hr | avatar
Illuminating Talent

Test Results and Interview Guide

Candidate: **Elizabeth Wantsajob**
Assessment: GIT - Usage and Concepts
Completed: July 1, 2026
Prepared for: Sara Maple
Example Company

What's Included

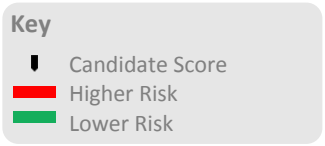
- Overall Score
- Competency Summary Table
- Comparison Matrix
- Detailed Competency Results with Interview Guide

Important Note: The GIT - Usage and Concepts assessment measures key factors related to high performance and tenure in this job. Attribute types measured vary by test, but can include cognitive ability, skills, knowledge, personality characteristics, emotional intelligence, and past behavioral history. This report includes a one page summary, followed by detailed results with an embedded interview guide. Note that these results should always be used as a part of a balanced candidate selection process that includes independent evaluation steps, such as interviews and reference checks.

Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com GIT - Usage and Concepts July 1, 2026	82	

The candidate exhibits a comprehensive and advanced understanding of GIT concepts, syntax, and workflows, including proficiency with repositories, branching and merging strategies, remote operations, staging, conflict resolution, history inspection, and collaborative development practices. This level of knowledge indicates the candidate can confidently and independently apply GIT in complex, real-world development environments with minimal guidance.



Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Branches and Merging	92	
Git Configuration and File Management	81	
Repositories and Commits	67	
Undoing and Reverting Changes	88	
Viewing History and Comparing Changes	68	
Working with Remotes	97	

Comparison

Percentile scores indicate how the candidate compares to other test-takers within various groups. The candidate scored equal to or better than the fraction of test-takers indicated by the percentile.

Test-Taker Group	Percentile	0 10 20 30 40 50 60 70 80 90 100
Global	82nd	
North America	68th	
United States	68th	
Example Company	76th	

Artificial Intelligence (AI) Generated Scores

This table includes one or more scores derived from a large language model AI query. AI-derived scores are non-deterministic. That is, they are not precisely repeatable. Therefore, these scores should always be treated as supplementary information and should never be used exclusively or compared to hard cutoff values.

Estimated Value	Score	Confidence	Interpretation
Knowledge, Skills, and Abilities Summary	-	-	Summary Points (AI): • (Generic Text for Sample Report) Strong performer in Drag and Drop Files tasks, indicating comfort with file management and basic computer interactions. • Demonstrates solid numerical accuracy in Recognizing and Confirming Numbers, a valuable asset in detail-oriented roles. • Moderate overall performance in Analytical Thinking and Attention to Detail, with adequate grammar skills but room for improvement. • Struggles with Reading and Analyzing Problems, which may limit effectiveness in roles requiring critical reading and complex problem-solving. • Lowest performance in Navigating Between Screens, suggesting difficulty with multi-screen software workflows that could impact productivity in computer-intensive roles. Narrative (AI): Elizabeth Wantsajob demonstrates a mixed profile of knowledge, skills, and abilities across the assessed competencies. Elizabeth shows a strong aptitude in Drag and Drop Files, performing well on this technical task and suggesting she is comfortable with this type of computer interaction. This is a notable strength that would translate well into roles requiring file management and basic computer navigation tasks. In the area of Analytical Thinking and Attention to Detail, Elizabeth performs at a moderate level. She demonstrates solid ability in Recognizing and Confirming Numbers, which suggests she is careful and accurate when working with numerical data — a valuable skill in detail-oriented work environments. Her Grammar performance is adequate but leaves room for improvement, indicating she may occasionally make written communication errors. Her weakest area within this competency is Reading and Analyzing Problems, where she struggled to consistently interpret and work through written problem scenarios. This may impact her effectiveness in roles that require critical reading, written comprehension, or complex problem-solving. Elizabeth's most significant area for development is Navigating Between Screens, where she scored considerably lower than the other competencies. This suggests she may have difficulty efficiently moving through software interfaces or multi-screen workflows, which could slow productivity in roles that rely heavily on navigating computer applications or data entry systems. Overall, Elizabeth brings some useful technical strengths, particularly in file management and numerical accuracy, but would benefit from targeted development in software navigation and analytical problem-solving to be fully effective in roles that demand these skills. Computed on: April 2, 2026, 11:09:49PM EDT

Detail

Candidate:	Elizabeth Wantsajob, beth.wantsajob@gmail.com
Assessment:	GIT - Usage and Concepts
Authorized:	July 1, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
Started:	July 1, 2026, 5:12:31PM EDT
Completed:	July 1, 2026, 5:12:31PM EDT
Overall Score:	82

Knowledge and Skills Detail

This section contains a list of job-related knowledge areas and skills that have been evaluated. Low scores in these areas often indicate that additional learning may be required before top performance can be achieved.

