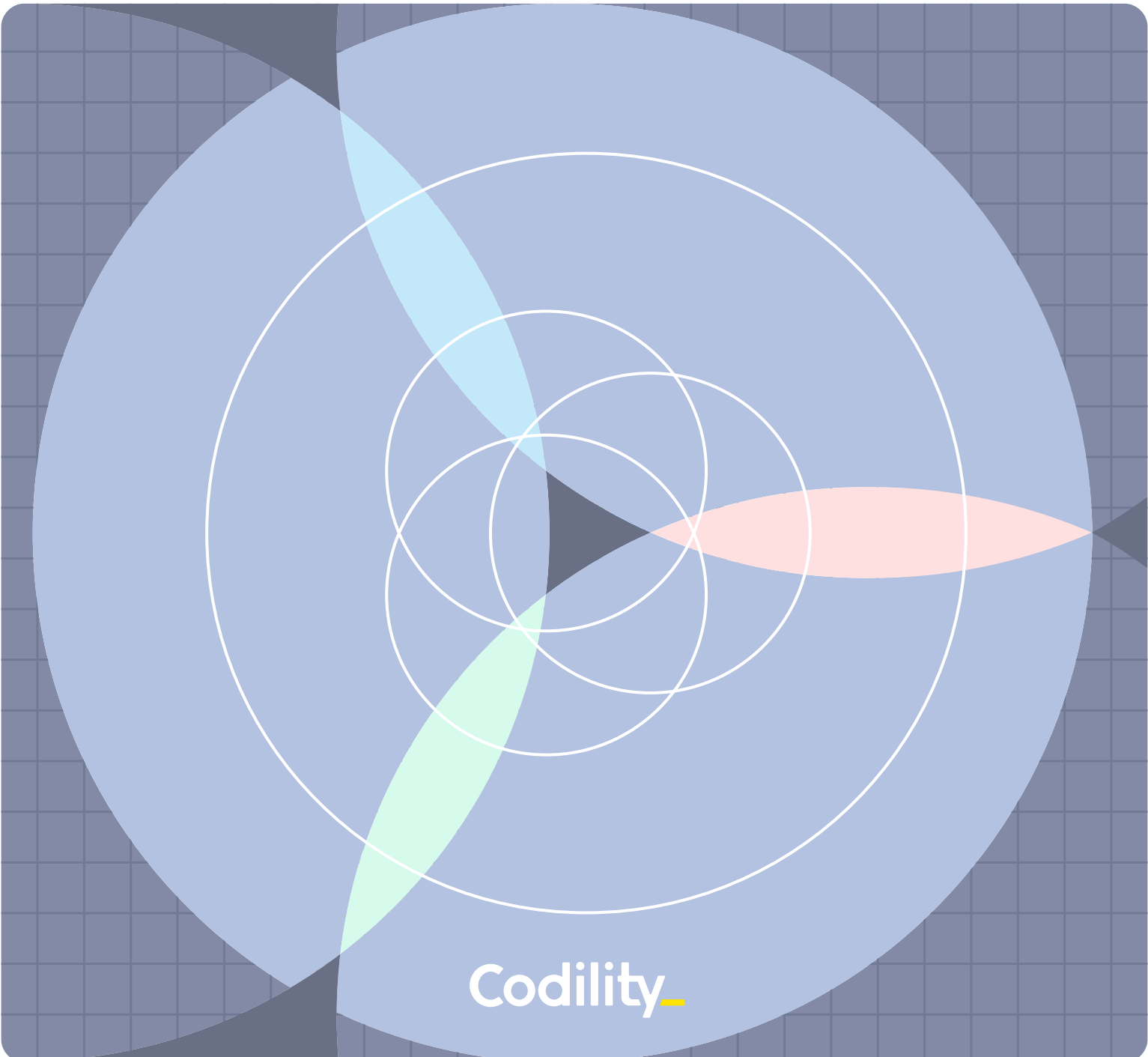


Development of **Codility's Engineering Skills Model (ESM 1.1)**

By Neil Morelli, PhD, Taylor Sullivan, PhD, and James Meaden, MA



Development of Codility's Engineering Skills Model (ESM 1.1)



