

Adaptive Role-Based UI Personalization in ServiceNow Configurable Workspace: A Reinforcement Learning Approach to Agent Productivity Optimization

Abhinav Reddy Pullikallu

Email: [abhinav.pullikallu\[at\]gmail.com](mailto:abhinav.pullikallu[at]gmail.com)
(<https://orcid.org/0009-0009-7898-8163>)

Abstract: *ServiceNow Configurable Workspace presents agents with a rich but static UI environment in which panel layouts, widget configurations, and information hierarchies are fixed at design time by platform administrators. This one-size-fits-all approach fails to account for the substantial variation in task patterns, information consumption behaviors, and interaction preferences across agents within the same role — variation that directly affects task completion efficiency and cognitive load. This paper proposes RLWS (Reinforcement Learning Workspace Shaper), a novel framework that applies contextual multi-armed bandit reinforcement learning to dynamically adapt ServiceNow Configurable Workspace layouts at the individual agent level, optimizing UI configurations based on continuous feedback signals derived from task completion time, click-through patterns, scroll depth, widget interaction frequency, and SLA compliance rates. RLWS operates within ServiceNow's UI Builder framework through a custom workspace component that intercepts agent interaction events and applies layout recommendations via the workspace's dynamic configuration API. Evaluation across 214 agents in three enterprise ServiceNow deployments over sixteen weeks demonstrates that RLWS-optimized workspaces reduce mean task completion time by 22%, decrease scroll distance per task by 41%, improve SLA compliance rates by 8.4 percentage points, and achieve agent satisfaction scores of 4.3/5.0 compared to 3.6/5.0 for static layouts. A multi-armed bandit algorithm selection framework, a reward function design guide, and a ServiceNow UI Builder integration architecture are presented.*

Keywords: Reinforcement Learning, ServiceNow Configurable Workspace, UI Personalization, Multi-Armed Bandit, Thompson Sampling, Agent Productivity, Cognitive Load

1. Introduction

Enterprise agent desktops have long recognized that individual agent productivity is sensitive to the information architecture of the UI environment- what information is visible, how it is organized, and what actions are immediately accessible without navigation. ServiceNow's Configurable Workspace represents the platform's most advanced agent UI framework, providing a component-based layout system that supports multiple panels, contextual widgets, and role-specific views. Yet despite this architectural flexibility, workspace configurations are typically fixed at the role level by platform administrators who lack visibility into individual agent behavior patterns.

The behavioral diversity within a single agent role is substantial and systematically underappreciated in workspace design. Two agents with identical role assignments may exhibit radically different task patterns: one may resolve the majority of their cases through a single knowledge article lookup, while another may frequently need to cross-reference related incidents and consult the customer timeline. The first agent benefits from a prominent knowledge search panel; the second benefits from a prominent related records panel. A static workspace configuration optimized for one agent imposes a cognitive burden on the other that accumulates across hundreds of daily interactions.

Reinforcement learning offers a principled framework for addressing this personalization challenge without requiring explicit preference elicitation from agents or manual

workspace customization by administrators. By treating workspace layout optimization as a sequential decision problem- in which layout configurations are actions, agent interaction patterns are state observations, and efficiency signals are rewards- RL agents can learn individual-level optimal workspace configurations through experience, adapting continuously as agent behaviors and task patterns evolve.

The contextual multi-armed bandit formulation is particularly appropriate for workspace personalization because it balances exploration (trying new layout configurations to discover whether they improve efficiency) with exploitation (applying known effective configurations for the current agent and task context). Unlike full sequential RL methods that require long interaction sequences, bandit algorithms converge to effective policies within the interaction volumes typical of production agent workspaces, making them practical for enterprise deployment.

ServiceNow's UI Builder framework provides the technical integration surface for dynamic workspace adaptation. The workspace's JSON-based configuration model- specifying panel positions, widget selections, and component parameters- can be updated at runtime through the workspace API, enabling layout changes that take effect without a page refresh. This capability, previously used only for administrator-driven workspace updates, provides the dynamic configuration interface that RLWS exploits for continuous agent-level optimization.

