

## **Research Question**

What are benefits and pitfalls of using technical selection tests during the hiring process?  
How can technical selection tests be used outside of the hiring process to assess internal talent?

## **Introduction**

Talent Acquisition departments are the frontline soldiers in the war for talent. Selection tests and simulations are common mechanisms by which firm's filter through potential job candidates. Nevertheless, a 2017 Deloitte survey reported that 71% of firms believed they were weak in their ability to use these tools effectively<sup>1</sup>. Using assessments is a balancing act between false positives (hiring the wrong candidate) and false negatives (rejecting the right candidate). As such it is important to understand the benefits and pitfalls. This is especially true with technical assessments for coders and programmers.

## **Benefits**

- I. **LOWER COSTS AND IMPROVED EFFICIENCY.** A 2010 CEB survey reported that for each vacant position, approximately 65% of the applicants were unqualified.<sup>2</sup> A screening test filters out the unqualified candidates thus utilizing the recruiter time and resources more appropriately. Filtering out candidates via remote coding assessments can save the company a lot of money from onsite interview food and travel due to expenses. Furthermore, it assures the recruiter can prioritize his or her time on qualified candidates who otherwise would be lost in the backlog of applications.<sup>3</sup> Additionally, it also makes better use of your current developer's time.<sup>4</sup> Given that on roll developers and engineers perform the interviews, there is an innate opportunity cost for every interview. When a current developer is interviewing, they are not coding and thus not generating revenue. As such, it is critical to use their time efficiently by interviewing only the most qualified applicants.
- II. **OBJECTIVITY AND STANDARDIZATION.** A SHRM survey of 520 recruiters yielded that only 20% of them were very confident in their traditional screening methods.<sup>5</sup> This is of no surprise given the subjective nature of interviews and resume screens. Job knowledge tests, a category of which coding tests fall under, brings in objectivity.<sup>6</sup> Candidates cannot "game" the process the same way as with personality tests and interviews. They are also immune to rater biases and varying technical interviews.<sup>7</sup> Additionally, job knowledge assessments are likely to feature results that vary according to gender and race.<sup>8</sup> This can reduce the risk of disparate impact in your hiring practices.

- III. UNCONVENTIONAL SOURCES OF TALENT.** Technical coding assessments permit companies to find talent they would have otherwise been ignored via resume screening. Coding tests allow the company to objectively distinguish difference between credentials and ability.<sup>9</sup>

### Pitfalls

- I. **Ignores Soft Skills:** 25% of skills in traditional tech job listings are soft skills<sup>10</sup>. Strong technologists should be capable of communicating their ideas into terms that non-technical people can understand. Coding assessments do not measure the emotional intelligence of coders or their ability to collaborate. This shortfall can be easily overcome by supplementing coding assessments with interviews.
- II. **Time Consuming for Candidates:** Experienced candidates can be dissuaded by online assessments due to the potentially lengthy nature of the tests. They potentially could opt for other opportunities that do not require a lengthy application. Companies can counteract this by a) using assessments strictly for entry level coding and portfolios of work for experienced hires or b) using live coding tests on site that range from 15 minutes to an hour.<sup>11</sup>
- III. **Can omit debugging skills:** A 2008 study concluded that “most...[programming] students who were good debuggers were good novice programmers, although not all of the good programmers were successful at debugging”.<sup>12</sup> As such, a test that identifies a good programmer will not necessarily identify someone good at finding and fixing bugs. This can be mitigated by incorporating bug finding into preexisting assessments or generating new ones strictly for this skill

### Assessment for Incumbent Coders

There is not much research available concerning companies that use coding assessments to develop their programmers once they are hired. The closest initiative is Google’s detailed online learning for Nooglers (recent hire). Google uses a combination of “Checklists”, “Code labs” ( on the job tutorials), and “Codewalks” to develop engineers.<sup>13</sup> Google supplements this development with an open forum called *Noogler Engineers Helping Engineering Nooglers* or NEHEN.<sup>14</sup>

### Conclusion

Given the difference in productivity between a good coder and a bad coder, it is essential that the right candidate is hired<sup>15</sup>. Coding assessments in conjunction with traditional interviewing techniques can help companies get bright individuals who also are able to work in an innovative and dynamic team environment. Given the dynamic nature of technology, companies that use assessments should also be prepared to modify their assessments given changes in business and the subsequent skills required to thrive.

