Dev B Phase 3 → Phase 4 Handoff Document

Objective

This document defines the complete handoff from Dev B (risk pipeline) to Phase 4 (integration, API exposure,

front-end consumption, and system orchestration).

Phase 3 Outputs Ready for Phase 4

PortfolioReturnSeries builder

Full risk metrics engine

FactorModel regression

ScenarioResult engine

RiskRails module

Unified RiskReportJSON object

Correlation matrix output

Required Consumption by Phase 4

Integrate Dev B Risk Engine with Backtest Pipeline

Use aggregate_from_backtest(backtest, weights) to link Dev A results into RiskReport builder.

Expose RiskReportJSON

Phase 4 must create:

REST endpoints OR internal service interface

Validation layer for inputs (weights, factor sets, scenarios)

Persist or Stream Output

Phase 4 decides whether RiskReportJSON is:

Stored in Postgres

Returned in real-time to UI

Streamed to a dashboard

Front-End Integration

Phase 4 takes the JSON and renders:

Time-series charts

Tables of exposures, risk rails, warnings

Scenario PnL breakdowns

Pipeline Coordination

Phase 4 ensures:

Dev A backtests feed into Dev B

Dev C macro/factor data feeds into Dev B

RiskReport delivered to UI/API layer

Interfaces to Use in Phase 4

Phase 4 should call:

```
from src.slice.risk.report import build_risk_report

report = build_risk_report(

portfolio=portfolio,

asset_returns=asset_returns,

weights=weights,

factor_data=factors,

scenarios=scenarios,

macro=macro_data,

)
```

Next Steps for Phase 4

Create REST API wrapper around build_risk_report

Create UI serializers for RiskReportJSON

Add caching and async execution where needed

Link outputs to dashboard charts

Conclusion

Dev B is complete. Phase 4 team can now integrate the risk engine into the full Slice system, expose its outputs, and build user-facing workflows.