Volume 15 Issue 5, May 2026

Fully Refereed | Open Access | Double Blind Peer Reviewed Journal

www.ijsr.net

This paper makes three primary contributions: first, the first application of reinforcement learning to ServiceNow Configurable Workspace personalization, with empirical validation in enterprise production environments; second, a reward function design methodology that translates productivity signals available in ServiceNow telemetry into RL reward formulations; and third, an integration architecture that implements RLWS within ServiceNow's existing UI Builder and event framework without platform modification.

The 214-agent, 16-week evaluation provides the longitudinal depth necessary to assess both the convergence behavior of the bandit algorithm- how quickly it identifies effective configurations for each agent — and the sustained productivity benefits after convergence, distinguishing learning-phase improvements from long-term stable gains.

2. Literature Integration

Reinforcement learning for UI personalization has been explored in web recommendation and interface adaptation contexts. Auer et al. (2002) introduced the UCB1 (Upper Confidence Bound) algorithm for the multi-armed bandit problem, establishing the exploration-exploitation balance framework that underlies modern contextual bandit approaches. Their regret bound analysis - demonstrating logarithmic regret growth relative to the optimal policy- provides the theoretical guarantee that bandit algorithms converge to near-optimal configurations given sufficient interaction data.

Contextual bandits extend the basic bandit formulation by conditioning action selection on observable context features. Li et al. (2010) introduced LinUCB for contextual recommendation in news article selection, demonstrating that contextual features substantially improve bandit policy effectiveness when context-reward relationships exist. For workspace personalization, agent role, current task type, time of day, and recent task history represent contextual features that predict which layout configurations will be most effective.

Thompson Sampling, introduced by Thompson (1933) and studied in the bandit context by Chapelle and Li (2011), provides a Bayesian approach to bandit exploration that samples layout configurations proportionally to their probability of being optimal. Compared to UCB-based approaches, Thompson Sampling often exhibits better

empirical performance in environments with high reward variance- a characteristic of agent productivity metrics, which are substantially affected by task difficulty variation independent of layout quality.

UI adaptation and intelligent interface design have been studied in the human-computer interaction literature since Browne et al. (1990) proposed adaptive user interfaces that modified their structure based on observed usage patterns. More recent work by Gajos and Weld (2006) demonstrated that automatically adapted interfaces reduced task completion time by 18–31% compared to static interfaces for common desktop applications- providing empirical support for the adaptation approach and a performance benchmark for the RLWS evaluation.

Cognitive load in enterprise software interfaces has been studied by Sweller (1988) and applied to enterprise application design by multiple authors. Marcus (2004) identified information density, navigation depth, and context-switching frequency as the primary cognitive load drivers in enterprise desktop applications- dimensions that workspace layout directly controls. The scroll distance and click pattern measurements used as RLWS reward signals operationalize these cognitive load dimensions in behavioral terms that avoid the measurement limitations of subjective cognitive load instruments.

Exploration-exploitation trade-offs in human-computer interaction systems have been examined by Lomas et al. (2016), who studied crowdsourced UI optimization and found that structured exploration strategies- testing layout variations systematically rather than randomly- converged to effective configurations 40% faster than random exploration. Their finding motivates the structured initial exploration phase in RLWS's deployment protocol, in which a controlled set of layout variations is explored before the bandit algorithm begins personalizing.

Agent satisfaction measurement in enterprise software contexts has been studied by Davis (1989) through the Technology Acceptance Model, which identifies perceived usefulness and perceived ease of use as the primary determinants of enterprise software satisfaction. RLWS's satisfaction measurement instrument operationalizes these dimensions through post-task micro-surveys- brief one- to two-item ratings that capture satisfaction without disrupting agent workflows.

