**FULL STACK DEVELOPER INTERVIEWER EVALUATION FORM**

|  |  |
| --- | --- |
| Applicant: Kate Penney  Interviewer: Tishana Trainor | Position: B3-C1 Full Stack Developer  Date: 10/21/2019 |
| **The following qualities help full stack developers to be successful:**   * Growth mindset characterized by curiosity, initiative, and persistence in learning new skills * Ability to take ownership of work, including asking for support when needed * Ability to communicate and collaborate on diverse teams across skill levels and subject knowledge * Web development knowledge and skills (as applicable to the C level, see expectations below) * Knowledge of programming skills and best practices (as applicable to the C level, see best practices below) * An interest in developing web applications and data visualizations for policy research and/or health care   **What are the candidate’s strongest and weakest areas? (If the candidate demonstrated other qualities that you think would help them succeed in this role, please include those as well.)**  Growth Mindset, definitely **.** | |
| **If you had to come up with your biggest fear/s in hiring this candidate, what would it/they be?**  As someone who was technically on a similar level as Kate when I started, I have no fears in working with her. Kate is easy to talk to and at our lunch she was able to express technical terms. She is also very aware of what she doesn’t know and seems eager to learn our stack. | |
| **What are your overall thoughts and recommended next step (Hire, Hold, Drop)? Any comments not covered in the above categories can be added here, as well.**  Hire | |

**B level web development knowledge and skills**

The B level staff represent the most junior staff at our company often coming straight from their undergraduate or masters degrees. They will be primarily responsible for writing code on software development projects ensuring that their code is highly maintainable and flexible. Junior level staff are expected to have an average billability between 90-100%.

Skills (By 6 months on the job):

* Being able to competently code the entire web stack
* Be competent in debugging their code
* Be able to unit test all their code
* Be able to communicate well to both technical and non-technical staff
* Know JIRA
* Know GIT
* Write frontend code that follows good accessibility standards (508)
* Know how modify an existing database

**C level web development knowledge and skills**

The C level staff represent employees that have typically worked in the field for 3-5 years. They will be expected to have a mastery level of all B level responsibilities/skills (above). In addition, they will be expected to do accomplish the skills below. C level staff are expected to have an average billability between 90-100%.

Skills:

* Task Lead on web projects - enforce best practices. The expectation would be leading on smaller tasks. Tasks with budgets <=200k and supervising 1-3 additional staff.
* Translate business requirements to technical solution at the feature or application level
* Be able to set up a web application from scratch
* Be able to set up CI environment for the application
* Design Database from scratch
* Fluent in D3
* Begin to gain fluency in a designated specialty
  + Data Visualization Design
  + UI/UX Design
  + QA Testing
  + Deployment/Architecture
  + Security
* Be able to design and implement e2e testing their code
* Design and oversee application specific testing and QA.
* Review other programmers code
* Mentoring application/code design of B Level Staff

**Programming best practices**

When approaching software development programmers should seek to follow the following principles (this is almost entirely stolen from the book *The Pragmatic Programmer* which should probably be read by all programmers):

1. **DRY – Don't Repeat Yourself**: Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.
2. **There Are No Final Decisions (or Requirements)** - No decision is cast in stone. Instead, consider each as being written in the sand at the beach, and **plan for change**.
3. **Always use source control** - enough said.
4. **Don't Live with Broken Windows** - Fix bad designs, wrong decisions, and poor code when you see them. (Note: term taken from the [*Broken windows theory*](https://en.wikipedia.org/wiki/Broken_windows_theory)).
5. **Test Test Test** - All code should be tested. More specifically you should design your code to be testable, ensure that your tests are automated and can be run with a single comment, and never consider code done until it is properly tested.
6. **Minimize Coupling/Maximize Orthogonality** - To the extent possible each piece of your code should not rely on the implementation of other parts of your code. This means when there are new requirements or bugs are found changes to the code can be minimal.

Interviewer Signature

# Please return completed form to HR.