Technical Digest

Sprint 4 Findings Interpreted via **AI-Agent Development Logic**

1 ▸ Agent-Architecture Context

Layer	Classical AI Taxonomy	Simulation Analogue	Design Implication
Reactive	Policy π(s) → a	Employee agents (rule-based morale changes)	Lightweight state machines suffice.
Deliberative / BDI	Belief–Desire–Intention reasoning	Manager agents (override, ethics rules)	Goal expansion & conflict resolution.
Learning (RL / MARL)	Value-function optimisation over time	AI restructuring agent (autonomy toggle)	Candidate for Deep RL upgrade in next version.

Take-away: The current AI agent is a **hand-coded policy**; results identify where learning modules should be added and where deterministic logic is adequate.

2 ▸ Reward-Function Engineering

$$R_t = \alpha\,\mathrm{Prod}_t + \beta\,\mathrm{Morale}_t - \gamma\,\mathrm{ExitRate}_t$$

Parameter	Empirical guidance	Development action
α productivity weight	High autonomy boosts productivity but risks morale under volatility.	Keep ≥ 0.4, but anneal downward when volatility > 0.6.
β morale weight	Morale predicts future productivity (lag = 5 ticks).	Increase to 0.3 to internalise lagged payoff.
γ turnover penalty	Exits surge costs > hiring lag.	Set ≥ 0.3; amplify during market shocks.

Multi-objective RL: deploy constrained-policy optimisation (e.g., PC-PG, Lagrangian method) to satisfy fairness while maximising composite return.

3 ▸ Alignment & Safety Mechanisms

Sprint-4 Element	AI-Safety Analogue	Future implementation
Bias-mitigation toggle	*Fairness regularisation* / side-constraint reward	Integrate as L2 penalty on demographic disparity.
Manager override loop	*RLHF* (Reinforcement Learning from Human Feedback)	Use preference queries to fine-tune policy.
Volatility shocks	*Adversarial domain randomisation*	Train with stochastic environment generator for robustness.

4 ▸ Autonomy Calibration Algorithm

Empirical rule of thumb:

```
If Volatility < 0.10 ➔ Autonomy_target = 0.8
If 0.10 ≤ Volatility ≤ 0.20 ➔ Autonomy_target = 0.6
If Volatility > 0.20 ➔ Autonomy_target = 0.4
```

Implementation path: contextual-bandit that selects autonomy level a_t to maximise rolling reward; treat manager vetoes as negative feedback.

5 ▸ Hierarchical Reinforcement Learning Proposal

```
Level-0  (HRL Worker)   ➔ Tactical actions: reassign / promote / terminate
Level-1  (HRL Manager)  ➔ Decides frequency & magnitude of restructuring
Level-2  (Meta-Controller) ➔ Adjusts autonomy threshold by sensing volatility
```

Benefits:

- credit-assignment clarity,
- smoother learning curves,
- plug-in slot for ethical governor at Level-1.

6 ▸ Multi-Agent Coordination & CTDE

Centralised Training, Decentralised Execution.

1. **Training stage:** AI and Manager agents share global state; gradient-based updates incorporate both performance and fairness.
2. **Execution stage:** Manager retains policy to veto; AI acts independently within calibrated autonomy band.
   Outcome: preserves the empirical "bounded autonomy" sweet spot found in Sprint-4.

---

7 ▸ Robustness & Generalisation

Sprint-4 finding	MARL technique	Rationale
Autonomy fails under volatility	Domain randomisation & ensemble policies	Provides worst-case experience during training.
Override loops risk deadlock	Opponent-modelling of manager policy	AI learns probability distribution of veto.
Morale-productivity lag	Recurrent (LSTM) critic	Captures temporal dependencies.

---

8 ▸ Implementation Road-Map (Agent-Centric)

Milestone	Tooling	KPI
M1 Replace rule-based AI with PPO on engineered reward.	Stable-Baselines3	Policy convergence < 1e-3.
M2 Add fairness penalty; evaluate Pareto frontier.	Constrained RL	p-value of bias < 0.05.
M3 Integrate RLHF override feedback loop.	DPO / InstructRL	Human approval rate ≥ 80 %.
M4 Deploy hierarchical policy with Level-2 meta-controller.	PyTorch + RLlib	QVI uplift ≥ 10 %.

---

9 ▸ Research Challenges & Extensions

1. **Causal RL** to separate correlation (morale) from causal drivers.
2. **Counterfactual Policy Evaluation** to test new reward shapes offline.
3. **Transparent RL** (saliency or concept bottlenecks) to satisfy managerial explainability demands.

10 ▸ Key References

SUTTON, Richard; BARTO, Andrew. *Reinforcement Learning: An Introduction*. 2. ed. Cambridge: MIT Press, 2018.
LI, Lihong. A review of hierarchical reinforcement learning. *Foundations and Trends in Machine Learning*, v. 10, n. 4, p. 367-457, 2017.
CHRISTIANO, Paul et al. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, p. 4299-4307, 2017.
LEIKE, Jan et al. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.
STONE, Peter; VELOSO, Manuela. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, v. 8, p. 345-383, 2000.