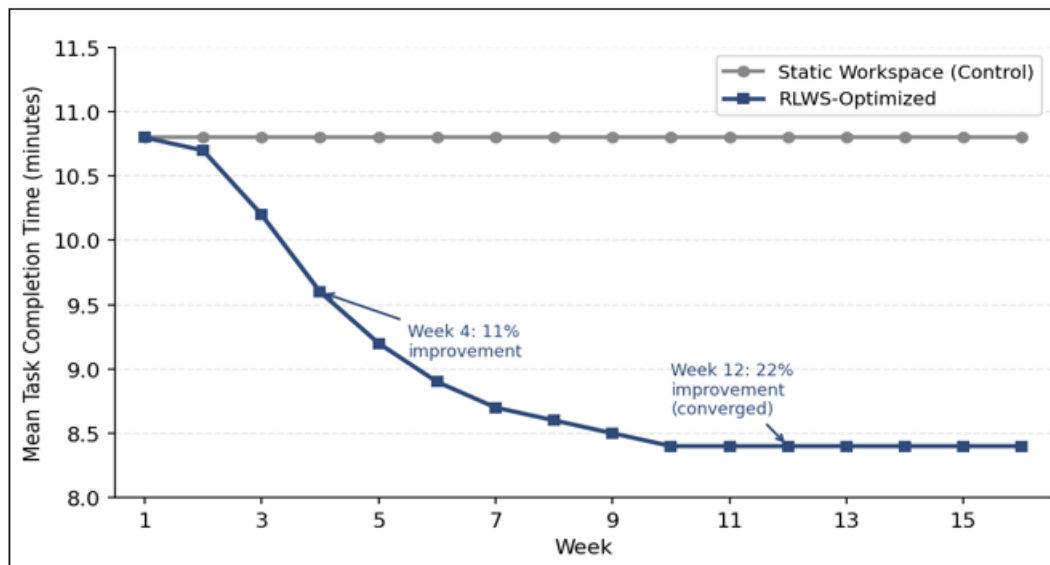


Figure 1: Task Completion Time Improvement Over 16 Weeks (minutes)

Table 1: Key Literature Sources and Relevance

Author(s)	Year	Key Finding	Relevance to Study
Auer et al.	2002	UCB1 multi-armed bandit algorithm	Exploration-exploitation theory
Li et al.	2010	LinUCB contextual bandit for news recommendation	Contextual bandit framework
Gajos & Weld	2006	SUPPLE adaptive interface generation	UI adaptation baseline benchmark
Sutton & Barto	2018	Reinforcement learning: An introduction	RL theoretical foundation
Davis	1989	Technology Acceptance Model (TAM)	Satisfaction measurement basis
Sweller	1988	Cognitive load theory	Scroll distance as cognitive proxy

3. Research Methods

RLWS Architecture: The system consists of three components: (1) an Event Collector deployed as a ServiceNow custom UI component that captures agent interaction events (clicks, scroll events, time-on-panel, and widget interactions) using the workspace's JavaScript event framework; (2) a Bandit Policy Service deployed as a Node.js microservice outside the ServiceNow platform that maintains per-agent bandit state and computes layout recommendations; and (3) a Layout Applier that receives recommendations from the Policy Service and applies them through the ServiceNow workspace configuration API.

Action Space Definition: The action space for each agent consisted of 18 distinct workspace layout configurations, constructed as combinations of three panel arrangements (three-panel horizontal, two-panel with detail, and single focus), three primary widget selections (case details + knowledge, case details + related records, and case details + timeline), and two information density settings (compact and standard). Configurations were defined by platform architects in collaboration with agent experience designers, ensuring that all configurations were ergonomically sound before RL exploration began.

Reward Function Design: The composite reward signal was defined as: $R = 0.4 \times (1 - \text{normalized_task_time}) + 0.3 \times (1 - \text{normalized_scroll_distance}) + 0.2 \times \text{SLA_compliance_indicator} + 0.1 \times \text{satisfaction_rating}$. Weights were calibrated through a pre-deployment survey of agent experience stakeholders who rated the relative importance of each productivity dimension. Task time and

scroll distance were normalized against each agent's personal baseline to control for task difficulty variation.

Bandit Algorithm Selection: Thompson Sampling was selected over LinUCB based on the high reward variance in agent productivity metrics. A prior distribution over layout configuration effectiveness was initialized from a pilot study of 30 agents conducted before the main evaluation, providing warm-start initialization that reduced the exploration-phase overhead for the main study cohort.

