals.

☐ **Given** hypotheses, **when** I run probes, **then** results support/contradict with metrics.

☐ **Given** insights, **when** I propose changes, **then** follow-up tickets are created with expected impact.

Evidence to Attach

☐ Notebooks; plots; summary memo.

CH17 — Fairness & Bias

ID: LLMDP-CH17-01
Title: Detect and mitigate bias across subgroups
Epic / Feature: Chapter 17 — Fairness Patterns
Business Value: Reduces harm and compliance risk.
Priority / Estimate: P1, 5 SP
Persona: Responsible AI Lead
Dependencies: Fairness metrics; subgroup labels where lawful.
Assumptions / Risks: Sensitive data handling.

User Story

☐ As an RAI lead, I want subgroup tests and mitigations so that we reduce disparate performance across groups.

Acceptance Criteria (BDD)

☐ **Given** subgroup data, **when** I evaluate, **then** gaps >= 5% are flagged with confidence.

☐ **Given** a mitigation, **when** I re-evaluate, **then** gap reduces by $\geq 50\%$ without global regression $> 2\%$.

☐ **Given** governance, **when** I log, **then** review artifacts are archived per policy.

Evidence to Attach

☐ Fairness report; mitigation PR; governance ticket.

CH18 — Adversarial Robustness

ID: LLMDP-CH18-01
Title: Build a prompt-attack harness and hardening plan
Epic / Feature: Chapter 18 — Robustness Patterns
Business Value: Protects against jailbreaks and abuse.

Priority / Estimate: P1, 6 SP
Persona: Security Engineer
Dependencies: Safety policies; red-team prompts.
Assumptions / Risks: Evolving threats.

User Story

☐ As a security engineer, I want adversarial tests and mitigations so that the model resists common jailbreaks.

Acceptance Criteria (BDD)

☐ **Given** a curated attack set, **when** I evaluate, **then** bypass rate is reported with severity tags.

☐ **Given** guardrails, **when** I harden, **then** bypass rate reduces by $\geq 50\%$ with minimal false positives.

☐ **Given** recurring tests, **when** CI runs weekly, **then** regressions are flagged.

Evidence to Attach

☐ Attack harness; reports; mitigation PRs.

CH19 — RLHF & Preference Optimization

ID: LLMDP-CH19-01
Title: Small-scale RM + PPO vs SFT baseline
Epic / Feature: Chapter 19 — Alignment Patterns
Business Value: Aligns outputs to user preferences.
Priority / Estimate: P2, 8 SP
Persona: Research Engineer
Dependencies: Preference data; compute; safety checks.
Assumptions / Risks: Instability; reward hacking.

User Story

☐ As a research engineer, I want a small reward model and PPO fine-tune so that outputs better match preferences.

Acceptance Criteria (BDD)

☐ **Given** preference pairs, **when** I train RM, **then** validation accuracy $\geq 65\%$.

☐ **Given** PPO, **when** I evaluate, **then** user-preference win rate improves $\geq 10\%$ vs. SFT-only.

☐ **Given** safety checks, **when** I run tests, **then** toxic/unsafe rate does not increase.

Evidence to Attach

☐ RM/ PPO configs; win-rate report; safety logs.

CH20 — Chain-of-Thought Prompting

ID: LLMDP-CH20-01
Title: Add CoT templates with cost/latency controls
Epic / Feature: Chapter 20 — Reasoning Patterns
Business Value: Improves step-by-step reasoning quality.
Priority / Estimate: P2, 4 SP
Persona: Prompt Engineer
Dependencies: Evaluation harness; sampling control.
Assumptions / Risks: Token bloat; leakage of traces.

User Story

☐ As a prompt engineer, I want CoT templates and self-consistency so that hard tasks are solved more reliably.

Acceptance Criteria (BDD)

☐ **Given** a reasoning set, **when** I enable CoT, **then** accuracy improves $\geq 8\%$ with cost reported.

☐ **Given** k-sampling, **when** I sweep k, **then** I select a setting within latency SLO.

☐ **Given** logs, **when** I store traces, **then** PII and secrets are redacted.

Evidence to Attach

☐ Template library; results table; cost analysis.

CH21 — Tree-of-Thoughts

ID: LLMDP-CH21-01
Title: Implement beam search ToT with pruning heuristics
Epic / Feature: Chapter 21 — Structured Reasoning
Business Value: Explores solution paths and reduces dead-ends.
Priority / Estimate: P3, 5 SP
Persona: Prompt Engineer
Dependencies: Search framework; eval tasks.
Assumptions / Risks: Higher cost; complexity.

User Story

☐ As a prompt engineer, I want ToT with pruning so that complex tasks benefit from structured exploration.

