Developer Trials

# Purpose

The goal of developer trials is to determine if a candidate's skill set is sufficient to be promoted to a full-fledged Software Developer. As a side effect, these trials also create a minimum standard that all Software Developers are expected to satisfy. It is not intended to guarantee promotion to Software Developer position, as other factors will be considered.

# Glossary

* Advisor – A senior developer assigned to a candidate to guide them through the trial.
* Advisory Panel – A group of advisors tasked with evaluating the final submissions at the end of the trial.
* Candidate – A developer that has elected to take the trial.

# Details

Trials will be held at the start of every quarter, during which a candidate will be assigned to write an application from scratch with the help of an advisor from the team. Candidates will be given two weeks to complete their assignments, after which the Advisory Panel will evaluate those assignments. Any Associate Developer may sign up to take a trial before it officially begins, though once a trial has begun, it cannot be started until the next quarter. Candidates may take the trial as many times as they’d like until they pass.

At the beginning of a trial, candidates will be given a problem statement. They will be charged with coming up with a software solution to the problem given and implement that solution within the time given. At the end of this period, an Advisory Panel will be convened which will evaluate the projects.

## Isolation

Candidates will ONLY be allowed to ask questions related to their assigned project to their advisor. The advisor will answer questions related to the requirements of the application but will avoid advising on how to solve the problem in a technical manner. This is to allow the trial to truly test the candidates’ skills in isolation, without accidentally testing the abilities of others on the team. Violations of this rule may result in either a warning or the candidate being ejected from the trial with a failing score, based on the severity of the violation.

## Skills Evaluated

During the trial, the following skills will be evaluated:

1. Database Design
   1. Have tables been properly named for their purpose?
   2. Are the relationships between the tables defined in a way that makes sense and discourages or prevents invalid states?
   3. Are the fields in the tables named in a way that expresses their purpose clearly?
   4. Do those fields have the correct constraints and types?
2. Software Architecture
   1. Are design patterns being used correctly?
   2. Has the functionality of the application been split up in a logical, clear, and consistent manner?
   3. Are methods/functions defined in a way that clearly expresses their intent?
   4. Do the parameters for methods/functions make sense?
3. Code Quality
   1. Do the classes defined have names that clearly express their purpose?
   2. Have all public classes, modules, methods, properties, fields, and functions been documented?
   3. Does the documentation provided clearly lay out the purpose and side effects of the documented item?
   4. Are there comments in the implementation itself that explain why certain decisions were made?
   5. Have difficult to understand sections of code been commented in a way that helps others understand that code?
   6. Are there any automated tests?
   7. Do the test names clearly indicate the case being tested for without requiring reading the test itself?
   8. Are the tests organized in a clear and consistent fashion?
   9. Are the tests useful?
   10. Do the tests produce consistent results, even if other parts of the application are modified?
   11. Have language features been used correctly?
4. Development Process
   1. Are the commit messages clear as to the changes made?
   2. How often are commits being made?
   3. Has the project itself been documented clearly, laying out the purpose, dependencies, and use of the application?
   4. Were additional requirements gathered from the Advisor?
5. User Interface
   1. Has the user interface been laid out in a clear and understandable fashion?
   2. Does the interface validate the user’s entered data?
   3. Is the interface pleasant to look at?
   4. Are errors communicated in a clear and understandable fashion to a lay-person?
6. Fitness
   1. Does the application solve the problem provided?
   2. Does the application avoid unnecessary complexity?
   3. Does that application work reliably?

## Scoring

At the end of the trial, candidates’ projects will be submitted to the Advisory Panel for evaluation. The panel will give a 1 to 5 score for each skill being tested. A candidate must achieve a minimum score of 2 in each category and a total score over 17. The scores mean the following:

1. Unacceptable / Fail: This aspect of the application is poor enough that significant work is needed to correct it.
2. Substandard: The work needs improvement.
3. Acceptable: This aspect matches what is expected of the average developer.
4. Good: This aspect of the application goes above the average.
5. Exemplary: This aspect is done well enough that it should be used as an example to others.

## Feedback

The Advisory Panel shall give written feedback for every skill that is being evaluated. This feedback should give the candidate suggestions on how to improve or explain why a skill was graded as it was. This feedback shall be given to the candidate along with the final scores.