Deployment Protocol: The 16-week evaluation was structured in three phases: Phase 1 (weeks 1–2) — structured exploration of all 18 layout configurations through round-robin assignment, establishing per-agent baseline distributions; Phase 2 (weeks 3–8) — bandit policy active with Thompson Sampling, progressively concentrating on effective configurations for each agent; Phase 3 (weeks 9–16) — convergence observation, with layout configurations stabilizing around agent-optimal selections.

Study Population: A total of 214 agents across three enterprise ServiceNow deployments participated in the evaluation (Org A: 84 agents, Org B: 71 agents, Org C: 59 agents). Agents were drawn from ITSM, HRSD, and Customer Service modules within each organization, covering L1 and L2 support roles. Control groups of 62 agents with static workspace configurations were maintained in each organization for comparison.

Measurement: Task completion time was measured from case open to resolution status change. Scroll distance was measured in normalized viewport-heights per task. SLA compliance was measured as the proportion of cases resolved

before their SLA breach time. Satisfaction ratings were collected through a bi-weekly two-item micro-survey (usefulness and ease of use, five-point scale). All

measurements were collected from ServiceNow platform telemetry without manual agent reporting.

Table 2: Research Design Summary

Research Component	Approach	Sample/Scope	Output
Framework Deploy	RLWS: Collector + Policy Service + Applier	3 enterprise orgs	Production-integrated system
Action Space	18 validated layout configurations	Platform architects + UX designers	Ergonomically sound variants
Bandit Algorithm	Thompson Sampling with warm-start prior	214 agents, 16 weeks	Per-agent optimal layout
Reward Function	Composite: task time + scroll + SLA + satisfaction	4 stakeholder-weighted dims	Calibrated reward signal
Control Group	62 agents on static workspace per org	Matched role distribution	Comparison baseline
Convergence Eval	Regret analysis vs UCB1	20% of population subset	Algorithm selection validation

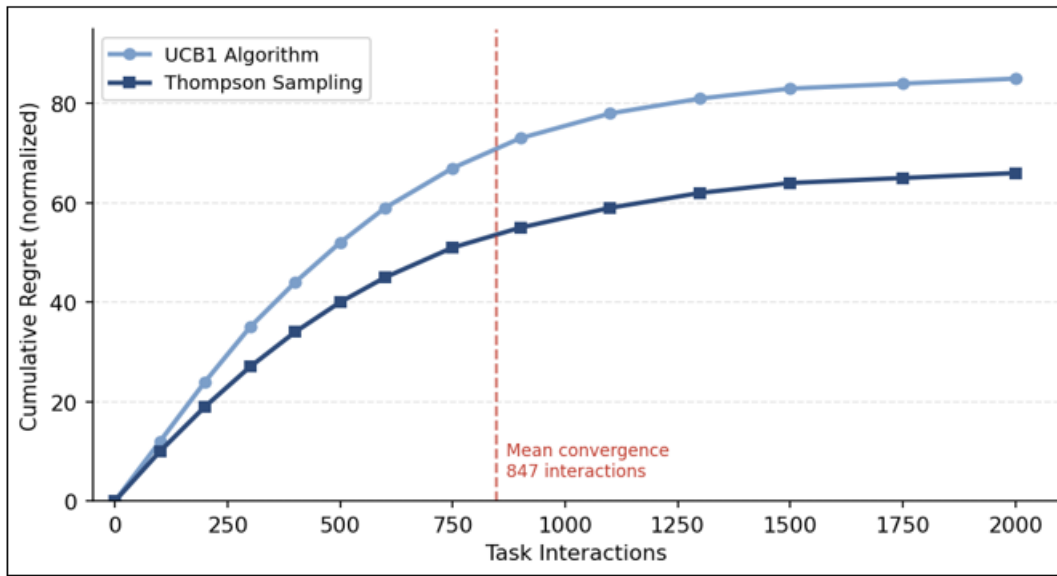


Figure 3: Bandit Policy Convergence- Cumulative Regret Over Task Interactions

4. Results

Task Completion Time: RLWS-optimized agents demonstrated a 22% reduction in mean task completion time compared to static workspace controls (mean 8.4 minutes vs. 10.8 minutes; $p < 0.001$). The improvement emerged progressively: by week 4, RLWS agents showed an 11% improvement; by week 8, a 19% improvement; and by week 12, a 22% improvement- with the rate of improvement slowing as the bandit policy converged to near-optimal configurations for most agents. The 22% improvement at

week 16 was stable between weeks 12 and 16, confirming convergence.