## References

1. Stephan, M., Brown, D., & Erickson, R. (2017, February 28). Talent acquisition: Enter the cognitive recruiter. Retrieved from <https://dupress.deloitte.com/dup-us-en/focus/human-capital-trends/2017/predictive-hiring-talent-acquisition.html>
2. Pulakos, E., & Kantrowitz, T. (n.d.). Choosing Effective Talent Assessments to Strengthen Your Organization. Retrieved from SHRM Foundation's Effective Practice Guidelines Series website: <https://www.shrm.org/foundation/news/documents/choosing%20effective%20talent%20assessments.pdf>
3. Hoffner, J. (2016, September 12). Hiring Programmers Is Hard: How to be More Efficient. Retrieved from The Qualified Coder website: <https://blog.qualified.io/hiring-programmers-is-hard-how-to-be-more-efficient-d5200c671f02>
4. Hoffner, J. (2016, September 12). Hiring Programmers Is Hard: How to be More Efficient. Retrieved from The Qualified Coder website: <https://blog.qualified.io/hiring-programmers-is-hard-how-to-be-more-efficient-d5200c671f02>
5. Maurer, R. (2017, August 24). Most Recruiters Not Fully Confident in Applicant Screening Methods. Retrieved from SHRM website: <https://www.shrm.org/resourcesandtools/hr-topics/talent-acquisition/pages/recruiters-not-confident-applicant-screening-methods.aspx>
6. Society for Industrial and Organizational Psychology. (n.d.). SIOP: Science for a Smarter Workplace. Retrieved from <http://www.siop.org/workplace/employment%20testing/testtypes.aspx>
7. Hoffner, J. (2016, September 12). Hiring Programmers Is Hard: How to be More Efficient. Retrieved from The Qualified Coder website: <https://blog.qualified.io/hiring-programmers-is-hard-how-to-be-more-efficient-d5200c671f02>
8. Society for Industrial and Organizational Psychology. (n.d.). SIOP: Science for a Smarter Workplace. Retrieved from <http://www.siop.org/workplace/employment%20testing/testtypes.aspx>
9. 3 Ways To Test Developers Before Hiring Them. (2016, November 29). Retrieved from Medium website: <https://medium.com/@trygigvy/3-ways-to-test-developers-before-hiring-them-d159b9a714b0>
10. 10 Soft Skills Every IT Professional Should Develop. (n.d.). Retrieved from Harvard Extension School website: <https://www.extension.harvard.edu/inside-extension/10-soft-skills-every-it-professional-should-develop>

11. 3 Ways To Test Developers Before Hiring Them. (2016, November 29). Retrieved from Medium website: <https://medium.com/@trygigvy/3-ways-to-test-developers-before-hiring-them-d159b9a714b0>
12. Fitzgerald, S., Lewandowski, G., McCauley, R., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: Finding, fixing and flailing, a multi-institutional study of novice debuggers. *Computer Science Education*, 18(2), 93-116. Retrieved from <https://search.proquest.com/docview/61963295?accountid=10267>
13. Johnson, M., & Senges, M. (2010). Learning to be a programmer in a complex organization. *Journal of Workplace Learning*, 22(3), 180-194.  
doi:<http://dx.doi.org.proxy.library.cornell.edu/10.1108/13665621011028620>
14. Johnson, M., & Senges, M. (2010). Learning to be a programmer in a complex organization. *Journal of Workplace Learning*, 22(3), 180-194.  
doi:<http://dx.doi.org.proxy.library.cornell.edu/10.1108/13665621011028620>
15. Hoffner, J. (2016, September 12). Hiring Programmers Is Hard: How to be More Efficient. Retrieved from The Qualified Coder website: <https://blog.qualified.io/hiring-programmers-is-hard-how-to-be-more-efficient-d5200c671f02>