TABLE OF CONTENTS

Project Overview	01
Phase 1: Skill Identification for a Preliminary Model	02
Technical Skills	02
Intrapersonal and Collaboration Skills	03
Problem-Solving Skills	03
Phase 2: SME Review	04
Step 1. Internal and External SME Sampling	04
Step 2. SME Review and Confirmation of Preliminary Model Skills	05
Phase 3: SME Feedback Review and Consolidation	06
Q4 2023 Work Analysis: AI-assisted Engineering Skills	06
Results and Final Model Summary	07
References	11



Neil Morelli, PhD
Chief I-O Psychologist



Taylor Sullivan, PhD
Sr. Dir. Product Insights
and Principal I-O
Psychologist



James Meaden, MA
Assessment Scientist

Project Overview

Building on nearly three years of research, Codility's team of tenured industrial-organizational (I-O) psychologists developed a conceptual framework that describes the skills required for success in technical roles. Codility's Engineering Skills Model (ESM) provides a holistic view of the important skills across an array of technical job families and can be used for effective talent assessment and development.

Codility's Engineering Skills Model (ESM) provides a holistic view of the important skills across an array of technical job families and can be used for effective talent assessment and development.

Focusing on a more holistic, skill-based approach to talent selection and development opens the door for more diverse contributions and perspectives. Ultimately, this will accelerate innovation and encourage diversity, equity, and inclusion in engineering.

Codility's mission is to unlock engineering potential. This potential comes in many forms. Focusing on a more holistic, skill-based approach to talent selection and development opens the door for more diverse contributions and perspectives. Ultimately, this will accelerate innovation and encourage diversity, equity, and inclusion in engineering.

For example, understanding the specific skills required in a job and then assessing job candidates on those skills will lead to identifying the best candidates for the role. In addition, identifying skill-specific strengths and opportunities for growth will enable the development of highly personalized learning paths for employees. Skills-based measurement also facilitates individual, team, and organization-level views of current skills and capabilities, which, in turn, informs strategic workforce planning.

The ESM offers a broad and easily adaptable approach to assessing skills across various engineering roles, industries, and organizations. Its simplicity and versatility make it a valuable tool for creating skill-based frameworks for engineering teams, allowing them to adjust skill targets easily as technology evolves. Moreover, the ESM promotes workforce diversity by avoiding the assumption that a specific set of skills is universally critical, providing a more inclusive approach to selecting qualified candidates for the job.

Generally, **cross-functional skills** are "developed capacities that facilitate the performance of activities that occur across jobs" (O*Net). **Technical skills** are the "developed capacities that are used to design, create, and/or correct applications of technological systems" (O*Net).

The ESM combines cross-functional, non-technical skills with technical skills. Notably, it includes a set of general problem-solving skills, such as computational thinking, that software engineers and other technologists need today and in the future. Collectively, nearly 70 job experts representing an array of engineering disciplines content-validated the skills in the ESM.

This report describes how Codility's I-O psychologists worked with subject matter experts (SMEs) to build a comprehensive, flexible, and inclusive skill model.

In the following sections, we describe the research and development of the ESM and present the skill and subskill names, skill definitions, and categories. Readers may request more detailed documentation as needed.

Phase 1: Skill identification for a preliminary model

An important first step when building talent management practices is identifying job-required skills or competencies.

Identifying job-relevant skills is a prerequisite for valid and defensible talent practices, as described by the U.S. federal government's Uniform Guidelines on Employee Selection Procedures [2] and the Principles for the Validation and Use of Personnel Selection Procedures [3] authored by the Society for Industrial and Organizational Psychology (SIOP) and adopted as policy by the American Psychological Association (APA)..

The research methodology employed by Codility's I-O psychologists to develop the ESM helps position it as a robust foundation for talent solutions, rendering them not only more defensible and job-related but also more actionable in terms of driving informed talent decisions. The process used to develop the categories and their contents is summarized below.