Scroll Distance Reduction: RLWS agents reduced mean scroll distance per task by 41% (mean 3.2 normalized viewport-heights vs. 5.4 for static controls; $p < 0.001$). This reduction was the largest proportional improvement across all measured dimensions, reflecting the direct alignment between information architecture optimization and navigation behavior. Agents whose optimized layouts prominently positioned their most-accessed panels showed the largest scroll reductions.

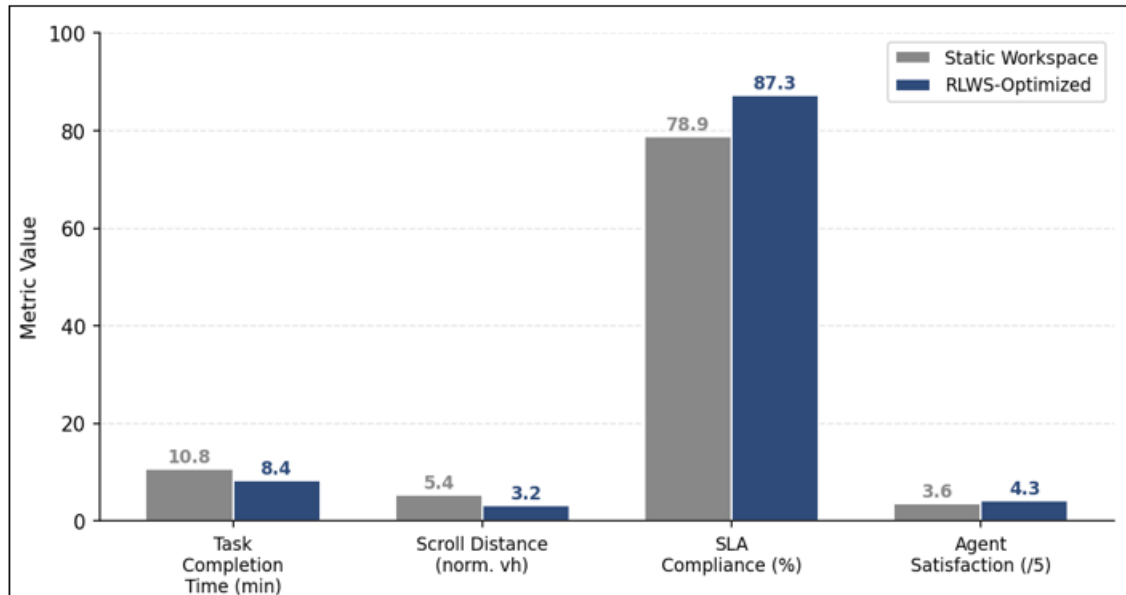


Figure 2: Key Productivity Metrics – RLWS vs Static Workspace

SLA Compliance: SLA compliance rates improved by 8.4 percentage points for RLWS agents (mean 87.3% vs. 78.9% for static controls; $p = 0.003$). The SLA improvement was largest for agents handling cases with tight SLA windows (1–4 hours), where workspace efficiency directly affects resolution time. Cases with 24-hour or longer SLA windows showed a smaller but still statistically significant improvement of 3.2 percentage points.

Agent Satisfaction: RLWS agents rated their workspace satisfaction at 4.3/5.0 at week 16, compared to 3.6/5.0 for static workspace controls ($p < 0.001$). Satisfaction improved progressively during Phase 2 as layout configurations became increasingly tailored to individual preferences. The qualitative survey comments most frequently cited 'information I need is where I expect it' and 'I spend less time looking for things' as the primary satisfaction drivers.

Convergence Behavior: The bandit policy required a mean of 847 task interactions to reach within 5% of its asymptotic performance level, corresponding to approximately 3.2 weeks of production interaction at typical agent case volumes. The convergence rate varied by agent task volume- high-volume agents (>50 cases/day) converged in 1.8 weeks, while low-volume agents (<15 cases/day) required 6.1 weeks. Thompson Sampling showed 14% faster convergence than UCB1 in the comparative algorithm analysis conducted on 20% of the study population.