Detail | Interview Guide

Branches and Merging

Score: 92

Description:

Covers how to create, switch between, and delete branches, as well as how to merge branches together. Includes understanding of how to identify and resolve merge conflicts that arise when changes in different branches affect the same content.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate exhibits an advanced and comprehensive mastery of Git branching and merging concepts. They are highly proficient in all aspects of branch management and conflict resolution, and can be relied upon to handle complex branching workflows with minimal oversight.

Describe what a merge conflict is, how it occurs, and what steps you would take to resolve one.

1	2	3	4	5
Cannot explain what a merge conflict is or how to resolve it.		Correctly identifies that conflicts occur when the same lines are changed in both branches but gives vague resolution steps.		Explains conflict causes clearly, describes editing conflict markers, staging resolved files, and completing the merge with a commit.

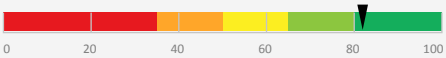
What is a branch in Git, and can you describe how you would create a new branch and switch to it?

1	2	3	4	5
Cannot define a branch or recall the commands to create and switch to one.		Correctly names 'git branch' and 'git checkout' or 'git switch' but cannot explain what a branch represents.		Clearly defines a branch as a pointer to a commit, explains 'git branch', 'git checkout -b' or 'git switch -c', and their purpose.

Detail

Git Configuration and File Management

Score: 81



Description:

Covers how to configure Git settings for a user or repository using git config, and how to control which files Git tracks using a .gitignore file. Includes setting up user identity, understanding configuration scope, and writing patterns to exclude files or directories from version control.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate exhibits a comprehensive and advanced understanding of Git configuration and file management. They are proficient in applying configuration settings across all scopes, managing user identity, and crafting precise and sophisticated gitignore patterns to control version tracking effectively.

Interview Guide

How would you use 'git config' to set up your name and email, and what is the difference between setting something globally versus locally for a single repository?

1	2	3	4	5
Cannot recall the 'git config' command or explain the difference between global and local scope.		Correctly names the command and flags but cannot clearly explain when to use global versus local configuration.		Explains '--global' applies to all repos and '--local' overrides for a specific repo, with practical reasons for using each.

What is a .gitignore file and why would you use one? Can you give an example of something you might put in it?

1	2	3	4	5
Cannot explain what .gitignore is or provide any valid example of its use.		Correctly explains .gitignore excludes files from tracking but cannot write a valid pattern or explain common use cases.		Explains purpose clearly, provides valid examples like '*.log' or 'node_modules/', and mentions it should be committed to the repo.

Detail

Repositories and Commits

Score: 67

Description:

Covers the core building blocks of Git, including how to initialize and clone repositories, stage changes using the index, and create commits with meaningful messages. Also includes understanding of key concepts like HEAD, tracked files, and the three-component Git architecture: working directory, staging area, and repository.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate exhibits a solid and competent understanding of GIT concepts, including branching, merging, staging changes, working with remotes, and navigating commit histories. Minor gaps may exist in advanced areas such as complex conflict resolution, tagging strategies, or fine-grained configuration, but overall proficiency is sufficient for most collaborative development workflows.

Interview Guide

What is the difference between the working directory, the staging area, and the repository in Git, and how do changes move between them?

1	2	3	4	5
Cannot distinguish between the three areas or describe how changes move between them.		Correctly identifies all three areas but gives a vague or incomplete explanation of how changes flow.		Clearly explains all three areas, the role of 'git add' and 'git commit', and how HEAD tracks the current state.

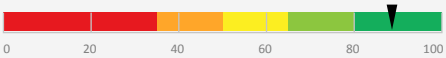
Can you walk me through what happens, step by step, when you make a change to a file and want to save that change in Git? What commands would you use and why?

1	2	3	4	5
Cannot describe staging or committing; confuses Git steps with saving files normally.		Mentions 'git add' and 'git commit' but cannot clearly explain the staging area's role.		Clearly explains working directory, staging with 'git add', committing with a message, and HEAD's role.

Detail

Undoing and Reverting Changes

Score: 88



Description:

Covers the various ways to undo or reverse changes in Git, including reverting a commit by creating a new one that undoes it, and resetting a branch to a previous state. Includes understanding the practical differences between revert and reset and when each is appropriate.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate demonstrates a strong and comprehensive understanding of undoing and reverting changes in Git. They are well-versed in both reverting commits and resetting branches, and can confidently apply the correct approach based on the context and requirements of a given situation.

Interview Guide

What is the difference between 'git revert' and 'git reset', and what are the risks of using 'git reset' on a branch that others are working on?

1	2	3	4	5
Cannot distinguish between revert and reset or is unaware of risks to shared branches.		Correctly distinguishes revert from reset but gives only a vague explanation of the risks to shared history.		Clearly explains revert adds a new commit, reset moves HEAD back, and describes how reset rewrites history causing conflicts for collaborators.