Acceptance Criteria (BDD)

☐ **Given** a puzzle set, **when** I enable ToT, **then** accuracy improves vs. CoT-only with cost tracked.

☐ **Given** pruning, **when** I tune thresholds, **then** expansions drop $\geq 40\%$ with minimal accuracy loss.

☐ **Given** failures, **when** I analyze reasons, **then** I file tuning actions.

Evidence to Attach

☐ ToT notebook; search stats; comparison plots.

CH22 — ReAct (Reason+Act with Tools)

ID: LLMDP-CH22-01
Title: Build a ReAct agent using search + calculator tools
Epic / Feature: Chapter 22 — Tool-Augmented Reasoning
Business Value: Enables grounded answers and calculations.
Priority / Estimate: P2, 6 SP
Persona: Agent Engineer
Dependencies: Tool schemas; sandbox.
Assumptions / Risks: Tool failures; injection attacks.

User Story

☐ As an agent engineer, I want a ReAct loop with two tools so that the agent can plan, cite, and compute.

Acceptance Criteria (BDD)

☐ **Given** tool specs, **when** I run tasks, **then** the agent performs plan → act traces with citations.

☐ **Given** tool errors, **when** a call fails, **then** the agent retries or falls back gracefully.

☐ **Given** prompt injection tests, **when** I evaluate, **then** the agent declines unsafe tool calls.

Evidence to Attach

☐ Agent code; transcripts; red-team report.

CH23 — ReWOO (Plan-then-Act)

ID: LLMDP-CH23-01
Title: Implement ReWOO and compare to ReAct
Epic / Feature: Chapter 23 — Planning Patterns
Business Value: Reduces tool thrashing and errors.
Priority / Estimate: P3, 5 SP
Persona: Agent Engineer
Dependencies: Planner; tool executor.
Assumptions / Risks: Longer initial planning; drift risk.

User Story

☐ As an agent engineer, I want a ReWOO pipeline so that the agent plans before calling tools and we can compare with ReAct.

Acceptance Criteria (BDD)

- ☐ **Given** a multi-hop task, **when** I run ReWOO, **then** the plan is explicit and tool calls follow the plan.
- ☐ **Given** the same tasks, **when** I compare with ReAct, **then** tool calls decrease by $\geq 20\%$ with equal or better accuracy.
- ☐ **Given** failure cases, **when** I analyze, **then** I identify plan quality issues and fixes.

Evidence to Attach

- ☐ Plans and traces; comparison report; tuning notes.

CH24 — Reflection Techniques

ID: LLMDP-CH24-01
Title: Add critique-and-retry loop to failing tasks
Epic / Feature: Chapter 24 — Self-Improvement Patterns
Business Value: Increases pass rate on hard cases.
Priority / Estimate: P2, 3 SP
Persona: Prompt Engineer
Dependencies: Eval harness; error taxonomy.
Assumptions / Risks: Overfitting to eval set.

User Story

- ☐ As a prompt engineer, I want a reflection step so that the model self-critiques and retries on hard tasks.

Acceptance Criteria (BDD)

- ☐ **Given** failing items, **when** I enable reflection, **then** pass rate improves $\geq 5\%$ with cost logged.
- ☐ **Given** risk of verbosity, **when** I cap tokens, **then** cost stays within budget.
- ☐ **Given** logs, **when** I analyze, **then** common failure modes are categorized.

Evidence to Attach

- ☐ Reflection prompts; before/after metrics; cost sheet.

CH25 — Automatic Multi-Step Reasoning & Tools

ID: LLMDP-CH25-01
Title: Build a task graph with tool selection heuristics
Epic / Feature: Chapter 25 — Multi-Step Patterns
Business Value: Autonomously decomposes tasks and selects tools.
Priority / Estimate: P3, 6 SP
Persona: Agent Engineer
Dependencies: Graph framework; tools registry.
Assumptions / Risks: Loops; dead-ends.

User Story

☐ As an agent engineer, I want a task graph with heuristics so that the agent executes multi-step tasks reliably.

Acceptance Criteria (BDD)

☐ **Given** composite tasks, **when** I run the graph, **then** steps execute with retries and backoff.

☐ **Given** tool registry, **when** I select tools, **then** correct tools are chosen $\geq 90\%$ on a labeled set.

☐ **Given** loops, **when** I enforce limits, **then** execution stops with a clear error.

Evidence to Attach

☐ Graph JSON; run logs; evaluation sheet.

CH26 — Retrieval-Augmented Generation (RAG)

ID: LLMDP-CH26-02
Title: Productionize RAG with ingestion + query APIs
Epic / Feature: Chapter 26 — RAG Patterns
Business Value: Grounded answers; lower hallucination.
Priority / Estimate: P1, 8 SP
Persona: Platform Engineer
Dependencies: Vector DB; embedding model; store.
Assumptions / Risks: Stale indexes; drift.