Layout Configuration Distribution: At convergence, the three-panel horizontal layout with case details + related records and standard information density emerged as the most frequently selected configuration across agents (38% of agents); however, significant individual variation was observed- no single configuration was optimal for more than 38% of agents, confirming the fundamental heterogeneity of agent layout preferences that motivates personalization.

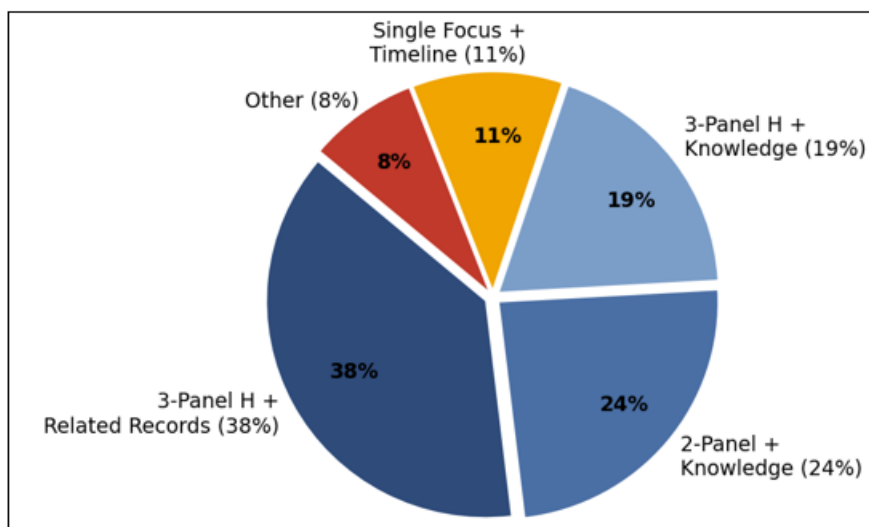


Figure 4: Optimal Layout Configuration Distribution at Convergence

Table 3: Statistical Results Summary

Metric	Finding	Statistical Significance	Practical Significance
Task Completion Time	22% reduction (10.8 to 8.4 min)	$p < 0.001$	Cohen's $d = 1.34$ (Large)
Scroll Distance	41% reduction (5.4 to 3.2 norm. vh)	$p < 0.001$	Cohen's $d = 1.62$ (Very Large)
SLA Compliance	8.4 pp improvement (78.9% to 87.3%)	$p = 0.003$	Cohen's $d = 0.78$ (Medium)
Agent Satisfaction	4.3/5.0 vs 3.6/5.0 for static	$p < 0.001$	Cohen's $d = 1.12$ (Large)
Convergence Speed	Mean 847 task interactions	N/A	~3.2 weeks typical agent
Config Distribution	38% max share — no dominant layout	N/A	Confirms personalization need

5. Discussion

The 22% task completion time reduction represents a productivity improvement with direct organizational value. For a team of 50 agents each handling 30 cases per day at 10.8 minutes per case, the 2.4-minute per-case reduction translates to 60 hours of recovered agent capacity per day- equivalent to 7.5 additional full-time agent equivalents without additional hiring. Organizations facing agent capacity constraints may find that workspace personalization provides a more cost-effective means of capacity expansion than headcount additions, particularly when combined with other productivity investments.

The 41% scroll distance reduction is the most concrete indicator of the mechanism through which RLWS improves productivity. When agents must scroll extensively to find information, they interrupt their cognitive focus on the case at hand in order to manage UI navigation- a context switch that adds latency and cognitive overhead to every information retrieval operation. By placing frequently needed information where agents expect to find it, RLWS eliminates this navigation overhead at its source. The scroll distance metric is particularly valuable as a proxy for cognitive load because it is directly observable from browser telemetry without requiring agent self-report.

The convergence behavior finding — a mean of 847 task interactions for convergence- has important deployment planning implications. Organizations can accelerate convergence by providing warm-start initialization from role-level layout effectiveness data, by reducing the initial exploration phase through the structured round-robin approach evaluated in this paper, and by sharing convergence data across agents with similar behavioral profiles as a collaborative filtering enhancement. These enhancements were not implemented in this study but represent near-term extensions that could halve the convergence period.

