ChatGPT

Elpis Trading Strategies Project Plan

Below is the revised, critically enhanced roadmap for your Elpis project. Each phase now includes clear objectives, prioritized tasks, key deliverables, time buffers, and suggested best practices.

Phase 1: Infrastructure & Project Governance

Objective: Establish a robust, reproducible, and version-controlled foundation.
Duration: 3 weeks (25–35 h)

Task	Description	Est. Time	Deliverable
1.1 Project Governance & Requirements Gathering	Define scope, success criteria, risk register, compliance needs. Set up Kanban/Agile board.	4 h	Project charter, risk log, backlog in Trello/Jira
1.2 Environment Management	Create reproducible Conda environment & Docker-compose setup. Automate via shell scripts.	6 h	`environment.yml`, `docker-compose.yml`, setup scripts
1.3 Version Control & CI/CD	Initialize Git repo. Configure CI pipeline (GitHub Actions) for linting, tests.	6 h	`.github/workflows/ci.yml`, code coverage badge
1.4 Database Foundation	Finalize SQL schema; deploy PostgreSQL + TimescaleDB (local + optional remote).	8 h	ER diagram, DDL scripts, hypertables ✓
1.5 Configuration & Secrets	Implement config management (Hydra/ pydantic) and secrets handling (Vault or `.env`).	4 h	Config templates + loader module
1.6 Logging & Monitoring	Set up structured logging (Loguru) and basic Prometheus metrics exporter.	4 h	Logging module, metrics endpoint
1.7 Documentation Setup	Bootstrap Jupyter Book or MkDocs; outline TOC for all phases.	3 h	Documentation skeleton

Buffer: + 20% time for unforeseen setup issues.

Phase 2: Automated Data Pipeline

Objective: Ingest, clean, and store market data with minimal manual intervention.
Duration: 3 weeks (35–45 h)

Task	Description	Est. Time	Deliverable
2.1 Data Ingestion	Saxo API client + adapters for other sources. Include retries & backfill logic.	8 h	`ingest.py` modules + tests
2.2 Resampling & Storage	Build ETL to resample raw 1 m ticks → 5 m, 15 m, 1 D. Persist in hypertables.	8 h	Resampled tables + sample queries
2.3 Feature Library	Compute returns, volatility, moving averages, RSI, custom indicators.	10 h	Feature module + UML diagram
2.4 Data Quality Checks	Implement checks: missing values, outlier thresholds, schema validation.	5 h	QA report, alerts on schema drift
2.5 Orchestration	Schedule jobs in Airflow or Prefect; include dependency graph.	4 h	DAG definitions, monitoring dashboard

> **Best Practices:** Containerize each step; write idempotent scripts; maintain clear logs.

Phase 3: Exploratory Data Analysis & Hypothesis Formulation

Objective: Understand market regimes, relationships, and candidate signals.
Duration: 2 weeks (20–30 h)

1. **Visual Analysis:** Price trends, volume heatmaps, spike detection.
2. **Statistical Tests:** Stationarity (ADF), autocorrelations (ACF/PACF), seasonality decomposition.
3. **Correlation & Feature Selection:** Pairwise; PCA for dimensionality reduction.
4. **Hypotheses:** Document promising signal–market relationships.

> **Deliverable:** EDA report with annotated charts, Jupyter notebooks, and hypothesis list.

Phase 4: Model Prototyping & Benchmarking

Objective: Establish performance baselines using statistical and ML methods.
Duration: 3 weeks (35–45 h)

Model Type	Key Activities	Deliverable
ARIMA/GARCH	Fit volatility & return models; capture heteroskedasticity.	Performance table (AIC, residual diagnostics)

Model Type	Key Activities	Deliverable
Machine Learning	Train RF, XGBoost, MLP on directional returns. Use time-aware CV.	Confusion matrix, ROC curves
Deep Learning (Optional)	Simple LSTM or temporal CNN for sequence modeling.	Initial train/val loss plots

Notes: Use MLflow for experiment tracking; compare against naive benchmark (random walk).

Phase 5: Backtesting & Risk Framework

Objective: Evaluate strategies under realistic trading conditions and risk constraints.
Duration: 3 weeks (30–40 h)

1. **Backtester Design:** Modular engine handling signals → orders.
2. **Market Impact & Slippage:** Parametric slippage models.
3. **Transaction Costs & Leverage:** Commission schedules, margin rules.
4. **Risk Controls:** Stop-loss, take-profit, position-sizing (Kelly or VAR).
5. **Performance Metrics:** Sharpe, Calmar, max drawdown, turnover.

Deliverable: Reusable backtest framework + sample reports.

Phase 6: Optimization & Sensitivity Analysis

Objective: Fine-tune parameters and assess robustness.
Duration: 4 weeks (50–65 h)

1. **Hyperparameter Tuning:** Optuna with multi-objective optimization.
2. **Feature Variants:** Test alternate indicators, lookback windows.
3. **Walk-forward Analysis:** Rolling re-optimize & test.
4. **Stress Testing:** Simulate extreme events, regime shifts.

Deliverable: Ranked strategies with parameter sets, performance distributions.

Phase 7: Live Simulation & Paper Trading

Objective: Validate end-to-end pipeline under near-live conditions.
Duration: 3 weeks (45–55 h)

1. **Real-Time Pipeline:** Hook into live data; generate signals continuously.
2. **Paper Trading Engine:** Execute simulated orders; log fills and P&L.
3. **Monitoring & Alerts:** Dashboard (e.g., Grafana) + anomaly detection.
4. **Feedback Loop:** Compare paper P&L vs backtest; refine latency/slippage models.

Deliverable: Live-simulation logs, dashboard link, post-mortem report.

Phase 8: Deployment & Scalability

Objective: Prepare for production deployment and multiasset scaling.
Duration: 4 weeks (50–65 h)

1. **Containerization & Orchestration:** Kubernetes manifests, Helm charts.
2. **Cluster Training:** Leverage GPUs (A100) for parallel hyperparameter searches.
3. **Multi-Asset Extension:** Add FX, commodities, equities universes.
4. **CI/CD for Production:** Automated builds, canary releases, rollbacks.
5. **Security & Compliance:** Secrets rotation, IAM roles, audit logging.

Deliverable: Production-ready pipeline, deployment guide, cost-analysis.

Phase 9: Performance Monitoring & Maintenance

Objective: Ensure ongoing strategy health, adapt to market changes.
Duration: Ongoing (5 h/week)

- **Daily Checks:** Data integrity, pipeline health.
- **Weekly Reviews:** Strategy P&L, risk metrics, drift detection.
- **Monthly Updates:** Retraining schedules, parameter refresh.
- **Incident Response:** Playbook for outages or large drawdowns.

Deliverable: Maintenance SOP, incident logs, performance dashboard.

Additional Recommendations

- **Time Buffers:** Always include a 15–25% buffer on each phase.
- **Documentation:** Maintain living docs; use automated doc tests.
- **Testing:** Unit tests for every module; integration tests in CI.
- **Code Reviews:** Enforce PR reviews and coding standards.
- **Stakeholder Updates:** Bi-weekly progress reports with KPIs.

This revised plan incorporates governance, risk management, and CI/CD best practices, extends to performance monitoring, and adds realistic buffers to help you stay on schedule and maintain code quality.