Article

Not peer-reviewed version

Toward AI-Augmented Technical Interviewing: A ConceptualFramework for Assessing Developer Competencies in AI-Mediated Software Development

Amit Rangari *

Posted Date: 4 March 2026

doi: 10.20944/preprints202603.0313.v1

Keywords: AI-augmented technical interviews; AI-augmented interview framework (AAIF); AImediated software development; developer competency assessment; human-AI collaboration; technical hiring; structured interviews; behaviorally anchored rating scales (BARS); interview validity; design science research; output evaluation; prompt engineering assessment; socio-technical systems; hiring fairness and bias mitigation; responsible ai in recruitment

 doi:10.20944/preprints202603.0313.v1

Article

Toward AI-Augmented Technical Interviewing: A Conceptual Framework for Assessing Developer Competencies in AI-Mediated Software Development

Amit Rangari

Independent Researcher, USA; amitrangari@gmail.com

Abstract

This paper presents a conceptual framework, the AI-Augmented Interview Framework (AAIF), requiring empirical validation before deployment. No interviews have been conducted; all thresholds, weights, and KPI linkages are conjectures pending empirical testing. The accelerating adoption of AI-powered development tools (GitHub Copilot, ChatGPT, Claude) is transforming software engineering practice. Industry surveys indicate that over 75% of professional developers now use AI coding assistants regularly (noting potential self-selection bias in survey samples), yet fewer than one in four organizations assess AI fluency during technical interviews. AAIF proposes a structured five-stage interview methodology (Stage 0 fundamentals gate plus four AI-augmented stages) for evaluating developer competencies in AI-mediated environments. The framework assesses: (1) toolchain fluency and prompt engineering, (2) AI output evaluation and critical reasoning, (3) system-oriented problem solving with AI integration, and (4) meta-reasoning about AI limitations, ethics, and failure modes. We develop evaluation rubrics with behaviorally anchored rating scales, propose configurable decision thresholds, and provide an integrated risk framework addressing bias, fairness, legal compliance, and ethical dimensions. The novelty lies in the systematic integration of established methods from industrial-organizational psychology, software engineering, and risk management for the specific and underexplored problem of assessing developers who use AI tools. A detailed four-phase empirical validation protocol is proposed as a key contribution.

Keywords: AI-augmented technical interviews; AI-augmented interview framework (AAIF); AI-mediated software development; developer competency assessment; human-AI collaboration; technical hiring; structured interviews; behaviorally anchored rating scales (BARS); interview validity; design science research; output evaluation; prompt engineering assessment; socio-technical systems; hiring fairness and bias mitigation; responsible ai in recruitment

---

1. Toward AI-Augmented Technical Interviewing: A Conceptual Framework for Assessing Developer Competencies in AI-Mediated Software Development

Amit Rangari

IT Leader with experience across JPMorgan Chase, IBM, Cognizant, and Virtusa in banking, telecom, and pharmaceutical domains

1.1. I. Introduction

1.1.1. A. Motivation

The rapid proliferation of AI-enabled development tools, from code generation assistants such as GitHub Copilot [3] to conversational AI systems such as ChatGPT [4] and Claude [50], has fundamentally transformed how software is built. Peng et al. [1] demonstrated that developers using Copilot completed tasks 55% faster than those without it (note: this finding is from an arXiv preprint; readers should assess accordingly). Chen et al. [5] documented how AI coding assistants reshape

 doi:10.20944/preprints202603.0313.v1

development workflows by shifting cognitive effort from syntax production to design reasoning and output validation. Industry surveys from Stack Overflow [2] and the OECD [6] confirm that AI tool adoption has reached critical mass across the profession. McKinsey Global Institute [47] estimates that generative AI could add $2.6–4.4 trillion annually in productivity gains across knowledge work, with software development among the highest-impact domains.

This transformation creates a fundamental misalignment in technical hiring. Traditional coding interviews, whiteboard exercises, algorithmic puzzles, and manual implementation tasks, evaluate skills that are decreasingly representative of actual development work. Behroozi et al. [7] found that traditional whiteboard interviews induce stress that systematically disadvantages certain candidates without predicting job performance. When 75% of developers use AI tools daily but interviews prohibit them, assessment ecological validity is fundamentally compromised. We note that the 75% figure comes from the Stack Overflow Developer Survey [2], which surveys a self-selected sample that may skew toward more technically engaged developers; the actual adoption rate across the full developer population may be lower.

This misalignment has implications beyond software engineering. Technical hiring is a gateway process that shapes workforce composition, organizational capability, and career trajectories across industries. HR researchers, organizational psychologists, and workforce policy analysts have long studied interview validity [21,22]; the AI augmentation of development work introduces new dimensions that these communities must address alongside software engineering researchers. Sackett et al. [51] have recently revised the foundational meta-analytic estimates of interview validity, underscoring the importance of evidence-based assessment design.

1.1.2. B. Problem Statement

Despite growing recognition of this misalignment, no systematic framework exists for evaluating candidates who use AI as a development tool. Prior research has addressed related but distinct problems: AI as an assessment tool for evaluating candidates; algorithmic fairness in AI-driven hiring decisions [8]; and AI tool adoption by developers [1,5,9]. However, the fundamental question this paper addresses is different: **How should we evaluate candidates who use AI as a development tool, rather than using AI to evaluate candidates?**

This distinction is critical. Existing AI-in-hiring research [8], [23] focuses on algorithmic decision-making in screening and ranking. Our concern is the assessment of human-AI collaboration competencies, how effectively a developer can leverage, evaluate, critique, and orchestrate AI-powered tools in realistic workflows. Industry platforms such as HackerRank, Karat, and CodeSignal have begun incorporating AI-augmented assessments, but as we analyze in Section III.B (Table II), these efforts differ from AAIF in specific dimensions including theoretical grounding, systematic risk analysis, and empirical validation protocols.

1.1.3. C. Approach Overview

We propose the AI-Augmented Interview Framework (AAIF), a structured five-stage methodology (Stage 0 fundamentals gate plus four AI-augmented stages) grounded in socio-technical systems theory [10,11], the Technology Acceptance Model [12], and Activity Theory [13]. AAIF assesses four core competencies: Toolchain Fluency (TF), AI Output Evaluation (AOE), System Orchestration (SO), and Meta-Reasoning (MR). Each competency is operationalized through behaviorally anchored rating scales, configurable stage weights, and proposed decision thresholds. We further develop an integrated risk framework synthesizing bias, fairness, legal compliance (GDPR, CCPA, EEOC), and ethical dimensions.

To illustrate potential framework extensibility, we apply AAIF to quantum computing, a specialized domain exhibiting the same fundamental hiring challenges while introducing domain-specific complexities.

Transparency note: This is a conceptual framework paper. No interviews have been conducted, no empirical data have been collected, and all proposed thresholds, weights, and KPI linkages are

 doi:10.20944/preprints202603.0313.v1

conjectures pending empirical validation. We consider the detailed empirical validation protocol itself to be a key contribution, providing a roadmap for the rigorous testing this framework requires.

1.1.4. D. Novelty Statement

The novelty of this work lies in the systematic integration of established methods from industrial-organizational psychology (BARS, structured interviews), software engineering (competency frameworks, DSR methodology), and risk management (algorithmic fairness, regulatory compliance) for the specific problem of AI-augmented developer hiring. No individual component of the framework is novel in isolation: structured multi-stage interviews are standard practice at major technology companies; BARS methodology dates to Smith and Kendall (1963) [26]; weighted scoring functions are basic psychometric methodology; risk frameworks for hiring draw on extensive algorithmic fairness literature. The contribution is in the structured combination, theoretical grounding, and integrated risk analysis applied to a genuinely underexplored problem, assessing developers who use AI tools rather than using AI to assess developers.

1.1.5. E. Contributions

This paper makes the following contributions:

1. **C1: AI-Augmented Interview Framework (AAIF)**: A structured five-stage interview methodology (Stage 0 gate plus four AI-augmented stages) for AI-augmented development roles, with formal competency definitions, a weighted assessment function, and configurable decision criteria. We position AAIF against existing industry platforms through a structured comparison (Table II). *(Section III)*
2. **C2: Evaluation Rubrics with Behavioral Anchors**: Detailed rubrics with behaviorally anchored rating scales (BARS) for each competency level (1–5), providing specific observable behaviors that distinguish adjacent performance levels, including common rater errors to avoid. We propose hypothesized linkages between AAIF stages and organizational KPIs, labeled as conjectures pending empirical calibration. *(Section IV)*
3. **C3: Illustration of Potential Domain Extensibility**: Application of AAIF to quantum computing, illustrating that the core assessment principles may transfer to highly specialized domains when adapted to domain-specific constraints. We frame this as an illustrative demonstration requiring independent empirical validation. *(Section VI)*
4. **C4: Integrated Risk Framework and Empirical Validation Protocol**: A practitioner-oriented synthesis of bias, fairness, legal compliance, and ethical considerations for AI-augmented hiring, together with a detailed four-phase empirical validation protocol specifying study designs, sample sizes with power analysis justifications, target metrics, and ethical safeguards for future research. *(Sections V, VII.C)*

1.1.6. F. Paper Organization

The remainder of this paper is organized as follows. Section II describes the research methodology and theoretical foundations. Section III presents the AAIF framework, including formal definitions, the assessment function with psychometric analysis, and a structured comparison with existing approaches. Section IV develops behaviorally anchored evaluation rubrics with common rater errors. Section V presents the integrated risk framework, expanded threats to validity (following Messick's [52] unified framework), and ethical considerations. Section VI illustrates domain extensibility through quantum computing. Section VII discusses implications, framework boundary conditions, limitations, the empirical validation protocol with power analyses, and future work. Section VIII provides an expanded illustrative use case with scored examples. Section IX concludes.

 doi:10.20944/preprints202603.0313.v1

1.2. II. Research Methodology

1.2.1. A. Design Science Research Approach

This work follows the Design Science Research (DSR) methodology as articulated by Hevner et al. [14] and refined by Peffers et al. [15]. DSR is appropriate for developing and evaluating artifacts, in this case, the AAIF framework, that address identified organizational problems. Following the DSR framework, our research proceeds through six activities:

1. **Problem Identification and Motivation**: The misalignment between traditional technical interviews and AI-augmented development workflows (Section I.A).
2. **Objectives of a Solution**: A structured interview framework that assesses AI collaboration competencies while maintaining evaluation of fundamental understanding (Section I.B).
3. **Design and Development**: The AAIF five-stage framework with formal definitions, rubrics, and decision criteria (Sections III–IV).
4. **Demonstration**: Application of AAIF to quantum computing as a specialized domain, and expanded illustrative use case with scoring (Sections VI, VIII).
5. **Evaluation**: Threats to validity analysis following Messick's [52] unified validity framework, and proposed empirical validation protocol (Sections V.C–V.D, VII.C). We acknowledge that DSR evaluation ideally includes empirical testing of the artifact [14]; the current work provides analytical evaluation (threats to validity) and descriptive evaluation (scenario-based demonstrations) but defers empirical evaluation to future work. This means the DSR process is incomplete, and the framework's validity remains unestablished pending empirical testing.
6. **Communication**: This manuscript.

We acknowledge that the partial completion of activity 5 is a significant limitation. Hevner et al. [14] list five evaluation methods (observational, analytical, experimental, testing, descriptive), and the current work employs only analytical and descriptive methods. The proposed four-phase validation protocol (Section VII.C) addresses the remaining evaluation types.