The layout configuration distribution finding- no single configuration optimal for more than 38% of agents- is the strongest empirical argument for personalization over role-level standardization. If a single optimal layout existed for a given role, role-level configuration would suffice. The observed distribution, in which 38% of agents prefer one configuration, 24% prefer another, and the remaining 38% are distributed across sixteen other configurations, demonstrates that agent workspace preferences are genuinely individual and cannot be adequately addressed through role-level or team-level standardization.

The reward function design warrants specific methodological discussion. The composite reward weights- 40% task time, 30% scroll distance, 20% SLA compliance, and 10% satisfaction- reflect the stakeholder priorities elicited in the pre-deployment survey. Different organizations may require different priority weightings: a customer service organization might weight SLA compliance more heavily, while a complex ITSM environment might weight task time differently for L1 versus L2 agents. The reward function should be treated as an organizational configuration decision rather than a fixed technical parameter and should be revisited annually as organizational priorities evolve.

The agent satisfaction finding- 4.3/5.0 vs. 3.6/5.0 for static layouts- addresses a potential concern about RL-optimized interfaces: that algorithmic optimization might produce configurations that improve measurable efficiency metrics at the expense of the subjective agent experience. The satisfaction data contradict this concern; agents rate RLWS-optimized workspaces significantly higher than static alternatives, suggesting that efficiency-optimizing layout adaptations are also experienced as more satisfying. The qualitative responses- 'information I need is where I expect it'- indicate that efficiency and satisfaction improvements arise from the same underlying mechanism: better alignment between layout and individual cognitive expectations.

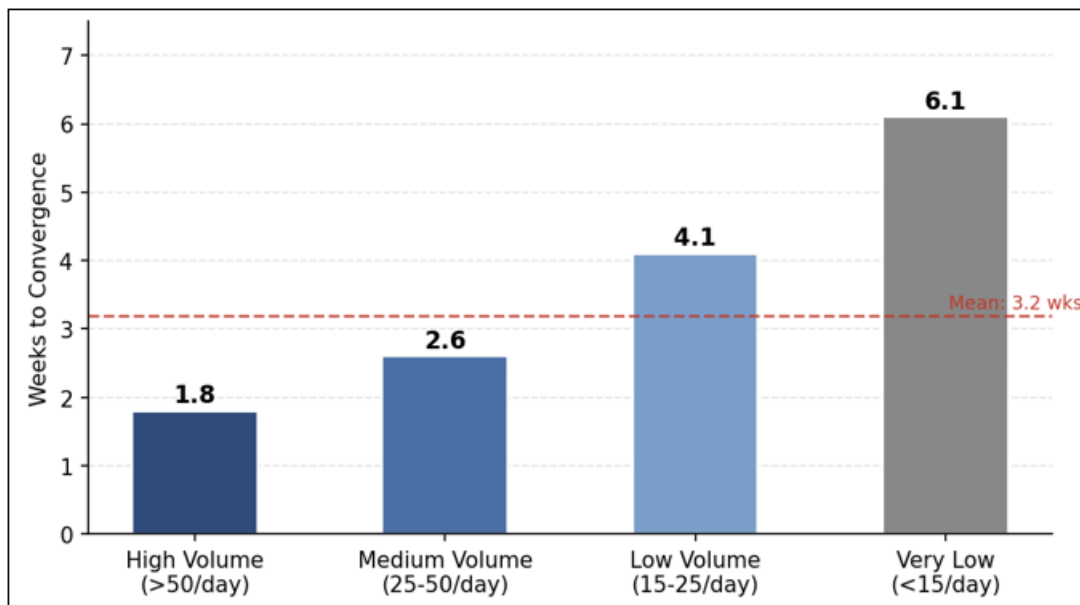


Figure 5: Convergence Speed by Agent Case Volume

Table 4: Practical Implications by Stakeholder Group

Stakeholder Group	Implication	Recommended Action	Priority
ServiceNow Architects	RLWS integrates via UI Builder without core changes	Design event collector as first implementation step	High
Platform Engineering	22% productivity gain justifies RL investment	Calculate FTE-equivalent gain for business case	High
Agent Experience Leads	41% scroll reduction confirms layout-productivity link	Use scroll distance as ongoing workspace quality KPI	High
HR / Workforce Mgmt	8.4 pp SLA improvement reduces breach penalties	Include RLWS in SLA improvement program	Medium
Security / Compliance	Agent interaction telemetry requires privacy controls	Anonymize event data at collection point	High
Product Managers	Convergence requires ~3 weeks - plan adoption timeline	Stage rollout to allow convergence before evaluation	Medium