If you committed something by mistake, what are some ways Git allows you to undo or fix that?

1	2	3	4	5
Cannot name any Git command for undoing changes or describes only non-Git workarounds.		Mentions 'git revert' or 'git reset' but cannot explain the difference or when to use each.		Explains 'git revert' as safe for shared history and 'git reset' for local changes, with clear use case distinctions.

Detail

Viewing History and Comparing Changes

Score: 68

Description:

Covers how to use Git tools to inspect the state of a repository over time. Includes using the log to browse commit history and using diff to compare changes between commits, branches, or the working directory and staging area.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates a solid working knowledge of Git history inspection and change comparison tools. They are generally proficient in using commit logs and diff commands across commits, branches, and the working directory, with only minor gaps in more advanced usage.

Interview Guide

How would you use 'git diff' to understand what has changed in your code, and what are the different ways you can use it?

1	2	3	4	5
Cannot describe what 'git diff' does or how to use it in any context.		Correctly states 'git diff' shows changes but only describes one use case, such as unstaged changes.		Explains comparing working directory to staging, staging to last commit, and differences between two branches or commits.

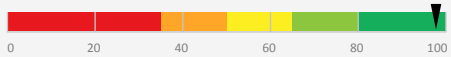
If you wanted to see a list of past commits in your repository, what command would you use, and what information does it show you?

1	2	3	4	5
Cannot recall 'git log' or describe what commit history information looks like.		Names 'git log' but cannot describe the output fields such as commit hash, author, date, or message.		Names 'git log', describes its output clearly, and mentions useful options like '--oneline' or '--graph'.

Detail

Working with Remotes

Score: 97



Description:

Covers how to connect a local repository to a remote, and how to synchronize changes between them. Includes pushing local commits to a remote, pulling or fetching changes from a remote, and understanding the difference between fetch and pull in collaborative workflows.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate exhibits a strong and comprehensive understanding of working with remotes in Git. They are highly proficient in connecting local repositories to remotes and confidently managing synchronization through push, pull, and fetch operations. They demonstrate a clear grasp of the distinctions between fetch and pull, making them well-suited for collaborative development environments.

Interview Guide

What is the difference between 'git fetch' and 'git pull', and when would you choose to use one over the other?

1	2	3	4	5
Cannot distinguish between fetch and pull or uses the terms interchangeably.		Correctly states that pull fetches and merges, but cannot articulate when to prefer fetch over pull.		Clearly explains fetch downloads without merging, pull fetches and merges, and describes scenarios favoring each approach.

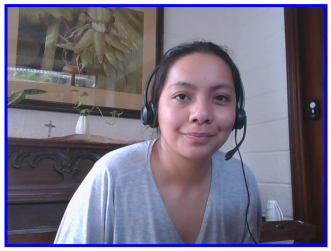
How would you share your local commits with others using Git? What command or commands would you use?

1	2	3	4	5
Cannot name 'git push' or describe the concept of a remote repository.		Mentions 'git push' but cannot explain what a remote is or how to set one up.		Explains remotes, 'git remote add', 'git push', and the relationship between local and remote branches clearly.

Identity Confirmation Photos

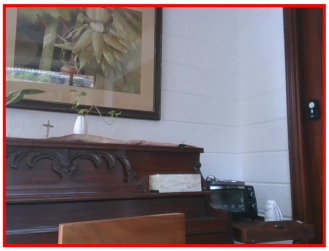
The following photos of the candidate and any identification were uploaded during the assessment session.

Photo Analysis Results	
- Risk:	Medium risk of cheating based on image inconsistencies
- Percent match among processed faces	100%
- Total images processed	17
- Total images with valid faces	14 (82%)
- Total pairs of faces compared	13
- Pairs in which faces matched	13 (100%)

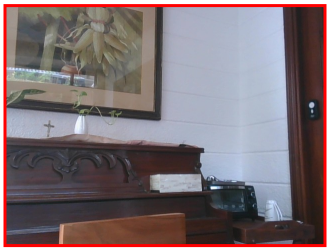


Pre/Post-Test Photo

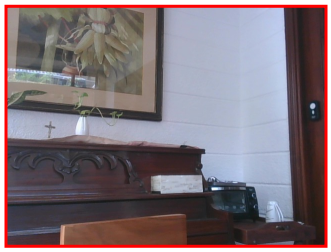
ID Photo



In-Test Error Detected (No Face Detected)

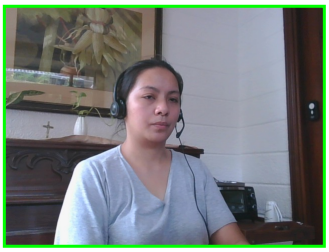


In-Test Error Detected (No Face Detected)

