Technical Interview Processes - Surely We Can Do Better

Ted Tencza

Introduction

Who am I?

Founder Code Purple Consulting

Over 25 years experience

Set up or revamped interview process multiple times

Trained interviewers at several companies

- Propsa
- Finder
- Bigcommerce
- Atlassian

Problems

Common Problems

- Gatekeeping
  - Degrees
  - Prior companies
  - Time at prior companies
- Bias
- “Thou shall not pass” syndrome
- Copy / Paste solutions
- Bad Coding Tests

Irrelevance / Unreliability

- Tests/tasks have no relation to work that will be done
  - CS Questions / tasks
- ESL skills
- Unclear expectations (on both sides)
- Variability (From Scratch, Code Review, Bug Fix, Whiteboard, Take Home)
- Interviewer lottery
  - Focusing on irrelevant criteria
  - Style bias (Just get it working vs. trying for optimal solution from start)
- Syntax Bingo - memory under pressure
- Tests with multiple solutions - but only accepting one
  - Take home code review - one line correction or long essay both possible interpretations

Other Issues

- Poor Interviewer skills
  - Assuming if someone can code, they can interview developers
  - Assuming they can properly evaluate performance
- Overwhelming candidate with interviewers
- Overwhelming interviewer with candidates

Candidate Impact

- Asymmetrical value/effort
- Skilled candidates will have other options

Some Solutions

Solutions proposed by others

- Scrap coding tests altogether
- GitHub project reviews
- Prepackaged coding tests - e.g., Hackerrank
- Take home projects
- Day contracts (or longer)

Issues with those Solutions

- Lack of data to make hiring decision
- Exclusionary - biased towards a specific cohort of developers
- Value Asymmetry
- Answers might be available online
- Assumptions not based on evidence
- Irrelevancy - especially pre-packaged tests

Keys to Good Solutions

Candidate Care at the forefront

- Should be driving factor in everything else that follows
- Clear expectations - and no trick questions
- Don’t assume free time (Take Home, Personal GitHub, side projects)
- Don’t assume prosperity (Personal super high spec laptop)
- Make it relevant
- Make it iterative
- Make tasks clear
- Focus on problem solving
- Help candidates that are stuck

Standardisation of experience

- Question library with exemplar answers
  - Shared understanding amongst interviewers
- Clear expectations on what is passing.
- Questions to avoid (blocking, leading, closed)
- Realistic conditions (IDE, Google/Stack Overflow, clarifying questions)

Candidate Briefing Package

- Reduces the unknown and therefore reduces stress
- Equipment / IDE / APIs etc.
- Any pre-work needed (***Minimise this)***

Dedicated Interviewing Program/Practice

- Training
- Recognise it is a skill
- Constant review
- Monitor feedback

Thank You