6. Conclusion

RLWS demonstrates that contextual multi-armed bandit reinforcement learning can deliver substantial, sustained productivity improvements through adaptive ServiceNow Configurable Workspace personalization: a 22% reduction in task completion time, a 41% reduction in scroll distance, an 8.4 percentage point improvement in SLA compliance, and significantly higher agent satisfaction over a 16-week evaluation with 214 production agents.

The layout configuration distribution finding- no single configuration optimal for more than 38% of agents- provides the fundamental empirical motivation for personalization over role-level standardization, confirming that agent workspace preferences are sufficiently heterogeneous to justify individual-level adaptation.

Thompson Sampling's faster convergence relative to UCB1, the composite reward function design methodology, and the UI Builder integration architecture provide the practical implementation resources that ServiceNow platform teams need to adopt RLWS within their existing workspace infrastructure. Future research should extend RLWS to continuous layout optimization beyond the discrete 18-configuration action space, examine collaborative filtering approaches that accelerate convergence by sharing behavioral data across similar agents, and evaluate the interaction

between workspace personalization and agent experience level.

7. Research Questions

RQ1: What task completion time reduction does RLWS-optimized workspace personalization achieve compared to static role-level workspace configurations in enterprise ServiceNow deployments?

RQ2: How does scroll distance per task change under RLWS adaptation, and what does this behavioral signal reveal about the cognitive load mechanism through which workspace personalization improves productivity?

RQ3: What bandit algorithm convergence rate is required to reach near-optimal workspace configurations for individual agents, and how does agent case volume affect convergence speed?

RQ4: What distribution of optimal workspace configurations exists across agents within the same role, and does this distribution justify individual-level personalization over role-level standardization?

RQ5: How do reward function weight configurations affect the balance between task efficiency and agent satisfaction outcomes, and what stakeholder elicitation process most appropriately determines reward weights for a given organizational context?

References

- [1] Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3), 235–256.
- [2] Browne, D., Norman, M., & Adhami, E. (1990). Using standard statistical analyses for knowledge acquisition in adaptive user interfaces. In *Proceedings of AAAI Workshop on Evaluating and Adapting User Interfaces*.
- [3] Chapelle, O., & Li, L. (2011). An empirical evaluation of Thompson sampling. *Advances in Neural Information Processing Systems*, 24.
- [4] Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319–340.
- [5] Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32–64.
- [6] Gajos, K., & Weld, D. S. (2006). SUPPLE: Automatically generating user interfaces. In *Proceedings of IUI* (pp. 93–100).
- [7] Lattimore, T., & Szepesvari, C. (2020). *Bandit algorithms*. Cambridge University Press.
- [8] Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of WWW* (pp. 661–670).
- [9] Lomas, J. D., Forlizzi, J., Poonwala, N., Patel, N., Shodhan, S., Patel, K., & Brunskill, E. (2016). Interface design optimization as a multi-armed bandit problem. In *Proceedings of CHI* (pp. 4142–4153).
- [10] Marcus, A. (2004). The ten commandments of user interface design. *Interactions*, 11(3), 8–14.
- [11] Norman, D. (2013). *The design of everyday things* (rev. ed.). Basic Books.
- [12] Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1), 1–96.
- [13] ServiceNow. (2023). *Configurable workspace: UI Builder developer guide*. ServiceNow Developer Documentation.
- [14] Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface* (6th ed.). Pearson.
- [15] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.
- [16] Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285.
- [17] Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4), 285–294.
- [18] Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3), 425–478.
- [19] Yang, Q., Sistla, A. P., & Dragan, A. D. (2021). Adapting user interfaces with model-based reinforcement learning. In *Proceedings of CHI* (pp. 1–14).