User Story

☐ As a platform engineer, I want ingestion and query APIs for RAG so that apps can use grounded answers at scale.

Acceptance Criteria (BDD)

☐ **Given** ingestion, **when** new docs arrive, **then** they are chunked, embedded, and indexed within SLA.

☐ **Given** queries, **when** I pass a question, **then** top-k docs and citations are returned with latency SLO.

☐ **Given** drift, **when** I re-index, **then** recall improves on a held-out query set.

Evidence to Attach

☐ API spec; load test; eval metrics (retrieval/generation).

CH27 — Graph-Based RAG

ID: LLMDP-CH27-01
Title: Build a small knowledge graph and hybrid retrieval
Epic / Feature: Chapter 27 — Graph RAG Patterns

Business Value: Improves retrieval on relational queries.
Priority / Estimate: P2, 6 SP
Persona: Data Engineer
Dependencies: KG store; ETL from docs.
Assumptions / Risks: Graph build cost; schema drift.

User Story

☐ As a data engineer, I want entity/relation extraction and graph queries so that retrieval exploits structure and semantics.

Acceptance Criteria (BDD)

☐ **Given** a corpus, **when** I extract entities/relations, **then** KG is populated with precision ≥ 0.9 on a checked sample.

☐ **Given** hybrid retrieval, **when** I compare to dense-only, **then** hit-rate improves on relational questions.

☐ **Given** updates, **when** I rebuild edges, **then** graph stays consistent with audit logs.

Evidence to Attach

☐ KG schema; extraction notebook; hybrid eval report.

CH28 — Advanced RAG (Query Reformulation & Re-Ranking)

ID: LLMDP-CH28-01
Title: Add re-ranking and iterative query reformulation
Epic / Feature: Chapter 28 — Advanced RAG Patterns
Business Value: Boosts answer quality on ambiguous queries.
Priority / Estimate: P3, 5 SP
Persona: LLM Engineer
Dependencies: Reranker; query strategies.
Assumptions / Risks: Extra latency; complexity.

User Story

☐ As an LLM engineer, I want iterative reformulation and re-ranking so that difficult queries retrieve better evidence.

Acceptance Criteria (BDD)

☐ **Given** ambiguous questions, **when** I enable reformulation, **then** answer correctness improves vs. baseline.

☐ **Given** a reranker, **when** I insert it, **then** nDCG@k increases with bounded latency overhead.

☐ **Given** tail queries, **when** I evaluate, **then** failure rate decreases by $\geq 15\%$.

Evidence to Attach

☐ Reformulation prompts; reranker config; metrics.

CH29 — Evaluating RAG Systems

ID: LLMDP-CH29-01
Title: Build an end-to-end RAG evaluation scorecard
Epic / Feature: Chapter 29 — RAG Evaluation Patterns
Business Value: Detects regressions across retrieval and generation.
Priority / Estimate: P1, 4 SP
Persona: QA Engineer
Dependencies: Ground-truth Q/A; judgments.
Assumptions / Risks: Labeling cost.

User Story

☐ As a QA engineer, I want a RAG scorecard so that retrieval and answer metrics are tracked together over time.

Acceptance Criteria (BDD)

☐ **Given** a test set, **when** I run the scorecard, **then** retrieval (recall, nDCG) and generation (factuality, faithfulness) are produced.

☐ **Given** a new release, **when** I compare, **then** deltas are highlighted and gates enforced.

☐ **Given** monitoring, **when** I sample prod queries, **then** drift alerts trigger when KPIs degrade.

Evidence to Attach

☐ Scorecard notebook; dashboard; gating policy.

CH30 — Agentic Patterns (Planning, Memory, Safety)

ID: LLMDP-CH30-01
Title: Ship a single-agent demo with short-term memory and guardrails
Epic / Feature: Chapter 30 — Agentic Patterns
Business Value: Demonstrates plan/act with risk controls.
Priority / Estimate: P2, 8 SP
Persona: Agent Engineer
Dependencies: Memory store; tools; policy filters.
Assumptions / Risks: Tool abuse; privacy constraints.

User Story

☐ As an agent engineer, I want an agent with planning, memory, and guardrails so that it can safely complete tasks with tools.

Acceptance Criteria (BDD)

☐ **Given** a goal, **when** I run the agent, **then** it creates a plan, executes tools, and stores relevant memory.

☐ **Given** safety policies, **when** unsafe requests occur, **then** the agent refuses and cites policy.

☐ **Given** eval tasks, **when** I measure, **then** success rate $\geq$ target with incident rate $< 1\%$.

Evidence to Attach

☐ Agent traces; memory snapshots; safety report.