1.2.2. B. Theoretical Lenses

We employ three established theoretical frameworks as analytical lenses, not as novel theoretical contributions, to inform AAIF design. For each lens, we describe both the theoretical basis and the specific design decision it informed.

Socio-Technical Systems (STS) Theory [10,11]: STS theory emphasizes the co-evolution of technical and social subsystems and the principle of joint optimization, designing technical and social components together rather than optimizing one at the expense of the other. We apply STS to conceptualize AI-augmented hiring as a socio-technical system where technology (AI development tools, assessment platforms), social structures (interviewer judgment, organizational culture, candidate expectations), and regulatory contexts co-evolve. *Specific design impact*: STS theory's emphasis on joint optimization directly informed the design of the integrated risk framework (Section V). Rather than treating technical assessment and social/ethical concerns as separate domains, AAIF integrates risk analysis across bias, fairness, legal, and ethical dimensions within the assessment design itself. The inclusion of Stage 4 (Meta-Reasoning) as a core competency, rather than an optional add-on, reflects STS's insistence that social and ethical dimensions are co-equal with technical capabilities in a well-functioning socio-technical system.

Technology Acceptance Model (TAM) [12]: TAM explains technology adoption through perceived usefulness (PU) and perceived ease of use (PEOU). We apply TAM to analyze stakeholder acceptance of AI-augmented interview formats. Venkatesh et al. [16] extended TAM to organizational contexts through UTAUT, which addresses social influence and facilitating conditions. *Specific design impact*: TAM analysis reveals distinct acceptance barriers for each AAIF stakeholder group:

 doi:10.20944/preprints202603.0313.v1

- *Hiring managers*: High PU (better signal on AI collaboration skills) but low PEOU (more complex interview logistics, tool provisioning). This analysis informed the phased rollout recommendation (Section VII.A) and quick-start guide, which progressively introduce complexity.
- *Candidates*: PU depends on perceived fairness; PEOU depends on prior AI tool experience. This informed the pre-interview orientation provision (Section V.A.3) and advance tool notification (Section IV.F).
- *HR professionals*: Low PU if compliance burden is unclear; moderate PEOU if rubrics are interpretable. This informed the regulatory compliance matrix (Table VII) and the design of BARS rubrics for HR accessibility.
- *Interviewers*: PU depends on whether framework improves hiring signal; PEOU depends on calibration burden. This informed the interviewer certification and calibration protocol (Section IV.F).

Activity Theory [13,17]: Activity Theory provides a framework for analyzing tool-mediated human activity, particularly contradictions within activity systems. In AI-augmented interviews, AI tools serve dual roles: they are simultaneously development tools candidates use and assessment instruments interviewers evaluate. *Specific design impact*: This dual-role tension creates a fundamental contradiction: candidates are motivated to use AI tools to perform well (tool-as-amplifier), while interviewers assess how candidates use AI tools (tool-as-assessment-object). AAIF explicitly addresses this contradiction through Stage 2 (AOE), which separates the *product* of AI-assisted work from the *process* of AI collaboration. By requiring candidates to critique AI outputs (not just produce them), AAIF ensures that effective AI tool use does not mask lack of understanding. The Stage 0 fundamentals gate further addresses this contradiction by establishing a baseline independent of AI tool access.

These theories inform framework design but are not empirically validated in this study. Future work should test specific theoretical propositions, particularly around stakeholder acceptance barriers (TAM analysis) and the dual-role tension resolution effectiveness (Activity Theory).

1.2.3. C. Literature Synthesis Method

The related work and gap identification in this paper draw on a structured (but not systematic) literature review. We document the synthesis method for reproducibility.

Search strategy: We searched IEEE Xplore, ACM Digital Library, Scopus, and Google Scholar using combinations of the terms: "technical interview," "AI-augmented development," "coding assessment," "prompt engineering evaluation," "hiring AI tools," "developer productivity AI," and "algorithmic fairness hiring." Searches were conducted between October 2025 and February 2026.

Inclusion criteria: Papers were included if they: (a) addressed technical interviewing methodology, validation, or reform; (b) examined AI tool usage in software development contexts; (c) analyzed bias, fairness, or legal dimensions of AI-augmented hiring; (d) provided theoretical frameworks applicable to technology adoption or tool-mediated work; or (e) addressed quantum computing workforce development. Papers were required to be published in peer-reviewed venues, with exceptions for significant arXiv preprints (clearly labeled) and authoritative industry reports.

Exclusion criteria: Papers were excluded if they: (a) addressed AI in hiring contexts unrelated to developer assessment (e.g., resume screening, video interview analysis); (b) focused on AI tool development rather than usage or assessment; or (c) were published before 2015 with no continued relevance to current assessment practices. Foundational works (e.g., Smith and Kendall, 1963 [26]; Messick, 1989 [52]) were retained regardless of date.

Selection process: We reviewed 120+ papers by title and abstract, conducted full-text review of approximately 80 papers, and selected 62 references spanning five domains: (1) technical interviewing research, (2) AI in hiring and algorithmic fairness, (3) AI tools in software development, (4) theoretical frameworks for technology adoption, and (5) quantum computing workforce development. The selection reflects the author's judgment; a single-reviewer process introduces potential selection bias.

 doi:10.20944/preprints202603.0313.v1

We acknowledge this is not a systematic review following PRISMA guidelines and recommend such a review as a priority for future work.

1.2.4. D. Guiding Research Questions

The following research questions guide the design and structure of the AAIF framework. We frame these as guiding questions rather than empirical hypotheses, consistent with the conceptual nature of this work. Empirical answers require the validation protocol described in Section VII.C.

- **GQ1**: Are the four competencies defined in the AAIF model (TF, AOE, SO, MR) sufficient and distinct constructs for assessing AI collaboration skills in software development roles?
- **GQ2**: How can interview frameworks evaluate AI fluency while maintaining rigorous assessment of foundational technical understanding?
- **GQ3**: How might a general-purpose AI-augmented interview framework be adapted for specialized domains (e.g., quantum computing)?
- **GQ4**: What bias, fairness, legal, and ethical risks arise from AI-augmented hiring, and how can they be systematically mitigated?

1.3. III. The AI-Augmented Interview Framework (AAIF)

1.3.1. A. Background: The Changing Landscape of Software Development

The role of the software developer is undergoing a fundamental transformation. Developers increasingly function as system designers, AI orchestrators, and model evaluators rather than solely as code authors [1,5,18]. AI-powered toolchains, encompassing code completion (Copilot [3], CodeWhisperer [19]), conversational assistants (ChatGPT [4], Claude [50]), AI-augmented CI/CD pipelines, and AI-powered incident response tools, reshape day-to-day work by shifting cognitive effort from syntax production to design reasoning, prompt engineering, and output validation.

Note on terminology[1]: Throughout this paper, we use the term "AI-augmented development" to describe the paradigm where developers leverage AI tools as productivity accelerators while retaining responsibility for design, validation, and judgment.

Table I: Competency Shift in AI-Augmented Development

Traditional Skills	Emerging Skills
Data structures and algorithms	Problem framing and decomposition
Manual system design	AI orchestration and prompt engineering
Syntax-driven coding	Toolchain integration and evaluation
Manual debugging	Model steering and interpretability
API memorization	AI output analysis and critique
Solo code authoring	Collaborative human-AI workflows

Figure 1. AAIF Framework Architecture: Five-stage assessment process showing Stage 0 (fundamentals gate) feeding into four AI-augmented stages (TF, AOE, SO, MR), with assessment function aggregation and decision criteria. Arrows indicate prerequisite relationships and score flow. [Author-created diagram]

[1] This term is preferred over "Model-Centric Programming" (which may be confused with Model Context Protocol) and over "AI-assisted coding" (which understates the scope of transformation).

 doi:10.20944/preprints202603.0313.v1

Figure 1. AAIF Framework Architecture

1.3.2. B. Related Work, Industry Comparison, and Research Gap

Technical interviewing research. Schmidt and Hunter [21] established through meta-analysis that structured interviews have higher predictive validity (r = 0.51) than unstructured formats (r = 0.38). However, Sackett et al. [51] substantially revised these meta-analytic estimates downward, finding that much of the apparent validity was attributable to cognitive ability rather than the interview format itself. The revised estimates suggest that the incremental validity of structured interviews, while still positive, is more modest than originally reported. Behroozi et al. [7] examined stress and anxiety in whiteboard coding interviews, finding systematic disadvantage for certain candidates. Huffcutt and Arthur [22] confirmed the superiority of structured behavioral interviews across occupational categories. These findings from industrial-organizational psychology provide the empirical foundation for our emphasis on structured, behaviorally anchored assessment.

AI in hiring. Raghavan et al. [8] analyzed bias in algorithmic hiring systems, finding demographic disparities in AI-based screening. Sanchez-Monedero et al. [23] studied how candidates game algorithmic hiring systems. The EU AI Act [24] classifies employment-related AI as "high-risk," imposing transparency and fairness requirements. This body of work informs our risk framework but addresses a different problem (AI evaluating candidates) than ours (evaluating candidates who use AI).

AI tools in software development. Peng et al. [1] provided large-scale evidence of Copilot's productivity impact (55% faster task completion; arXiv preprint). Vaithilingam et al. [9] studied developer expectations versus experiences with AI code generation. Barke et al. [25] identified two interaction modes (acceleration and exploration) with AI code generators. Kaddour et al. [48] surveyed challenges and applications of large language models.

Human-AI collaboration in work contexts. The Computer-Supported Cooperative Work (CSCW) literature provides relevant frameworks for understanding AI-mediated work. Research on human-AI teaming [53], AI-mediated collaboration [54], and the distribution of cognitive labor between humans and AI systems [55] informs how we conceptualize the competencies required for effective AI-augmented development. AAIF draws on these insights to design assessment stages that evaluate not just AI tool operation but the broader cognitive and collaborative skills required for effective human-AI work.

Industry platform analysis. To position AAIF relative to existing industry efforts, we conducted a structured comparison of major technical assessment platforms. Table II presents this analysis.

Table II: Structured Comparison of AAIF with Industry Assessment Platforms

Feature	HackerRank	CodeSignal	Karat	AAIF
AI tool access in assessment	Emerging (AI-allowed tracks)	AI-powered assessment generation	Limited	Core design principle

 doi:10.20944/preprints202603.0313.v1

Feature	HackerRank	CodeSignal	Karat	AAIF
Fundamentals gate (no AI)	Yes (traditional track)	Yes (traditional track)	Yes	Yes (Stage 0, explicit gate with defined threshold)
Prompt engineering evaluation	No dedicated assessment	Experimental	No	Yes (Stage 1, dedicated rubric)
AI output critical evaluation	No	Partial (code review tasks)	No	Yes (Stage 2, dedicated rubric)
System-level AI orchestration	No	No	Partial (system design rounds)	Yes (Stage 3, dedicated rubric)
Meta-reasoning assessment	No	No	No	Yes (Stage 4, dedicated rubric)
Behaviorally anchored rubrics	Proprietary scoring	Proprietary scoring	Structured rubrics	Open BARS with published anchors
Theoretical grounding	Not published	Not published	Not published	STS, TAM, Activity Theory
Published risk framework	Annual reports	Limited	Not published	Integrated (bias, legal, ethical)
Regulatory compliance matrix	Not published	Not published	Not published	Published (Table VII)
Empirical validation protocol	Internal (not published)	Internal (not published)	Internal (not published)	Published four-phase protocol
Domain extensibility method	Platform-dependent	Platform-dependent	Interview-dependent	Explicit adaptation template

Note: This comparison is based on publicly available information as of February 2026. Internal capabilities of proprietary platforms may exceed what is publicly documented. We acknowledge that the characterization of industry approaches is limited by information availability and invite corrections from platform providers.