Technical Skills

Codility's I-O psychologists examined the language-agnostic skills required across job levels for six technical job families to build the technical skills section of the model.

These job families included backend developers, frontend developers, data scientists, quality assurance (QA) engineers, mobile developers, and developer operations (DevOps) professionals.



To cover all job families, the I-O psychology team gathered over 30 software engineering skill and competency frameworks from public and private organizations (e.g., IEEE Computer Society's Software Engineering Competency Model (SWECOM) [4] and Software Engineering Body of Knowledge (SWEBOK) [5], SEI Software Assurance Competency Model [6], and the Inclusive Engineering Framework [7]). They also reviewed professional training and academic curricula and published academic literature. A complete list of sources is available upon request.

The job information was reviewed and iteratively consolidated into a single streamlined skills taxonomy. The team then consulted over 25 internal Subject Matter Experts (SME) and 22 external SMEs to discuss and further refine the taxonomy and map it to the six job families listed previously. In focus groups designed for each job family, SMEs provided quantitative ratings of each skill's importance and whether or not the skill was required upon entry for various engineering job types. Based on these ratings, an initial list of technical skills was retained.

Intrapersonal and Collaboration Skills

The I-O psychologists also reviewed extant job information and skill-related materials (e.g., existing systems and software engineering practice overviews, training curricula, competency models, and research articles) to extract cross-functional, non-technical skills (i.e., soft skills) that are required for performance in technical roles. These skills are fundamental to an engineering career as they enable the application of technical skills to job-related behaviors.

The team identified an initial list of 16 non-technical skills, grouping them into two categories: intrapersonal skills required to function effectively in a professional environment and collaboration skills required to work well with others.

A group of seven Codility engineering managers with diverse specialties was then assembled

to review the list and share feedback on the accuracy and clarity of each skill's name and definition. SMEs also indicated whether the skill was essential for effective performance across representative software/system engineering roles. SMEs provided a yes/no rating for each skill and comments. Skills deemed essential by at least two-thirds of the SMEs were retained.

Problem-Solving Skills



Upon further review and consolidation of the job information and skill frameworks gathered (e.g., MITRE's Systems Engineering (SE) Competency Model [8], INCOSE Systems Engineering Competency Framework) [9], it was clear that including a problem-solving skills category was warranted. Initially the team identified six critical problem-solving skills: computational thinking, systems thinking, creative thinking, continual learning, design thinking, and product thinking. After further review, the team identified a high degree of overlap between design thinking and product thinking's definitions, so they combined these categories into a single "design/product thinking" category.

A key design requirement was that the ESM should contain skills essential for software engineering roles today and in the future. To help future-proof the model, Codility's I-O psychologists conducted additional research to identify skills that are particularly important when working with generative AI. The team reviewed the academic and best practice literature on generative AI use and prompt engineering ([see whitepaper for a summary](#)) and consulted with several thought leaders in this space.

From this review, they concluded that the skills required for generative AI/future engineering roles were largely already captured in the ESM. However, the team identified one key problem-solving skill that is additive to the model and critical for successfully using generative AI tools in technical roles: **computational thinking**. For a full discussion on computational thinking and its importance to software engineering, please review Wing (2006) [10].

Phase 2: SME Review

Internal and External SME Sampling

Once assembled, the I-O psychologists sampled 11 demographically diverse, experienced internal and external SMEs to review the preliminary model, including its skill and subskill names, definitions, and categories.

The sampling plan identified SMEs with a well-rounded representation of knowledge and experience within the target domain. As such, the plan included engineering managers and individual contributors across several representative job families employed at Codility, including frontend engineers, backend engineers, a DevOps engineer, and a data scientist. The plan also included Codility engineering leaders and experts, including Codility's founder, CTO, and two technical board members.

To complement the internal experts' perspective, the I-O psychologists invited external SMEs to offer a broader industry perspective. As a result, two external CTOs representing midmarket and enterprise technology organizations with strong engineering teams joined the expert panel review of the preliminary model.