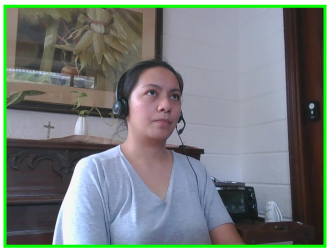


In-Test Error Detected (No Face Detected)

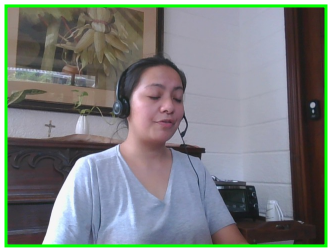
In-Test Photo



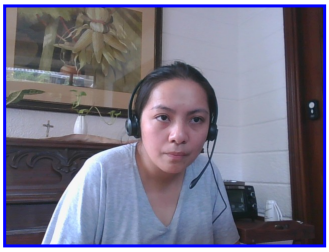
In-Test Photo



In-Test Photo



In-Test Photo



Pre/Post-Test Photo

Resume or CV

Summary Updated on

Motivated career professional with extensive experience in office administration and management. Proven track record of improving efficiency, reducing costs, and enhancing office operations through strategic initiatives and technology implementation.

Objective

I am seeking a role where I can use my many skills and my exceptional judgment and empathy for customers to make a difference to a growing company.

Education

- Associate of Applied Science in Office Administration, Portland Community College, 2020

Experience

- General Office Clerk, Paramount Office Management, 09/2023 – Present
- Administrative Assistant, Global Enterprises Inc., 04/2021 – 08/2023
- Administrative Assistant, Innovative Business Solutions Ltd., 07/2019 – 03/2021

Other Qualifications

- Microsoft Office Specialist (MOS) Certification
- Certified Administrative Professional (CAP)
- International Association of Administrative Professionals (IAAP) Certification

Report Preparation Notes

- Hiring decisions should never be based on a single source of information. The most effective use of this assessment report is as a part of a multi-faceted program of candidate evaluation that includes resume review, interviews, and reference checks.
- Overall vs Percentiles Scores: The overall score reflects the success in the test, based on the mean (average) and standard deviation of the test scores. The percentile score reflects the percentage of test-takers who scored equal or below this overall score. We recommend you use the Overall Score as your primary evaluation criteria. However, percentile scores can often be useful in comparing specific candidates against one another and with a group, such as for test takers in a certain organization or within a certain account.
- Note that comparison information is calculated based on completed instances of this assessment at that time the assessment is scored. As additional instances are completed, the comparative data may change. You can always update a report to the current values by clicking on 'Recalculate Percentiles' within the online results viewing pages at www.hravatar.com.
- Most competency scores are norm-based, which means that they can be interpreted in terms of their distance from the average or mean score. For all scales, a score equal to the mean receives a score of 65 and scores above and below this value are set so that a score change of 15 equals one standard deviation.
- For linear competencies, higher is better across the entire scale. For these scales a score between 65 and 80 (light green) represents 0 to 1 standard deviation above the mean and a score above 80 (dark green) represents more than one standard deviation above the mean. Similarly, a score of 50 - 65 (yellow) represents 0 to 1 standard deviation below the mean, while a score of 35 - 50 (orange) equates to 1 to 2 standard deviations below the mean, and a score below 35 represents more than 2 standard deviations below the mean.
- Sim ID: 20857-1, Key: 0-0, Rpt: 68, Prd: 9680, Created: 2026-07-01 17:12 EDT
- UA: Mozilla/5.0 (Windows NT 6.3; Trident/7.0; Touch; rv:11.0) like Gecko

Score Calculation Detail

The following table provides a summary of how the overall score was calculated from each of the individual competency scores. First, all competency scores are calculated on a scale of 0-100. Note that some competencies use their color category rather than their actual numeric score in the overall calculation. For these, a standard score associated with the assigned color category is used in the overall score calculation rather than the actual numeric score. This is reflected in the ''Score Value Used'' column. Next, a weighted average of scores is computed using individual competency weights, typically set using job analysis data provided by the US Government Occupational Information Network (O*Net).

Competency	Score	How applied to overall	Score Value Used	Weight (%)
Branches and Merging	92.2149	Numeric Score	92.2149	16.6667
Git Configuration and File Management	81.8808	Numeric Score	81.8808	16.6667
Repositories and Commits	67.2363	Numeric Score	67.2363	16.6667
Undoing and Reverting Changes	88.6564	Numeric Score	88.6564	16.6667
Viewing History and Comparing Changes	68.4535	Numeric Score	68.4535	16.6667
Working with Remotes	97.9897	Numeric Score	97.9897	16.6667
Weighted Average:				82.7386
Final Overall Score:				82

Notes

(This area is intentionally blank - it's reserved as space for your notes.)