Gap. Prior work addresses AI as an assessment tool (AI evaluating candidates), not assessment of candidates who use AI as a development tool. If a substantial proportion of developers use AI tools daily [2], interviews prohibiting these tools lack ecological validity. Based on our analysis (Table II), existing industry platforms have begun addressing elements of AI-augmented assessment but differ from AAIF in providing an integrated approach combining: (a) explicit theoretical grounding; (b) published behaviorally anchored rubrics; (c) systematic risk analysis; (d) a transparent empirical validation protocol; and (e) a domain extensibility methodology. We acknowledge that proprietary platform capabilities may not be fully reflected in public documentation.

1.3.3. C. Competency Model

Definition 1 (AAIF Competency Model): Let C = {c_1, c_2, c_3, c_4} be the set of core competencies assessed by AAIF:

- c_1: **Toolchain Fluency (TF)**, Proficiency with AI development tools and prompt engineering
- c_2: **AI Output Evaluation (AOE)**, Critical assessment of AI-generated outputs
- c_3: **System Orchestration (SO)**, Integration of AI tools into larger systems and workflows

 doi:10.20944/preprints202603.0313.v1

- c_4: **Meta-Reasoning (MR)**, Reflection on AI limitations, ethics, and failure modes

Competency Completeness Discussion: The four competencies were identified through synthesis of the literature on AI-augmented development [1,5,9,25], structured interviewing [21,22], and the author's professional experience across four organizations. We intend these as *representative* core competencies, not an exhaustive taxonomy. Several competencies that practitioners may encounter are not explicitly represented as standalone dimensions:

- **Collaborative AI usage** (pair programming with AI suggestions, team-based AI workflows) spans TF, AOE, and SO simultaneously.
- **AI-assisted debugging** requires AOE (evaluating AI diagnostic suggestions) and TF (formulating debugging prompts) in combination.
- **Knowledge management with AI** (using AI for documentation, onboarding, knowledge transfer) is partially captured by SO but may warrant independent assessment in knowledge-intensive roles.
- **AI tool configuration and customization** (setting up code completion rules, training custom models) is partially captured by TF but may emerge as a distinct competency as AI tools mature.

We chose four competencies based on the principle of parsimony: enough dimensions to capture the essential aspects of AI-augmented work without creating an impractically complex assessment. The model should evolve as empirical data becomes available. Phase 2 of the validation protocol (Section VII.C) includes confirmatory factor analysis to test whether these four competencies are empirically distinct constructs, and whether additional competencies emerge from the data.

1.3.4. D. Assessment Function and Psychometric Analysis

Specification 1 (Assessment Function): For candidate k, the overall AAIF assessment score is:

$$A(k) = w\_TF * S\_TF(k) + w\_AOE * S\_AOE(k) + w\_SO * S\_SO(k) + w\_MR * S\_MR(k)$$

where w_i are configurable stage weights (summing to 1.0) and S_i(k) in [1, 5] is the stage score.

We label this a "Specification" rather than "Definition" to accurately reflect that it is a practical scoring protocol, not a mathematically novel contribution. Its value lies in operationalizing the competency model with explicit, configurable parameters.

Default stage weights (recommended defaults pending empirical calibration):

Stage	Weight	Rationale
Toolchain Fluency (TF)	w_1 = 0.25	Foundation for AI-augmented work
AI Output Evaluation (AOE)	w_2 = 0.25	Critical for code quality and safety
System Orchestration (SO)	w_3 = 0.30	Highest complexity; closest to senior-level work
Meta-Reasoning (MR)	w_4 = 0.20	Important for long-term success and trust

These weights are proposed defaults based on the author's assessment of competency importance. They should be adjusted based on organizational context and, when available, empirical correlation with job performance data.

Psychometric Assumptions and Limitations: The weighted average specification makes several assumptions that users should understand:

1. **Compensatory model**: The weighted average is fully compensatory, a high score in one competency can offset a low score in another, subject to minimum threshold constraints (Specification 2).

 doi:10.20944/preprints202603.0313.v1

We chose a compensatory model because AI-augmented development roles typically allow individuals to compensate for moderate weakness in one area through strength in others. However, the minimum-score floors in the decision criteria provide a partial non-compensatory constraint. *Example of a potential limitation*: A candidate scoring TF=5, AOE=5, SO=5, MR=1 would receive A(k) = 0.25(5) + 0.25(5) + 0.30(5) + 0.20(1) = 4.2, meeting the "Hire (Mid-level)" threshold despite having no awareness of AI limitations. Organizations should evaluate whether such outcomes are acceptable and adjust minimum thresholds accordingly.

2. **Equal-interval assumption**: The model treats the difference between scores 1 and 2 as equivalent to the difference between scores 4 and 5. This equal-interval assumption is standard for Likert-type scales but is rarely validated and may be violated in practice. If the behavioral distance between "Insufficient" and "Developing" is substantially larger than between "Strong" and "Exceptional," the linear combination may be distorted. Empirical calibration in Phase 2 should include analysis of score distributions to test this assumption.
3. **Independence assumption**: The model assumes competencies contribute independently to the overall score. In practice, competency interactions are likely significant, strong SO likely requires adequate TF, and effective AOE may be a prerequisite for meaningful MR. A multiplicative or interaction-based model (e.g., A(k) = product of weighted scores, or inclusion of interaction terms) could capture these dependencies. We defer to empirical evidence from the validation protocol to determine whether interaction effects are large enough to warrant a more complex model.
4. **Ordinal vs. interval scaling**: The 1–5 BARS scores are technically ordinal data. Treating ordinal scores as interval data in a weighted average is a common but methodologically questionable practice. Item Response Theory (IRT) models, which account for item difficulty and rater severity, would be more psychometrically rigorous. We recommend IRT analysis as part of Phase 2 validation to determine whether the simpler weighted average provides adequate approximation.
5. **Sensitivity to weight changes**: Small changes in weights can shift candidates across decision thresholds. For example, changing w_SO from 0.30 to 0.25 (and redistributing 0.05 to MR) could change a borderline candidate's outcome. Organizations should conduct sensitivity analyses before deploying specific weight configurations.

Score aggregation across interviewers: When multiple interviewers assess the same stage, scores should be aggregated as follows:

- **Preferred method**: Discussion to consensus, where interviewers compare observations and agree on a single score. This produces the most reliable assessments but requires scheduling coordination.
- **Alternative method**: Mean of independent scores, used when consensus discussion is impractical. Report the standard deviation; if SD > 1.0, convene a discussion to resolve the disagreement.
- **Minimum panel size**: Two interviewers per stage minimum; three recommended for senior-level assessments. Single-interviewer scores should be flagged as lower-reliability.
- **Disagreement resolution**: If interviewers disagree by 2+ points after discussion, a third interviewer independently evaluates the same stage using recorded session materials. The modal score is used.

Specification 2 (Decision Criteria): Proposed decision thresholds (pending empirical calibration):

- **Hire (Senior)**: A(k) >= 4.5 AND min(S_i) >= 4 AND Stage 0 passed
- **Hire (Mid-level)**: A(k) >= 3.5 AND min(S_i) >= 3 AND Stage 0 passed
- **Hire (Junior)**: A(k) >= 3.0 AND S_TF >= 3 AND learning velocity score >= 3 AND Stage 0 passed
- **No Hire**: A(k) < 3.0 OR any S_i < 2 OR Stage 0 not passed

Learning velocity operationalization: For Junior-level assessments, "learning velocity" is assessed through a behavioral anchor: the interviewer rates the candidate's improvement trajectory during the interview itself on a 1–5 scale. Score 5: Visibly improves prompt quality and evaluation

 doi:10.20944/preprints202603.0313.v1

depth within the session, applying feedback immediately. Score 3: Shows some improvement when given guidance but does not self-correct. Score 1: No observable improvement despite feedback and guidance.

These thresholds are recommended starting points. Organizations should calibrate thresholds using ROC analysis against job performance data when available. The optimal thresholds depend on the organization's tolerance for false positives (hiring candidates who underperform) versus false negatives (rejecting candidates who would have succeeded).

1.3.5. E. Interview Stages

Stage 0: Fundamentals Assessment (No AI Tools)

Goal: Verify core computer science knowledge independent of AI assistance. This stage addresses the "superficial correctness" risk: candidates who prompt AI effectively but lack foundational understanding [7].

Activity: A focused assessment (30–45 minutes) without access to AI tools, evaluating: - Core data structures and algorithmic reasoning (not memorized solutions) - System design fundamentals (scalability, consistency, availability trade-offs) - Language fundamentals appropriate to the role

Pass/fail criteria: Stage 0 uses a separate 5-point fundamentals scale:

Score	Label	Criteria
5	Exceptional	Demonstrates deep understanding across all assessed areas; explains trade-offs fluently
4	Strong	Solid understanding with minor gaps; can reason through unfamiliar problems
3	Adequate	Demonstrates working knowledge of core concepts; may struggle with advanced topics
2	Developing	Significant gaps in foundational knowledge; can explain basic concepts only
1	Insufficient	Cannot demonstrate basic understanding of core concepts

Pass threshold: Score >= 3 (Adequate) required to proceed to AI-augmented stages. Candidates scoring 2 receive feedback and may re-apply after a defined period (recommended: 3–6 months). The threshold of 3 is chosen because Stage 0 is a gate, not a differentiator: it ensures minimum competence rather than ranking candidates.

Rationale: AI toolchain fluency is trainable within months; foundational computer science knowledge takes years to develop. This stage ensures AAIF does not inadvertently select "prompt engineers who cannot code" over fundamentally strong developers who are new to AI tools.

Note: Stage 0 is not weighted in the AAIF assessment function. It serves as a prerequisite gate with defined pass/fail criteria as specified above.

Stage 1: Toolchain Fluency and Prompt Engineering (TF)

Goal: Assess the candidate's proficiency with AI development tools, focusing on prompt design, iteration, and effective tool selection.

 doi:10.20944/preprints202603.0313.v1

Activity: Present a multi-part prompt engineering task (60–90 minutes) requiring candidates to navigate complex scenarios using AI tools. Example tasks include: - Solving a problem both with and without AI tools, then comparing approaches - Designing and iterating prompts to generate a specific software component - Selecting appropriate AI tools for different subtasks

Skills Assessed: Prompt design, iterative refinement, tool selection, problem framing.

Hypothesized KPI Linkage *(conjecture)*: On-the-job velocity, this stage is conjectured to predict how quickly a new hire adapts to AI-augmented workflows.

Stage 2: AI Output Evaluation (AOE)

Goal: Test the ability to critically assess, debug, and improve AI-generated code and outputs.

Activity: Present candidates with AI-generated artifacts containing subtle errors, design flaws, security vulnerabilities, or conceptual mistakes (60–90 minutes). Candidates must identify problems, explain root causes, and propose corrections.