SME Review and Confirmation of Preliminary Model Skills

Codility's I-O psychologists provided SMEs with a document describing the ESM and the following instructions:

Please read through the background information on the ESM's development and rationale (supplied as an attachment) and then use the attached rating sheet to provide feedback on the following:

1 Model Organization

*Do the skill categories make sense?
Is the model intuitive at first glance?
Does the format (i.e., the "wheel" layout) make sense?*

2 Skill Naming

*Are the skill names clear?
What edits would you suggest?*

3 Skill Definitions

*Are the skill definitions clear?
Is anything missing from these definitions?
Should anything be removed from the definitions?
Is anything unclear about the definitions?*

4 Gaps

*What's missing?
What have we miscategorized?
What have we overlooked?*



SMEs were reminded that ESM skill and subskill names, definitions, and category labels should be comprehensive enough to apply across roles and seniority levels yet configurable enough to reflect the required skill profile for a given role, team, or project.

By meeting these goals, the ESM should support an array of company-specific strategies, projects, and talent needs by offering a holistic view of an engineer's skills for specific roles.

Phase 3: SME Feedback Review and Consolidation

In August and September 2023, Codility's I-O psychologists iteratively reviewed and consolidated SME feedback from the preliminary model into a final model (ESM 1.0).

The team compiled all written and verbal feedback into proposed changes to skill names and definitions, as well as skills (and subskills) that should be consolidated or moved. In a series of iterative discussions, the I-O psychologists proposed changes and accepted, modified, or rejected those proposed changes after the group reached a consensus. Changes included restating vague definitions, consolidating overlapping skill areas, moving subskills to different skills, and adding skills that SMEs collectively agreed needed to be included.



After generating a final version of the model, the team re-engaged SMEs (including engineering leaders and experts) to confirm these model changes.

Q4 2023 Work Analysis: AI-assisted Engineering Skills

Generative artificial intelligence (genAI) is a new computing paradigm changing how engineers write, improve, and maintain software. The suite of tools and systems (e.g., large language models or LLMs) that power this technology is rapidly evolving, yet early data suggest that we'll likely see increased AI adoption and integration in the software engineering workflow. Thus, we conducted a work analysis to understand the skills software engineers need for success in an AI-driven world.

A complete summary of the work analysis methodology and findings is available upon request. We incorporated the findings of this analysis into ESM version 1.1.

The scope of the work analysis included general use cases and work activities where software engineers incorporate AI assistants into their day-to-day workflow. Working with SMEs who regularly perform these activities, we identified 21 unique work statements describing how engineers use (incorporate AI tools into performing daily work tasks), integrate (use AI tools to streamline an existing system), or (build and maintain AI models and systems directly) AI systems. From these statements, we identified several skills that were additive to the ESM 1.0. We confirmed with SMEs that these skills link to legitimate engineering work across several roles and seniority levels. Except for one skill (large-scale data analysis), we added these as subskills under a new skill label ("Using AI tools") within the ESM's technical skill category.

It is likely that these skills will change over time and, in some cases, blend with currently established ESM skills (e.g., 'prompt writing' may combine as another method or tool within the 'software construction' skill). However, given that genAI is a novel and powerful toolkit for technical jobs, it is important for engineering leaders and stakeholders to identify and assess the unique skills that contribute to using these tools effectively.

Results and Final Model Summary

Codility's ESM 1.1 (see Figure 1) employs a distinctive wheel-shaped design. This design highlights the equal importance given to technical skills, collaboration skills, and problem-solving skills while emphasizing the central role of intrapersonal skills. This presentation reinforces the growing need for engineers to have more well-rounded skill sets.

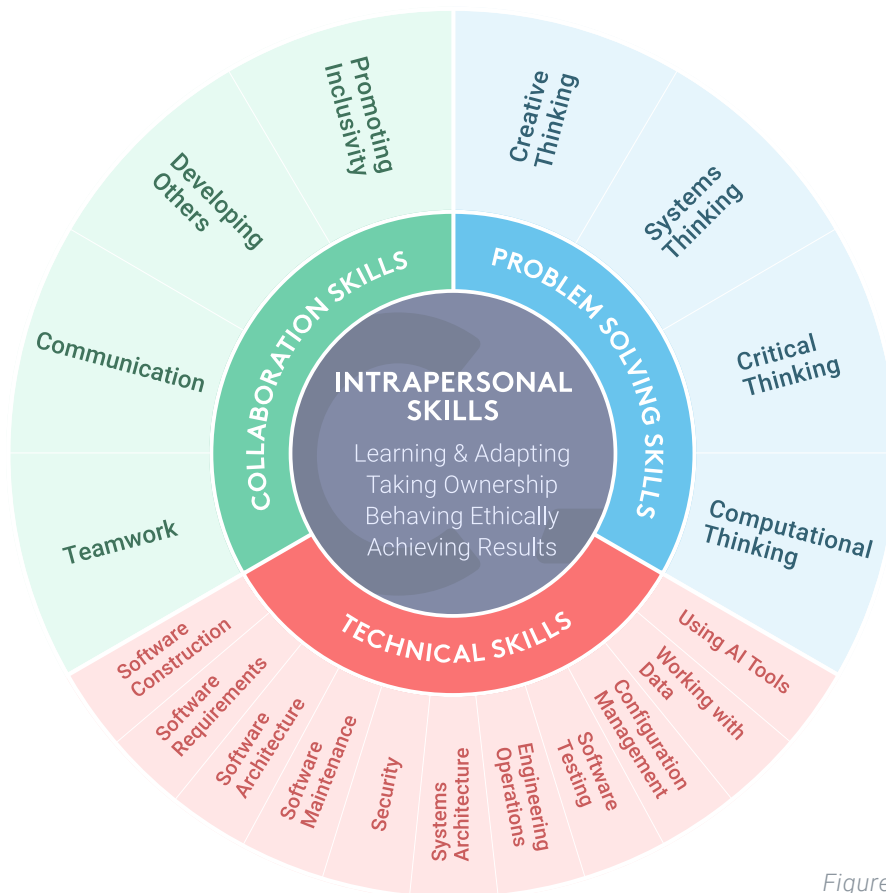


Figure 1. Codility's ESM 1.1

Table 1 provides a detailed overview, including definitions of each skill and a list of subskills accompanying each technical skill. We note that not all skills have accompanying subskills, and the listed subskills are not necessarily exhaustive.

The ESM 1.1 encapsulates SME feedback, job information in the form of competency models, training curricula, and professional standards, experienced I-O psychologist review, a second and third round of SME feedback and confirmation, and a posthoc comparison with the [beta V4 version of the IEEE's SWEBOOK](#).

Codility's I-O psychologists designed the ESM to be applicable to most technical roles, easily adjustable as target jobs and technologies change, and flexible to a range of proficiency levels and skill sets.

Codility's I-O psychologists designed the ESM to be applicable to most technical roles, easily adjustable as target jobs and technologies change, and flexible to a range of proficiency levels and skill sets. Still, Codility researchers will continually monitor the ESM by researching and gathering feedback to explore how changes in the field might signal changes in the skills required for effective practice.

Iteration is vital for any product, software, or system to stay relevant, and this will be a primary focus of Codility's research moving forward.

Iteration is vital for any product, software, or system to stay relevant, and this will be a primary focus of Codility's research moving forward.

Table 1: Final Model Skill Names and Definitions

TECHNICAL SKILLS		
SKILL	DEFINITION	SUBSKILLS
Working with Data	Examining and interpreting data through querying, visualization, reporting, and statistical/ML modeling to gain insights and make informed decisions. Includes data preparation, sourcing, exploration, and ongoing model tuning and maintenance.	<ul style="list-style-type: none"> - Data Querying - Data Visualization - Data Reporting - Data Preparation - Data Sourcing - Data Exploration - Statistical/ML Modeling - Model Tuning - Model Maintenance - Large-Scale Data Analysis
Software Construction	Developing software by implementing user interfaces, managing errors, asynchronous tasks, and memory, utilizing object-oriented principles, algorithms, and APIs, while supporting cross-platform compatibility, scripting, concurrency, event-driven and functional programming, recursion, linear and low-level programming and ensuring user accessibility.	<ul style="list-style-type: none"> - Basic Programming - Data Structures - UI Implementation - Object-oriented Programming - Error/Exception Handling - Asynchronous Operations - Cross-Platform Development - Leveraging APIs - Scripting - Concurrency - Resource Management - Algorithms - Event-driven Programming