Skills Assessed: Critical evaluation of AI outputs, debugging reasoning, reliability assessment, distinguishing correct from plausible-but-wrong outputs.

Hypothesized KPI Linkage *(conjecture)*: Retention and code quality, critical thinkers who identify AI limitations are conjectured to succeed longer in complex roles.

Stage 3: System-Oriented Problem Solving (SO)

Goal: Evaluate the ability to decompose real-world problems and orchestrate solutions using AI tools within larger system contexts.

Activity: High-level problem scenario (60–90 minutes) requiring orchestration of AI tools, APIs, and manual logic to build a working solution (e.g., designing an MVP with AI assistance, architecting a microservice with AI-generated components).

Skills Assessed: System design with AI components, tool orchestration, reasoning under ambiguity, trade-off analysis.

Hypothesized KPI Linkage *(conjecture)*: Time-to-productivity, a candidate's ability to solve realistic problems with AI tools is conjectured to predict operational readiness.

Stage 4: Meta-Reasoning and Reflection (MR)

Goal: Assess how candidates reason about AI limitations, failure modes, ethical considerations, and responsible use.

Activity: Structured discussion (30–45 minutes) exploring trade-offs, hallucination risks, bias concerns, and fallback strategies. Scenario-based prompts encourage candidates to demonstrate genuine reasoning rather than rehearsed answers.

Skills Assessed: AI ethics reasoning, failure mode understanding, stakeholder communication, responsible AI deployment.

Hypothesized KPI Linkage *(conjecture)*: Long-term retention and organizational trust, candidates with realistic expectations about AI capabilities are conjectured to build stakeholder confidence.

Important implementation note: Meta-reasoning assessment can devolve into "who can recite AI ethics talking points." Interviewers should use concrete scenarios (e.g., "Your AI-generated code passes all tests but you suspect a subtle bias in the training data; walk me through your next steps") rather than abstract questions to elicit genuine reasoning.

1.3.6. F. Comparison: AAIF vs. Traditional Interviews

Table III: AAIF vs. Traditional Interview Frameworks

Aspect	Traditional Interviews	AAIF
AI Tool Access	Prohibited	Provided and assessed
Fundamentals Check	Implicit in coding tasks	Explicit Stage 0 with defined pass/fail threshold

 doi:10.20944/preprints202603.0313.v1

Aspect	Traditional Interviews	AAIF
Primary Focus	Algorithm implementation	AI orchestration and evaluation
Evaluation Target	Code correctness	Judgment and critical thinking
Failure Mode Testing	Edge cases in code	AI hallucinations and limitations
Theoretical Grounding	Implicit/none	Explicit (STS, TAM, Activity Theory)
KPI Linkage	Weak or absent	Hypothesized (pending validation)
Risk Framework	Ad hoc	Integrated (bias, legal, ethical)
Score Aggregation	Varies	Defined protocol (consensus or mean with SD)

1.4. IV. Behaviorally Anchored Evaluation Rubrics

1.4.1. A. Rubric Design Principles

Industrial psychology research demonstrates that unanchored rating scales produce inter-rater reliability of 0.5–0.6 (poor), while behaviorally anchored rating scales (BARS) *in general* achieve 0.7–0.9 (good to excellent) [26,27]. We emphasize that these reliability estimates are for BARS methodology broadly, not for the specific AAIF rubrics proposed here. The actual inter-rater reliability of AAIF rubrics is unknown and must be established empirically (Phase 1 target: Cohen's kappa > 0.70).

Following BARS methodology, we provide specific observable behaviors for each score level across all four AAIF competencies. We acknowledge that standard BARS construction involves critical incident generation by subject matter experts, retranslation by an independent panel, and empirical testing [26]. The current rubrics were developed by a single author based on professional experience; they have not undergone expert panel validation. This represents a departure from standard psychometric methods and is the most significant threat to the rubrics' content validity. Expert panel validation is recommended as a priority before deployment (see Section VII.B).

Each rubric table includes a "Common Rater Errors" row based on Hauenstein's (1998) research showing that including negative anchoring (what NOT to look for) improves inter-rater reliability.

1.4.2. B. Toolchain Fluency (TF) Behavioral Anchors

Table IV: Behavioral Anchors for Toolchain Fluency (TF)

 doi:10.20944/preprints202603.0313.v1

Score	Label	Observable Behaviors
5	Exceptional	Demonstrates sophisticated prompt strategies (decomposition, few-shot examples, constraint specification) that show deep understanding of AI tool capabilities. Strategically combines multiple AI tools (e.g., ChatGPT for architecture, Copilot for implementation). Adapts tool selection based on task characteristics. Proactively anticipates edge cases AI might miss. May achieve effective results on first attempt *or* through deliberate iterative refinement, both patterns indicate expertise.
4	Strong	Writes effective prompts within 2–3 iterations. Validates AI output against requirements before accepting. Uses AI tools appropriately for task context. Shows comfort with multiple tools and can explain tool selection rationale.
3	Adequate	Writes functional prompts after multiple iterations. Checks AI output for obvious errors. Can use at least one AI tool competently. Relies on trial-and-error but eventually achieves desired results.
2	Developing	Struggles to craft effective prompts; prompts are vague, overly broad, or miss critical constraints. Accepts AI output without validation. Limited experience with AI development tools. Requires guidance to use tools productively.
1	Insufficient	Cannot use AI tools effectively even with guidance. Shows no understanding of prompt quality factors. Unable to iterate on prompts based on output quality.

 doi:10.20944/preprints202603.0313.v1

Score	Label	Observable Behaviors
–	**Common Rater Errors**	Do not confuse rapid prompting with effective prompting, speed is not a proxy for skill. Do not penalize candidates who ask clarifying questions before prompting; this may indicate thoughtful problem decomposition. Do not reward memorized prompt templates over genuine understanding of when and why to use specific techniques.

1.4.3. C. AI Output Evaluation (AOE) Behavioral Anchors

Table V: Behavioral Anchors for AI Output Evaluation (AOE)

Score	Label	Observable Behaviors
5	Exceptional	Identifies subtle logical, security, and design flaws in AI-generated code. Explains root causes with reference to underlying principles. Proposes improvements that address systemic issues, not just surface symptoms. Articulates confidence levels in AI outputs with reasoning.
4	Strong	Identifies most errors in AI-generated code, including non-obvious issues. Explains why errors occur and proposes corrections. Demonstrates balanced skepticism, neither blindly trusting nor blindly rejecting AI outputs.
3	Adequate	Identifies obvious errors (syntax, basic logic). May miss subtle design or security issues. Can explain some error causes. Asks clarifying questions about AI-generated code.

 doi:10.20944/preprints202603.0313.v1

Score	Label	Observable Behaviors
2	Developing	Identifies only syntax-level errors. Misses logical and design flaws. Tends to accept AI output if it compiles/runs without deeper inspection. Cannot articulate *why* a particular output might be problematic beyond surface-level observations.
1	Insufficient	Cannot distinguish correct from incorrect AI output. Does not attempt to review or question AI-generated code. No framework for evaluating AI output quality. Accepts AI outputs without any critical examination.
–	**Common Rater Errors**	Do not confuse excessive skepticism (rejecting everything) with genuine critical evaluation. Do not penalize candidates who initially accept an output and then revise their assessment upon deeper analysis, this shows intellectual honesty. Do not reward candidates who identify many minor issues (style, naming) while missing critical flaws (security, correctness).

1.4.4. D. System Orchestration (SO) Behavioral Anchors

Table VI: Behavioral Anchors for System Orchestration (SO)

Score	Label	Observable Behaviors
5	Exceptional	Designs end-to-end systems that integrate AI components with clear interfaces, error handling, and fallback strategies. Identifies which subproblems benefit from AI vs. manual implementation. Considers scalability, maintainability, and production constraints.

 doi:10.20944/preprints202603.0313.v1

Score	Label	Observable Behaviors
4	Strong	Designs functional multi-component systems with AI integration. Addresses error handling and defines clear component boundaries. Demonstrates pragmatic trade-off reasoning between AI-generated and manual components.
3	Adequate	Produces a basic system design incorporating AI tools. May lack error handling or fallback strategies. Can articulate component interactions but misses edge cases.
2	Developing	Produces incomplete or unrealistic system designs. Over-relies on AI for all components without considering integration challenges. Limited awareness of production constraints.
1	Insufficient	Cannot produce a coherent system design. Does not understand how to integrate AI components into larger workflows. No awareness of system-level concerns (reliability, scalability).
–	**Common Rater Errors**	Do not penalize unconventional but valid architectural approaches. Do not over-weight whiteboard diagram quality, focus on reasoning about component interactions and trade-offs. Do not confuse familiarity with specific technologies (e.g., Kubernetes) with system orchestration competency.

1.4.5. E. Meta-Reasoning (MR) Behavioral Anchors

Table VII: Behavioral Anchors for Meta-Reasoning (MR)

 doi:10.20944/preprints202603.0313.v1

Score	Label	Observable Behaviors
5	Exceptional	Demonstrates nuanced understanding of AI limitations, bias sources, and failure modes. Provides concrete examples from experience. Articulates governance strategies and stakeholder communication approaches. Reasons about long-term implications of AI tool adoption.
4	Strong	Identifies key AI limitations and ethical concerns. Proposes reasonable mitigation strategies. Can discuss trade-offs between AI speed and reliability. Aware of regulatory landscape (GDPR, EEOC).
3	Adequate	Aware of common AI limitations (hallucinations, bias). Can discuss basic ethical concerns. Limited depth in mitigation strategies. May recite talking points rather than demonstrate genuine reasoning.
2	Developing	Vague awareness of AI limitations. Cannot articulate specific risks or mitigations. Over-trusts AI or dismisses AI concerns without nuance.
1	Insufficient	No awareness of AI limitations or ethical concerns. Treats AI as infallible. Cannot discuss failure modes or responsible use.
–	**Common Rater Errors**	Do not confuse verbal fluency or articulateness with genuine ethical reasoning, focus on the substance of the argument, not the polish of delivery. Be aware that communication norms vary across cultures; measured or indirect responses may indicate thoughtful reasoning in some cultural contexts. Do not penalize candidates who acknowledge uncertainty; this may indicate intellectual honesty.

 doi:10.20944/preprints202603.0313.v1

Figure 2. AAIF Assessment Process Flowchart: Candidate application -> Stage 0 fundamentals gate (pass/fail) -> Stage 1 TF -> Stage 2 AOE -> Stage 3 SO -> Stage 4 MR -> Score aggregation -> Decision criteria -> Hire/No-hire decision. Feedback loops for calibration and interviewer training are shown. [Author-created diagram]