- User Accessibility
- Functional Programming
- Recursion
- Linear Programming
- Low-Level Programming
- Responsive Design

Software Requirements

Collecting and communicating functional and non-functional requirements for a software product and translating them into technical designs, while considering user and business perspectives and the technical context.

Software Architecture

Applying principles that govern the fundamental organization of software, encompassing its components, properties, and relationships (e.g., design patterns, microservices, object-oriented design, and code structure).

- Design patterns

Software Maintenance

Continuously managing and improving software code to sustain its functionality, performance, and reliability.

- Code Optimization
- Code Readability
- Debugging

Security

Safeguarding software and systems through adherence to secure coding standards, implementing cryptographic techniques, and ensuring robust infrastructure security measures to protect against potential threats and vulnerabilities.

- Secure Coding Standards
- Cryptography
- Infrastructure Security

Systems Architecture

Determining system configurations, monitoring systems, employing architectural principles, and analyzing/debugging systems to ensure robust and efficient software systems.

- System Monitoring
- System Design
- System Analysis
- System Debugging

Engineering Operations

Planning, delivering, and controlling operations to streamline software development. Applying automation, continuous integration and delivery (CI/CD), deployment strategies, and infrastructure as code paradigms, while adhering to security and observability practices.

Software Testing

Applying software testing techniques, levels, and automation to ensure software functionality, reliability, and quality.

- Software Testing Techniques
- Quality Assurance
- Data Generation
- Software Design Review
- Test Automation
- Testing Levels

Configuration Management

Understanding and implementing processes and tools to track, control, and document individual components and assets within IT systems and software to ensure consistency, traceability, and reproducibility throughout their lifecycle (e.g., setting up build systems, makefiles, versioning software components, maintaining a comprehensive record of system configurations).

- Release Management

Using AI Tools

Utilizing and integrating artificial intelligence (e.g., large language model-based tools and systems) in various aspects of the engineering process. This includes crafting effective AI prompts, reviewing and refining AI-generated outputs, validating their reliability, understanding AI models and LLM frameworks, and managing all stages of machine learning project development from inception to maintenance.

- Prompt Writing
- AI proofreading
- AI output validation
- AI models
- Large Language Model (LLM) frameworks
- Machine Learning Operations



PROBLEM-SOLVING SKILLS

SKILL	DEFINITION
Systems Thinking	Understanding how different parts of a system interrelate and how changes in one part affect the whole. Taking into account the user's perspective and the broader context in which the system and its components will function together to fulfill its intended objectives and satisfy user requirements.
Creative Thinking	Generating original ideas, finding unique solutions to problems, and combining existing ideas in new ways.
Critical Thinking	Evaluating information and arguments through reasoned analysis to make well-informed decisions, solve problems, and facilitate learning. It involves skepticism, logical reasoning, and questioning assumptions and biases.
Computational Thinking	Formulating and solving problems using strategies common to computer science and programming. Breaking down complex problems into smaller, more manageable components and expressing solutions in a language that a computer can execute.

COLLABORATION SKILLS

SKILL	DEFINITION
Teamwork	Working effectively with others on shared tasks (e.g., co-creating, pair programming). Resolving conflicts through facilitation and reconciliation. Actively participating in team discussions, assisting other members in addressing problems, and emphasizing the team's collective interests. Readily sharing information and knowledge with the team. Acknowledging and advocating for others' contributions.
Communication	Conveying information, ideas, thoughts, or feelings effectively and clearly to others through various means, such as verbal, written, non-verbal, or digital channels. Listening actively, expressing oneself coherently, adapting communication style to different audiences, and comprehending and interpreting information from others accurately.
Developing Others	Developing, guiding, teaching, coaching, and mentoring others toward achieving personal and professional goals. Delivering effective feedback on an ongoing basis to improve performance.
Promoting Inclusivity	Respecting, appreciating, and encouraging contributions and participation from all individuals, including those with diverse backgrounds and cultures. Fostering an inclusive mindset that evaluates perspectives, practices, and products from different cultural perspectives.

INTRAPERSONAL SKILLS

SKILL	DEFINITION
Learning and Adapting	Responding to volatility, uncertainty, complexity, and ambiguity (VUCA) with curiosity and flexibility. Seeking new ideas and approaches by applying learning to work and discarding ineffective methods. Adapting plans, priorities, and goals in response to difficult conditions, tight deadlines, obstacles, or setbacks.

Taking Ownership	Taking responsibility for one's decisions and behaviors, holding oneself accountable for meeting work objectives within set timeframes, and taking action without waiting for direction.
-------------------------	--

Behaving Ethically

Exhibiting integrity, trustworthiness, honesty, and fairness in one's decisions and actions. Demonstrating global, social, intellectual, and technological responsibility.

Achieving Results

Setting goals for personal and group accomplishment, monitoring progress toward goal attainment, and working to meet or exceed goals. Persisting to accomplish tasks. Being outcome-driven versus output-driven.

Conclusion

In summary, the ESM 1.0 developed by Codility's Industrial-Organizational (I-O) psychology team represents a significant advancement in technical talent assessment and development. It helps organizations place skills at the forefront of their talent practices and offers tangible advantages such as more precise candidate selection based on job-relevant skills, tailored learning pathways for employees, and insights for strategic workforce planning. Its adaptable nature and inclusive approach make it a versatile tool that stands to promote diversity and innovation within organizations.



Neil Morelli, PhD
Chief I-O Psychologist



Taylor Sullivan, PhD
Sr. Dir. Product Insights and
Principal I-O Psychologist



James Meaden, MA
Assessment Scientist



[1] National Center for O*NET Development. Browse by Cross-Functional Skills. O*NET OnLine. Retrieved September 22, 2023, from <https://www.onetonline.org/find/descriptor/browse/2.B>

[2] Equal Employment Opportunity Commission, Civil Service Commission, Department of Labor, and Department of Justice. (1978). Uniform guidelines on employee selection procedures. Federal Register, 43(166), 38290-38315.

[3] Society for Industrial Organizational Psychology. (2018). Principles for the validation and use of personnel selection procedures (5th ed). Bowling Green, OH: American Psychological Association. Accessed at <https://www.apa.org/ed/accreditation/about/policies/personnel-selection-procedures.pdf>

[4] Fairley, R. E. (2014). A Software Engineering Competency Model (SWECOM). IEEE Computer Society Press.

[5] Bourque, P. and Fairley, R. E. (2014). Guide to the software engineering body of knowledge (SWEBOK(R)): Version 3.0. IEEE Computer Society Press.

[6] Hilburn, T., Ardis, M., Johnson, G., Kornecki, A., & Mead, N. (2013, March 11). Software Assurance Competency Model. (Technical Note CMU/SEI-2013-TN-004). Retrieved September 22, 2023, from <https://doi.org/10.1184/R1/6583991.v1>.

[7] Bonfield, D. (n.d.). Inclusive Engineering Framework. Retrieved September 22, 2023 from <http://www.inceng.org/inclusive-engineering-framework.html>.

[8] Metzger, L. S., & Bender, L. R. (2007). MITRE Systems Engineering (SE) competency model version 1.13 E. The MITRE Corporation.

[9] Presland, I., Beasley, R., Gelosh, D., Heisey, M., & Zipes, L. (2018). INCOSE systems engineering competency framework. INCOSE Technical Product Reference: INCOSE-TP-2018-002-01.0.

[10] Wing, J. M. (2006). [Computational thinking](#). Communications of the ACM, 49(3), 33-35. DOI: 10.1145/1118178.1118215