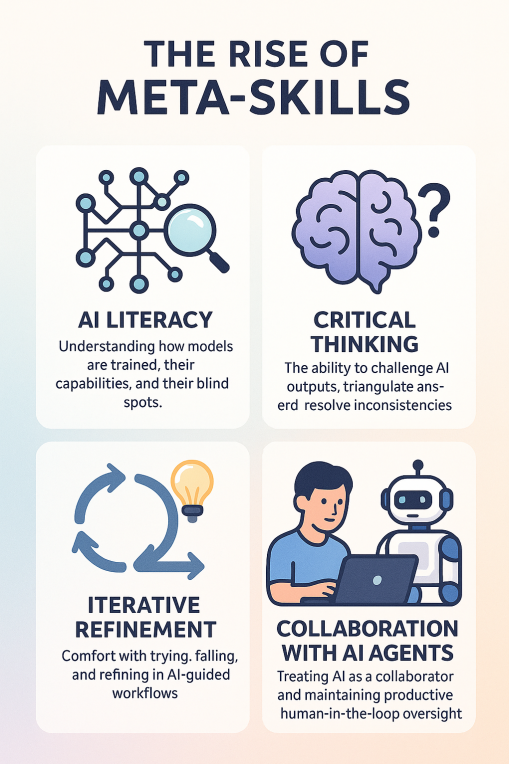

Figure 2. AAIF Assessment Process Flowchart

1.4.6. F. Scoring Process and Calibration

To achieve reliable scoring, organizations should implement:

1. **Interviewer certification**: Interviewers must demonstrate proficiency with AI development tools (scoring 4+ on TF and AOE) before evaluating candidates. *Bootstrapping strategy*: For the first cohort, organizations should identify 3–5 senior engineers with demonstrated AI tool proficiency and have them cross-evaluate each other using the BARS rubrics. Those achieving consistent scores (within 1 point of each other) become the founding calibration panel. Subsequent interviewers are certified by this panel.
2. **Calibration sessions**: Quarterly sessions where multiple interviewers independently evaluate the same candidate (recorded or live) and compare scores. Interviewers consistently deviating by 1+ point from consensus should receive retraining. After two consecutive quarters of deviation, interviewers should be removed from the panel until recalibration is achieved. *Recommended minimum*: Three calibration exercises per quarter, using anonymized recorded sessions to avoid demographic bias in calibration. This is reportedly analogous to calibration practices at major technology companies.
3. **Progressive rollout**: Begin with Stage 0 and Stage 1 for 3–6 months, measure outcomes, then incrementally add stages. Do not attempt full AAIF deployment simultaneously.
4. **Tool standardization**: Standardize on a defined set of AI tools per role and provide candidates 5–7 days advance notice to familiarize themselves, reducing measurement of tool familiarity rather than underlying competency. Specify tool versions and access dates for reproducibility.

 doi:10.20944/preprints202603.0313.v1

1.5. V. Risk Framework and Threats to Validity

1.5.1. A. Integrated Risk Analysis

AI-augmented interviews introduce complex ethical, technical, and procedural risks that traditional interviews do not face. We synthesize risks across five dimensions, drawing on algorithmic fairness [8], employment law [24,28], AI ethics literature [29], [30], and the CSCW literature on AI-mediated work [53], [54].

1) Bias and Discrimination

AI systems used in assessments may encode societal biases through training data, prompt design, or output evaluation criteria [8], [23]. Specific risks include:

- Cultural mismatch in prompt interpretation (prompts favoring Western engineering education conventions)
- AI model output preferences for certain problem-solving approaches or communication styles
- Underrepresentation of minority perspectives in AI-generated assessment content
- Language bias: AI tools perform best in English, disadvantaging non-native English speakers and developers in non-Anglophone countries. Research on LLM performance across languages [56] provides empirical evidence of significant performance disparities, with non-English languages showing substantially lower accuracy in code generation and explanation tasks

Proposed mitigations: (a) Audit all prompts and evaluation criteria for cultural bias before deployment. (b) Monitor scoring distributions across demographic groups using disparate impact analysis (four-fifths rule). (c) Maintain human-in-the-loop oversight for all final hiring decisions. (d) Conduct regular third-party bias audits. (e) Develop localized prompt sets and rubric variants for international deployment.

2) The Superficial Correctness Problem

AI tools can mask a candidate's lack of foundational understanding by generating correct-appearing code that the candidate cannot explain or maintain. This risk is mitigated by Stage 0 (fundamentals without AI) and Stage 2 (AI output evaluation), which require candidates to demonstrate understanding independent of AI-generated artifacts.

Candidate gaming strategies: Beyond superficial correctness, sophisticated candidates may employ gaming strategies including: rehearsed AI interaction sequences designed to appear spontaneous; memorized prompt templates applied without genuine understanding; and strategic staging of "discovery moments" where candidates pretend to identify issues they already knew about. AAIF addresses these risks through: (a) novel scenario variations that cannot be prepared in advance; (b) probe questions requiring candidates to explain their reasoning process ("Why did you choose that prompt structure?"); (c) unexpected AI outputs (intentionally providing different AI tools or versions than candidates may have practiced with); and (d) Stage 4's requirement for reasoning about novel scenarios rather than rehearsed talking points. However, the effectiveness of these countermeasures is conjectural and requires empirical validation.

3) Fairness Across AI Literacy Backgrounds

AI literacy correlates with socioeconomic status, educational background, geography, and age. Access to AI tools (ChatGPT Plus at $20/month, Copilot at $10–$39/month) is not universal. Requiring AI fluency as a hiring criterion may systematically disadvantage candidates from lower-income backgrounds, institutions without AI curriculum, non-English-speaking contexts, older developers, and developing economies.

Proposed mitigations: (a) Provide all candidates equal access to AI tools during assessment (pre-configured environment). (b) Offer optional pre-interview AI literacy orientation (30–60 minutes). We acknowledge that 30–60 minutes cannot close the gap between a developer with two years of daily AI tool usage and one with no prior exposure; the orientation addresses basic tool mechanics only, not

 doi:10.20944/preprints202603.0313.v1

deep fluency. (c) Weight Stage 0 (fundamentals) sufficiently to ensure strong developers new to AI tools are not unfairly excluded. (d) Include DEI metrics in validation studies. (e) Consider providing practice environments one week before the assessment.

4) Legal and Regulatory Compliance

Table VIII: Regulatory Compliance Requirements for AI-Augmented Hiring

Regulation	Jurisdiction	Key Requirements	Penalties
GDPR [31]	European Union	Right to explanation of automated decisions; consent required; data minimization	Up to 4% annual revenue
CCPA/CPRA [32]	California, USA	Opt-out rights; disclosure of profiling; data access rights	$2,500–$7,500 per violation
EEOC Guidelines [28]	United States	Disparate impact analysis; reasonable accommodation; documentation	Back pay, reinstatement, damages
NYC Local Law 144 [33]	New York City	Bias audits required for automated employment decision tools; public disclosure	$500–$1,500 per violation
EU AI Act [24]	European Union	Employment AI classified as "high-risk"; conformity assessment required	Up to 35M EUR or 7% revenue
DPDP Act [57]	India	Consent-based processing; data localization requirements; fiduciary obligations	Up to 250 crore INR
China AI Regulations [58]	China	Algorithm registration; fairness requirements; labeling of AI-generated content	Varies by regulation

Organizations must ensure AI-augmented interview processes comply with applicable regulations. International organizations should note that regulatory requirements vary significantly across jurisdictions, and compliance in one jurisdiction does not guarantee compliance in another.

5) Surveillance and Power Asymmetry Concerns

AI-augmented interviews may involve detailed logging of candidate-AI interactions: prompt history, iteration patterns, time allocation, and revision sequences. This level of behavioral monitoring raises surveillance concerns beyond standard interview observation.

Data minimization principle: Organizations should collect only the interaction data necessary for scoring decisions and delete detailed logs after the assessment period. Candidates should be informed of exactly what data is collected, how long it is retained, and their rights regarding access and deletion.

Power asymmetry: In hiring contexts, candidates may feel compelled to consent to AI tool usage and data collection because refusal may disqualify them. AAIF addresses this by: (a) providing

 doi:10.20944/preprints202603.0313.v1

clear disclosure of all data collection before the interview; (b) ensuring that the framework assesses competency with AI tools, not willingness to submit to surveillance; and (c) limiting interaction logging to what is necessary for scoring. However, the power asymmetry inherent in hiring contexts cannot be fully eliminated through framework design alone.

1.5.2. B. Ethical Considerations

Beyond legal compliance, organizations have ethical obligations [29,30]. We structure ethical analysis using the ACM Code of Ethics [59] and IEEE Ethically Aligned Design [60] frameworks:

- **Transparency** (ACM 1.3, IEEE P7001): Candidates should know they are being assessed on AI tool usage and understand evaluation criteria
- **Autonomy** (ACM 1.2): Candidates should not be penalized for choosing not to use AI tools if the role does not require them
- **Dignity** (ACM 1.1): Assessment should respect candidates' professional identities regardless of AI tool adoption level
- **Proportionality** (IEEE Principle 5): AI fluency requirements should be proportional to actual job requirements
- **Informed consent** (ACM 1.6): Candidates should understand what data is collected, how it is used, and their rights regarding that data

Societal impact note: How organizations assess technical talent shapes who enters the profession, who advances, and whose contributions are valued. The transition from traditional to AI-augmented interviews has potential to either widen or narrow existing inequities. We urge organizations implementing AAIF to actively monitor demographic outcomes.

1.5.3. C. Threats to Framework Validity

Following Messick's [52] unified validity framework and Wohlin et al. [34], we identify threats across six validity dimensions. The use of Messick's framework is particularly important because AAIF proposes a high-stakes assessment instrument where validity must be comprehensively evaluated. We emphasize that these threats apply to the framework as designed; empirical validation is required to determine their actual impact. All mitigations listed below are *proposed mitigations* that have not yet been implemented.

1) Construct Validity

Threat: AAIF stages may measure surface-level behaviors rather than underlying competencies.

- TF stage may conflate tool-specific knowledge with general AI orchestration ability
- AOE stage may reward excessive skepticism over balanced critical thinking
- SO stage may favor candidates with prior exposure to specific problem domains
- MR stage may confuse verbal fluency with genuine ethical reasoning

Proposed mitigations: (a) Multi-method assessment per competency. (b) Behavioral anchoring (Section IV). (c) Cross-validation with independent skill assessments. (d) Inter-rater reliability testing with target Cohen's kappa > 0.70 (threshold based on Landis and Koch's [61] classification of kappa 0.61–0.80 as "substantial agreement," with 0.70 representing the midpoint). (e) Discriminant validity analysis in Phase 2.

2) Criterion Validity

Threat: AAIF scores may not predict actual job performance. This is the most critical validity type for a selection instrument.

- The assessment function (Specification 1) may not capture the competencies that actually predict success in AI-augmented development roles

 doi:10.20944/preprints202603.0313.v1

- The weighted combination of four competency scores may not be the correct functional form for predicting performance
- Job performance itself is multidimensional and difficult to measure, introducing criterion contamination and deficiency

Proposed mitigations: (a) Phase 2 regression of AAIF scores on multiple job performance indicators (code review ratings, sprint velocity, bug rates, peer evaluations). (b) Phase 3 longitudinal tracking of KPI linkages. (c) Comparison with traditional interview predictive validity using the same criterion measures. (d) Target predictive validity $r > 0.35$–0.40, which we note may be ambitious given Sackett et al.'s [51] revised validity estimates. If achieved validity is lower (e.g., $r = 0.25$), the framework's utility must be evaluated relative to base rates and selection ratios, not rejected outright.

3) Content Validity

Threat: AAIF may not comprehensively cover required competencies.

- Framework may omit emerging competencies as AI tools evolve
- Four-stage model may not capture all dimensions (e.g., collaborative AI usage, production debugging with AI tools)
- Stage weights may not reflect actual job requirements

Proposed mitigations: (a) Annual competency review with expert panels. (b) Expert panel validation using Delphi methodology. (c) Empirical weight calibration against job performance data.

4) Convergent and Discriminant Validity

Threat: The four competencies (TF, AOE, SO, MR) may not be empirically distinct constructs. Strong AOE plausibly requires strong TF (toolchain fluency to generate outputs worth evaluating), suggesting potential construct overlap.

Proposed mitigations: (a) Confirmatory factor analysis in Phase 2 to test a four-factor model. (b) Correlation matrix among stage scores, inter-stage correlations above 0.85 would suggest insufficient discrimination. (c) Exploratory factor analysis if the four-factor model does not fit, allowing data to reveal the empirical competency structure.

5) Consequential Validity

Threat: Using AAIF for high-stakes hiring decisions may have unintended consequences, following Messick's [52] insistence that the social consequences of assessment use are a core validity dimension.

- AAIF could systematically disadvantage candidates from backgrounds with less AI tool access
- The framework could incentivize shallow AI tool proficiency over deep domain expertise
- Organizations might over-rely on AAIF scores, reducing holistic evaluation

Proposed mitigations: (a) Continuous adverse impact monitoring (disparate impact ratio > 0.80 required). (b) Periodic review of whether AAIF adoption has changed workforce composition along demographic dimensions. (c) Clear guidance that AAIF is one input to hiring decisions, not a sole determinant.

6) Face Validity and Ecological Validity

Threat: Candidates who perceive the assessment as irrelevant or unfair may underperform due to motivational factors. Interview conditions may not reflect real-world work.

- Time constraints (60–90 minutes per stage) may not reflect realistic project timelines
- Interview pressure may distort performance relative to actual work behavior
- AI tools provided may differ from those used in target organization

Proposed mitigations: (a) Include candidate experience surveys in Phase 1 to assess perceived fairness and relevance. (b) Use the same AI tools in interviews as on the job. (c) Design tasks based

 doi:10.20944/preprints202603.0313.v1

on actual organizational problems. (d) Provide adequate preparation materials to reduce interview-specific anxiety [7].

1.5.4. D. Validity Threat Summary

Table IX: Validity Threats and Proposed Mitigations

Validity Type	Primary Threat	Key Proposed Mitigation	Target Metric
Construct	Surface vs. underlying competency	Multi-method assessment + BARS	Inter-rater reliability kappa > 0.70
Criterion	Scores may not predict job performance	Regression on multiple performance indicators	Predictive validity r > 0.35
Content	Competency coverage gaps	Expert validation panels (Delphi)	Coverage gap analysis
Convergent/Discriminant	Competencies may not be distinct	Confirmatory factor analysis	Inter-stage r < 0.85
Consequential	Adverse demographic impact	Continuous adverse impact monitoring	Disparate impact ratio > 0.80
Face/Ecological	Perceived irrelevance; artificial conditions	Candidate experience surveys; realistic tasks	Candidate satisfaction > 3.5/5

1.6. VI. Domain Extensibility: Quantum Computing Illustration

1.6.1. A. Rationale and Scope

To illustrate that AAIF may generalize beyond general-purpose software development, we apply it to quantum computing, a domain where AI augmentation and deep theoretical expertise must coexist. This is a *demonstration of structural extensibility*, not empirical validation of quantum-adapted AAIF. No quantum computing interviews were conducted, and the domain-specific rubrics have not been reviewed by quantum computing experts.

Quantum computing was selected because: (a) AI hallucination risk is amplified, LLMs can confidently generate quantum circuits that violate fundamental principles (e.g., the no-cloning theorem [35]); (b) talent scarcity amplifies false positive cost [36]; and (c) platform evolution is rapid. We acknowledge that quantum computing, being a software-adjacent domain, may not stress-test extensibility as strongly as a domain with fundamentally different artifact types (e.g., hardware design, regulatory compliance writing).

1.6.2. B. Quantum AAIF Adaptation

The quantum adaptation maps the four AAIF competencies to domain-specific equivalents:

AAIF Competency	Quantum Adaptation	Key Domain-Specific Concern
Toolchain Fluency (TF)	Quantum Fundamentals + AI Learning (QF)	AI-generated quantum explanations may be subtly incorrect
AI Output Evaluation (AOE)	Quantum Algorithm Design (QAD)	Circuits may appear valid but violate quantum mechanical principles
System Orchestration (SO)	Hybrid System Orchestration (HSO)	Three-paradigm orchestration (classical + quantum + AI)
Meta-Reasoning (MR)	Critical Evaluation and Meta-Reasoning (CEM)	Quantum hype vs. NISQ reality assessment

 doi:10.20944/preprints202603.0313.v1

The quantum adaptation elevates the QF minimum threshold to 3.5 (vs. 3.0 for general AAIF) because quantum fundamentals are non-negotiable given the high cost of undetected physics violations.

Illustrative example (Stage 1 QF): "Debug this Qiskit circuit that claims to clone a quantum state", the CNOT gate creates entanglement, not cloning; a strong candidate identifies the no-cloning theorem violation while a weak candidate accepts the "cloning" characterization.

1.6.3. C. Extensibility Template

The quantum demonstration suggests a general adaptation template:

1. Identify domain fundamentals that AI cannot replace
2. Characterize domain-specific AI hallucination risks
3. Define hybrid system orchestration patterns for the domain
4. Establish domain-specific meta-reasoning challenges
5. Calibrate rubrics with domain experts

Candidate domains include computational biology, climate technology, advanced robotics, and cybersecurity. We note that some domains may require restructuring the four-competency model itself (e.g., adding a fifth competency or merging two existing ones).

1.7. VII. Discussion

1.7.1. A. Implications for Practice

For hiring managers and engineering leaders: AAIF suggests that interview processes should evolve to include AI tools as part of the assessment environment. Interview panels themselves must develop AI fluency, interviewers who cannot score 4+ on TF and AOE stages should not evaluate candidates on these competencies. See Section IV.F for bootstrapping strategies for the first certified panel.

For HR professionals and organizational psychologists: Job descriptions should evolve to include AI fluency alongside traditional requirements. The BARS-based rubrics (Tables IV–VII) are designed to be interpretable by HR professionals familiar with competency-based assessment. The risk framework (Section V) maps directly to existing compliance and DEI workflows.

For candidates: Career development should emphasize AI tool fluency alongside foundational skills. Stage 0 ensures that AI fluency does not substitute for computer science knowledge.

For researchers: The guiding research questions (Section II.D) and validation protocol (Section VII.C) define a research agenda spanning empirical software engineering, industrial-organizational psychology, HR analytics, and CSCW.

For policymakers: The AI augmentation of technical work has implications for STEM education, workforce training programs, and employment regulation.

Quick-start guide for practitioners:

- **2-week adoption**: Add a take-home challenge with AI tools permitted. Use the TF rubric (Table IV) to evaluate. Train interviewers to ask "How did you verify this AI output?" after every coding question.
- **3-month adoption**: Implement Stage 0 (fundamentals) and Stage 1 (toolchain fluency). Pilot Stage 2 (AI output evaluation) with 10–20 candidates. Measure interviewer agreement using the calibration process in Section IV.F.
- **12-month adoption**: Full AAIF implementation with calibration sessions. Begin collecting correlation data between AAIF scores and job performance for empirical weight calibration.

1.7.2. B. Framework Boundary Conditions

AAIF is not appropriate for all hiring contexts. Organizations should **not** adopt AAIF when:

1. **AI development tools are not used in daily work**: If the target role does not involve AI-augmented development, assessing AI collaboration competencies is not relevant and may introduce construct-irrelevant variance.

 doi:10.20944/preprints202603.0313.v1

2. **Legacy system maintenance roles**: Roles focused exclusively on maintaining legacy systems where AI tools are inapplicable or restricted should use traditional assessment methods aligned with actual job requirements.
3. **Security-constrained environments**: Defense, classified systems, and regulated healthcare environments where AI tool access is restricted or prohibited on the job should not assess candidates on competencies they cannot use.
4. **Early-career/internship hiring**: AI fluency expectations may be inappropriate for entry-level candidates. A simplified version focusing on Stage 0 and basic TF may be more appropriate, with the understanding that AI fluency will develop on the job.
5. **Very small organizations**: Organizations without resources for multi-stage interviews or calibration panels may find the full AAIF impractical. The quick-start guide (Section VII.A) offers abbreviated alternatives.
6. **Rapid hiring scenarios**: Contract roles or urgent staffing needs where 4–6 hours of assessment is impractical. A single-stage abbreviated assessment focusing on AOE may provide the highest signal-to-effort ratio.
7. **Cultural contexts without validation**: AAIF has been designed from a Western technology industry perspective. Deployment in non-Western contexts should be preceded by cultural adaptation and local pilot validation.

1.7.3. C. Limitations

This work has several limitations beyond the threats to validity discussed in Section V.

1. **No empirical validation**: The framework is entirely conceptual. No interviews have been conducted, no inter-rater reliability has been measured, no predictive validity has been established. All proposed KPI linkages, decision thresholds, and stage weights are conjectures. This is the most significant limitation.
2. **Single-author perspective**: The framework reflects one author's industry experience. The BARS rubrics were not developed using standard psychometric methods (critical incident technique, expert retranslation). Expert panel validation is a priority recommendation.
3. **Western-centric context**: The framework assumes Western technology industry practices, English-language AI tools, and US/EU regulatory frameworks. All AI tools referenced (Copilot, ChatGPT, Claude, CodeWhisperer) are from US-based companies. Regional AI tools with significant market share in non-Western markets (e.g., Tongyi Lingma, CodeGeeX in China) are not addressed. The MR behavioral anchors (Table VII) may reward communication styles aligned with Western professional norms; in cultures emphasizing indirect communication or deference to seniority, these rubrics may systematically disadvantage qualified candidates.
4. **Rapidly evolving landscape**: AI development tools and capabilities are changing rapidly. Framework elements may require frequent updating. The framework includes no explicit sunset mechanism or evolution pathway for when AI tools become so capable that certain stages (e.g., TF) become trivially easy or irrelevant.
5. **Scope boundaries**: AAIF focuses on software development roles. Extension to other knowledge work requires separate adaptation.
6. **Interview time burden**: The full framework requires 4–6 hours. Abbreviated versions are possible but their validity is unknown.
7. **Tool cost**: Providing enterprise AI tool access to candidates creates per-interview costs. At enterprise scale (10,000+ candidates per year), per-candidate tool licensing at $5–20 per session could cost $50,000–200,000 annually. Organizations should include tool costs in adoption planning and consider negotiating enterprise assessment licenses with AI tool providers.

 doi:10.20944/preprints202603.0313.v1

1.7.4. D. Proposed Empirical Validation Protocol

While this paper presents a conceptual framework, we recognize that empirical validation is essential. We consider this validation protocol a contribution of the paper. The protocol follows guidelines for empirical software engineering research [34,43] and instrument validation [44,45,46].

Phase 1: Pilot Study (Months 1–6)

- *Participants*: 50–100 candidates across 3–5 organizations (technology companies, financial institutions, and at least one non-Western organization).
- *Sample size justification*: For Cohen's kappa > 0.70 with a 5-category BARS scale, assuming moderate marginal distributions, Sim and Wright (2005) [62] recommend minimum n=40 per rater pair. We target n=50–100 to provide adequate precision for reliability estimation and to account for incomplete data.
- *Design*: Within-subjects; all candidates complete all AAIF stages plus a traditional interview track.
- *Primary outcomes*: Inter-rater reliability (target: Cohen's kappa > 0.70); candidate completion rates; interviewer satisfaction; time-to-administer.
- *Analysis*: Descriptive statistics; reliability analysis; thematic analysis of qualitative feedback from interviewers and candidates.
- *Deliverables*: Refined instruments, preliminary reliability data, feasibility assessment.

Phase 2: Controlled Study (Months 7–18)

- *Participants*: 300–500 candidates across 10–15 organizations, with deliberate inclusion of organizations in at least 3 countries and 2 industry sectors.
- *Sample size justification*: For regression analysis with target effect size r = 0.35 (revised from 0.40 to align with Sackett et al.'s [51] updated validity estimates), alpha = 0.05, and power = 0.80, G*Power analysis indicates minimum n = 62. We target n=300–500 to enable subgroup analyses (by role level, organization size, cultural context) and to provide adequate power for confirmatory factor analysis (recommended minimum n=200 for four-factor models).
- *Design*: Randomized comparison of AAIF vs. traditional interview tracks. Stratified by role level and organization size.
- *Statistical analysis*: Regression of AAIF scores on job performance; adverse impact analysis (disparate impact ratio > 0.80 required); confirmatory factor analysis; discriminant validity between AAIF stages. Bonferroni correction applied for multiple comparisons across the primary analysis family.
- *Feasibility risks*: Recruiting 10–15 organizations willing to randomize hiring is extremely challenging. Fallback design: quasi-experimental with matched comparison groups (organizations using AAIF vs. traditional methods), acknowledging reduced internal validity.
- *Deliverables*: Predictive validity evidence, adverse impact data, refined thresholds.

Phase 3: Longitudinal Tracking (Months 19–36)

- *Participants*: Follow-up of Phase 2 hires (n = 150–250).
- *Expected attrition*: Industry software developer turnover rates of 20–30% annually suggest 30–50% attrition over the 17-month tracking period. If attrition exceeds 40%, the effective sample may be insufficient for the planned hierarchical regression and survival analyses. Multiple imputation will be used for missing data under the missing-at-random assumption. If attrition is non-random (e.g., low AAIF scorers leave at higher rates), sensitivity analyses using pattern-mixture models will be conducted.
- *Outcomes*: Validate hypothesized KPI linkages.
- *Analysis*: Hierarchical regression; survival analysis for retention; mediation analysis.
- *Deliverables*: Empirically calibrated weights, validated KPI linkages.

Phase 4: Cross-Domain Validation (Months 24–48)

- *Domains*: 2–3 specialized domains (quantum computing, computational biology, cybersecurity).

 doi:10.20944/preprints202603.0313.v1

- *Participants*: 30–50 candidates per domain.
- *Power limitation*: n=30–50 per domain is likely underpowered for detecting moderate effect sizes in domain-specific analyses. Phase 4 results should be treated as exploratory, identifying domains where adapted AAIF shows promise for larger-scale validation.
- *Deliverables*: Domain adaptation guidelines, preliminary cross-domain validity evidence.

Ethical requirements: Informed consent from all participants; continuous adverse impact monitoring; anonymization of all candidate data; IRB or equivalent ethics committee approval before Phase 1; data retention and deletion policies specified in advance; right of withdrawal without consequence.

Adverse impact contingency: If empirical validation reveals that AAIF has greater adverse impact on protected groups than traditional interviews, the framework should not be abandoned outright but should be systematically analyzed to identify which components drive the disparate impact. Stage-level adverse impact analysis can identify whether specific stages (e.g., Stage 1 TF, which may reflect AI tool access disparities) are responsible. Remediation strategies include: adjusting stage weights, enhancing pre-interview orientation, providing extended practice periods, or redesigning specific rubric elements. If adverse impact cannot be reduced to acceptable levels (four-fifths rule compliance), deployment should be suspended pending redesign.

1.7.5. E. Future Directions

Beyond empirical validation, future work should address:

1. **Automated scoring support**: Developing AI-assisted evaluation tools that help interviewers apply BARS consistently
2. **Non-linear assessment models**: Exploring multiplicative or interaction-based scoring functions that capture competency dependencies
3. **Continuous assessment platforms**: Moving from point-in-time interviews to continuous evaluation environments
4. **International adaptation**: Developing AAIF variants for non-English-speaking contexts and non-Western organizational cultures
5. **Longitudinal role evolution**: Tracking how AI-augmented roles and required competencies evolve over time
6. **Systematic literature review**: Conducting a PRISMA-compliant systematic review of AI-augmented assessment practices

1.8. VIII. Illustrative Use Case: Kafka Performance Testing

1.8.1. A. Scenario Description

To demonstrate AAIF in action, we describe a hypothetical application to a realistic development task. We present two contrasting candidate profiles with explicit scoring to illustrate the framework's discriminative power. **This is an illustrative example, not empirical evidence of framework effectiveness.**

Scenario: A team must create a performance testing utility for a messaging system. The existing codebase publishes messages to Kafka, but the target system requires messages conforming to a specific schema.

1.8.2. B. Candidate A: Strong Performance (Overall Score: 4.35)

Stage 0 (Fundamentals): Score 4/5. Demonstrates solid understanding of distributed messaging concepts, CAP theorem trade-offs, and serialization formats. Minor gap in explaining exactly-once delivery semantics.

Stage 1 (TF): Score 5/5. Uses Copilot and Claude strategically: Claude for architecture analysis ("Analyze this Kafka publisher and identify the message format transformation needed to conform to [schema]. Generate a checklist of constraints and validation rules"), Copilot for implementation.

 doi:10.20944/preprints202603.0313.v1

Demonstrates deliberate tool selection, uses Claude for open-ended analysis and Copilot for code completion. Iterates on prompts purposefully, narrowing constraints when initial output is too broad.

Stage 2 (AOE): Score 4/5. Reviews AI-generated schema validation code and identifies that Copilot's implementation does not handle nested schema fields correctly. Explains the root cause (recursive schema traversal missing in the generated code). Manually corrects the validation logic. Misses a subtle thread-safety issue in the AI-generated producer code but identifies it when prompted.

Stage 3 (SO): Score 4/5. Integrates AI-generated components (schema validator, message transformer, error handler) into the existing Kafka infrastructure. Designs data flow between components and adds retry logic that AI tools did not suggest. Considers backpressure scenarios. Does not fully address monitoring/observability integration.

Stage 4 (MR): Score 4/5. Documents design decisions, acknowledges limitations of AI-generated test coverage (AI-generated tests covered happy paths but not failure modes). Discusses trade-offs between AI-accelerated development and thorough manual testing. Identifies that the AI-suggested retry strategy could cause message duplication in certain failure scenarios.

Overall: $A(k) = 0.25(5) + 0.25(4) + 0.30(4) + 0.20(4) = 4.25$. $\text{Min}(S_i) = 4$. **Decision: Hire (Mid-level)**, approaching Senior threshold.

1.8.3. C. Candidate B: Weak Performance (Overall Score: 2.55)

Stage 0 (Fundamentals): Score 3/5 (passes gate). Can explain basic Kafka concepts (topics, partitions, consumers) but struggles with consistency-availability trade-offs and cannot explain why message ordering guarantees differ between partitions.

Stage 1 (TF): Score 2/5. Uses only ChatGPT with vague prompts ("write me a Kafka performance tester"). Does not iterate on prompt quality when initial output is generic. Cannot explain why different AI tools might be better suited for different subtasks. Accepts the first output without evaluating whether it addresses the specific schema requirements.

Stage 2 (AOE): Score 2/5. Identifies a syntax error in AI-generated code but misses a logical flaw where the schema validation accepts invalid nested fields. When asked about the correctness of the AI-generated producer configuration, states "it looks fine" without verification. Cannot articulate a framework for evaluating AI output quality.

Stage 3 (SO): Score 3/5. Produces a basic system design but relies entirely on AI-generated components without considering how they interact. Does not address error handling for schema validation failures in the message pipeline. When asked about scaling the performance testing tool, provides a generic answer about "adding more instances" without addressing Kafka-specific scaling concerns (partition assignment, consumer group coordination).

Stage 4 (MR): Score 3/5. Aware that AI-generated tests "might miss things" but cannot articulate specific failure modes. When asked about the reliability of the AI-generated Kafka configuration, says "we should test it" without specifying what testing would involve. Cannot discuss scenarios where AI-generated performance benchmarks might be misleading.

Overall: $A(k) = 0.25(2) + 0.25(2) + 0.30(3) + 0.20(3) = 2.5$. $\text{Min}(S_i) = 2$. **Decision: No Hire** ($A(k) < 3.0$).

1.8.4. D. Scoring Observations

This contrasting example illustrates several framework properties:

1. **Discriminative power**: The BARS anchors differentiate Candidate A (strategic tool use, genuine critical evaluation) from Candidate B (passive tool use, surface-level evaluation) at each stage.
2. **Compensatory model in action**: Candidate B's adequate SO and MR scores (3/5) cannot compensate for weak TF and AOE scores (2/5), resulting in a No Hire decision.
3. **Stage 0 gate function**: Both candidates pass Stage 0, but Candidate B's weaker fundamentals predict weaker performance in AI-augmented stages.

 doi:10.20944/preprints202603.0313.v1

4. **Common rater error illustration**: An inexperienced interviewer might rate Candidate B's Stage 1 higher because the candidate produced output quickly, confusing speed with effectiveness (see Common Rater Errors, Table IV).

1.9. IX. Conclusion

Traditional technical interviews, centered on manual coding, algorithmic memorization, and whiteboard problem-solving, are increasingly misaligned with AI-augmented software development workflows. This paper proposed the AI-Augmented Interview Framework (AAIF), a structured five-stage methodology grounded in socio-technical systems theory and designed through Design Science Research methodology.

Our contributions include: (1) the AAIF framework with formal competency definitions, a configurable assessment function with psychometric analysis, and a Stage 0 fundamentals gate with defined pass/fail criteria; (2) behaviorally anchored evaluation rubrics with common rater errors, providing specific observable behaviors for each performance level; (3) illustration of framework extensibility through quantum computing application; and (4) an integrated risk framework addressing expanded validity threats (following Messick's unified framework), framework boundary conditions, and a detailed empirical validation protocol with power analyses.

We acknowledge significant limitations: the framework is conceptual and requires empirical validation. All proposed decision thresholds, stage weights, and KPI linkages are conjectures. The BARS rubrics were developed by a single author and have not undergone expert panel validation. The framework reflects Western technology industry assumptions and requires cultural adaptation for international deployment.

Despite these limitations, the core contribution, a systematic framework for assessing how developers collaborate with AI tools, addresses a genuine and growing gap in technical hiring practice. We hypothesize that organizations adapting their hiring practices to assess AI collaboration competencies will be better positioned to identify candidates who can think critically, orchestrate intelligently, and maintain human judgment in an AI-augmented world. This hypothesis requires empirical testing.

We invite the research community, spanning software engineering, industrial-organizational psychology, HR analytics, CSCW, and workforce policy, to engage with this framework through empirical validation, critique, and extension.

1.10. Replicability and Supplementary Materials

This section describes the materials necessary for independent replication and extension of this work, following best practices for empirical software engineering research [34,43].

Materials fully specified in this manuscript: - AAIF competency model with formal definitions and completeness discussion (Section III.C) - Behaviorally anchored rating scales with common rater errors for all four competencies (Tables IV–VII) - Assessment function with psychometric analysis (Section III.D) - Decision criteria with proposed thresholds and learning velocity operationalization (Specification 2) - Stage 0 pass/fail criteria with behavioral anchors (Section III.E) - Score aggregation and disagreement resolution protocol (Section III.D) - Scoring process and calibration protocol with bootstrapping strategy (Section IV.F) - Risk framework with expanded regulatory compliance matrix (Section V) - Expanded validity threats following Messick's framework (Section V.C) - Quantum computing adaptation illustration (Section VI) - Scored illustrative use case with contrasting candidate profiles (Section VIII) - Empirical validation protocol with power analyses and attrition handling (Section VII.D)

Minimal replication package (available at [repository URL]): - Rubric scoring templates (spreadsheet format) - Sample interview questions for each stage (2–3 per stage) - Worked scoring example from Section VIII - Interviewer quick-start guide

Materials to be developed and released as part of validation: - Comprehensive interviewer training manual - Candidate briefing documents - Standardized interview question bank (stratified by role level and domain) - Full scoring spreadsheet templates with automated weight calculation -

 doi:10.20944/preprints202603.0313.v1

Interviewer calibration exercise materials - Candidate experience questionnaire - Validation study data collection instruments

Upon commencement of the empirical validation protocol, all instruments and anonymized data will be archived on Zenodo with a persistent DOI and linked from a public GitHub repository.

1.11. Data Availability Statement

This paper presents a conceptual framework synthesized from published literature and theoretical foundations. No primary experimental data were collected. All referenced materials are cited within the manuscript. The AAIF framework rubrics, interview stage descriptions, and evaluation criteria are fully specified within the manuscript. A minimal replication package (rubric templates, sample questions, worked scoring example) is available at [repository URL]. A comprehensive replication package will be made available via GitHub + Zenodo archival DOI upon commencement of the empirical validation protocol. The author commits to open data and open materials practices for all future empirical work.

Funding: The author received no specific funding for this work.

Acknowledgments: Portions of this document were refined using AI writing assistants. All intellectual content, claims, and interpretations have been reviewed and verified by the author. This disclosure aligns with current journal policies on AI tool usage in manuscript preparation.

Conflicts of Interest: The author declares no competing interests.

[1] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, "The impact of AI on developer productivity: Evidence from GitHub Copilot," *arXiv preprint arXiv:2302.06590*, 2023. (Note: arXiv preprint; readers should verify peer-reviewed publication status.)

[2] Stack Overflow, "2024 Developer Survey: AI Tools," Stack Overflow, 2024. [Online]. Available: https://survey.stackoverflow.co/2024/ai

[3] GitHub, "GitHub Copilot: Your AI pair programmer," 2024. [Online]. Available: https://github.com/features/copilot

[4] OpenAI, "GPT-4 Technical Report," *arXiv preprint arXiv:2303.08774*, 2023.

[5] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[6] OECD, "The impact of AI on the workplace: Main findings from the OECD AI surveys of employers and workers," *OECD Social, Employment and Migration Working Papers*, 2023.

[7] M. Behroozi, S. Shirolkar, T. Barik, and C. Parnin, "Does stress impact technical interview performance?" in *Proc. 28th ACM Joint Meeting European Software Engineering Conf. Symp. Foundations Software Engineering (ESEC/FSE)*, 2020, pp. 481–492.

[8] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, "Mitigating bias in algorithmic hiring: Evaluating claims and practices," in *Proc. Conf. Fairness, Accountability, and Transparency (FAccT)*, 2020, pp. 469–481.

[9] A. Vaithilingam, T. Zhang, and E. L. Glassman, "Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models," in *Proc. ACM CHI Conf. Human Factors in Computing Systems Extended Abstracts*, 2022, pp. 1–7.

[10] E. Trist and K. W. Bamforth, "Some social and psychological consequences of the longwall method of coal-getting," *Human Relations*, vol. 4, no. 1, pp. 3–38, 1951.

[11] G. Baxter and I. Sommerville, "Socio-technical systems: From design methods to systems engineering," *Interacting with Computers*, vol. 23, no. 1, pp. 4–17, 2011.

[12] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989.

[13] Y. Engestrom, *Learning by Expanding: An Activity-Theoretical Approach to Developmental Research*. Helsinki: Orienta-Konsultit, 1987.

 doi:10.20944/preprints202603.0313.v1

[14] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[15] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[16] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view," *MIS Quarterly*, vol. 27, no. 3, pp. 425–478, 2003.

[17] K. Kuutti, "Activity theory as a potential framework for human-computer interaction research," in *Context and Consciousness: Activity Theory and Human-Computer Interaction*, B. A. Nardi, Ed. Cambridge, MA: MIT Press, 1996, pp. 17–44.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.

[19] Amazon Web Services, "Amazon CodeWhisperer," 2024. [Online]. Available: https://aws.amazon.com/codewh

[20] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, *et al.*, "Holistic evaluation of language models," *Transactions on Machine Learning Research*, 2023.

[21] F. L. Schmidt and J. E. Hunter, "The validity and utility of selection methods in personnel psychology: Practical and theoretical implications of 85 years of research findings," *Psychological Bulletin*, vol. 124, no. 2, pp. 262–274, 1998.

[22] A. I. Huffcutt and W. Arthur Jr., "Hunter and Hunter (1984) revisited: Interview validity for entry-level jobs," *J. Applied Psychology*, vol. 79, no. 2, pp. 184–190, 1994.

[23] J. Sanchez-Monedero, L. Dencik, and L. Edwards, "What does it mean to 'solve' the problem of discrimination in hiring? Social, technical, and legal perspectives from the UK on automated hiring systems," in *Proc. FAccT*, 2020, pp. 458–468.

[24] European Parliament, "Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (EU AI Act)," *Official Journal of the European Union*, 2024.

[25] S. Barke, M. B. James, and N. Polikarpova, "Grounded copilot: How programmers interact with code-generating models," *Proc. ACM Programming Languages (OOPSLA)*, vol. 7, no. 1, pp. 85–111, 2023.

[26] P. C. Smith and L. M. Kendall, "Retranslation of expectations: An approach to the construction of unambiguous anchors for rating scales," *J. Applied Psychology*, vol. 47, no. 2, pp. 149–155, 1963.

[27] G. P. Latham and K. N. Wexley, *Increasing Productivity Through Performance Appraisal*, 2nd ed. Reading, MA: Addison-Wesley, 1994.

[28] U.S. Equal Employment Opportunity Commission, "The Americans with Disabilities Act and the Use of Software, Algorithms, and Artificial Intelligence to Assess Job Applicants and Employees," EEOC Technical Assistance Document, 2022.

[29] B. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, and L. Floridi, "The ethics of algorithms: Mapping the debate," *Big Data and Society*, vol. 3, no. 2, 2016.

[30] J. Morley, L. Floridi, L. Kinsey, and A. Elhalal, "From what to how: An initial review of publicly available AI ethics tools, methods and research to translate principles into practices," *Science and Engineering Ethics*, vol. 26, pp. 2141–2168, 2020.

[31] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 (General Data Protection Regulation)," *Official Journal of the European Union*, 2016.

[32] California State Legislature, "California Consumer Privacy Act (CCPA) as Amended by CPRA," 2020.

[33] New York City Council, "Local Law 144: Automated Employment Decision Tools," 2021.

[34] C. Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell, and A. Wesslen, *Experimentation in Software Engineering*. Berlin: Springer, 2012.

[35] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, pp. 802–803, 1982.

 doi:10.20944/preprints202603.0313.v1

[36] McKinsey and Company, "Quantum computing: An emerging ecosystem and industry use cases," McKinsey Digital, Dec. 2021.

[37] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[38] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, pp. 625–644, 2021.

[39] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, 2019.

[40] V. Dunjko and H. J. Briegel, "Machine learning and artificial intelligence in the quantum domain: A review of recent progress," *Reports on Progress in Physics*, vol. 81, no. 7, 2018.

[41] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Physical Review Letters*, vol. 122, 2019, Art. no. 040504.

[42] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, pp. 195–202, 2017.

[43] P. Runeson and M. Host, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[44] D. Straub, "Validating instruments in MIS research," *MIS Quarterly*, vol. 13, no. 2, pp. 147–169, 1989.

[45] G. A. Churchill Jr., "A paradigm for developing better measures of marketing constructs," *J. Marketing Research*, vol. 16, no. 1, pp. 64–73, 1979.

[46] S. B. MacKenzie, P. M. Podsakoff, and N. P. Podsakoff, "Construct measurement and validation procedures in MIS and behavioral research: Integrating new and existing techniques," *MIS Quarterly*, vol. 35, no. 2, pp. 293–334, 2011.

[47] McKinsey Global Institute, "The economic potential of generative AI: The next productivity frontier," McKinsey and Company, Jun. 2023.

[48] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, "Challenges and applications of large language models," *arXiv preprint arXiv:2307.10169*, 2023.

[49] Boston Consulting Group, "The next decade in quantum computing—and how to play," BCG, Nov. 2018.

[50] Anthropic, "Claude: AI assistant," 2024. [Online]. Available: https://www.anthropic.com/claude

[51] P. R. Sackett, C. Zhang, C. M. Berry, and F. Lievens, "Revisiting meta-analytic estimates of validity in personnel selection: Addressing systematic overcorrection for restriction of range," *J. Applied Psychology*, vol. 107, no. 11, pp. 2040–2068, 2022.

[52] S. Messick, "Validity," in *Educational Measurement*, 3rd ed., R. L. Linn, Ed. New York: Macmillan, 1989, pp. 13–103.

[53] E. Kamar, S. Hacker, and E. Horvitz, "Combining human and machine intelligence in large-scale crowdsourcing," in *Proc. Int. Conf. Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2012, pp. 467–474.

[54] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, *et al.*, "Guidelines for human-AI interaction," in *Proc. ACM CHI Conf. Human Factors in Computing Systems*, 2019, pp. 1–13.

[55] K. Z. Gajos, D. S. Weld, and J. O. Wobbrock, "Automatically generating personalized user interfaces with Supple," *Artificial Intelligence*, vol. 174, no. 12–13, pp. 910–950, 2010.

[56] V. D. Lai, N. Ngo, A. P. B. Veyseh, H. Man, F. Dernoncourt, T. Bui, and T. H. Nguyen, "ChatGPT beyond English: Towards a comprehensive evaluation of large language models in multilingual learning," in *Proc. Findings of EMNLP*, 2023, pp. 13171–13189.

[57] Government of India, "Digital Personal Data Protection Act, 2023," *The Gazette of India*, 2023.

[58] Cyberspace Administration of China, "Interim Measures for the Management of Generative Artificial Intelligence Services," 2023.

[59] ACM, "ACM Code of Ethics and Professional Conduct," Association for Computing Machinery, 2018. [Online]. Available: https://www.acm.org/code-of-ethics

 doi:10.20944/preprints202603.0313.v1

[60] IEEE, "Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems," IEEE Global Initiative, 2019.

[61] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[62] J. Sim and C. C. Wright, "The kappa statistic in reliability studies: Use, interpretation, and sample size requirements," *Physical Therapy*, vol. 85, no. 3, pp. 257–268, 2005.