# Writing Prompt Rubrics

The writing prompt rubrics below run on a 1 - 4 scale, with some categories covering less of the scale due to a more binary demonstrable criteria ('either they did or they didn't').

The current 'total points' that an apprentice can earn from the writing prompts is 46, and a minimum of 0. At this time, what would be considered a 'passing score' using these rubrics is to-be-determined. Some tuning will be required to hone in on what a fair passing score would be, when combined with the technical portion of the assessment.

Defining Project Requirements

Automated Teller Machine

| Criteria | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| **Customer Requirements** | Identifies important customer requirements, such as those necessary to perform the key function of the product. | Identifies a good number of meaningful customer requirements, but fails to list key requirement<br>OR<br>Is not detailed in describing key requirements. | Identifies few requirements, or only identifies vague requirements without providing justification. | Fails to identify customer requirements, or incorrectly includes user actions or technical requirements as customer requirements. |
| **Potential User Actions** | Identifies a substantial number of potential user actions (approx. 5+). | Most user actions are covered, but the listed actions could be broken down into more specific actions. | Identifies few user actions, only covering a small sections of possible actions<br>OR<br>provides only generic actions for any system. | Fails to identify any user actions, or incorrectly includes customer or technical requirements as user actions. |
| **Technical Requirements** | Identifies a substantial number of technical requirements (approx. 8+) for the software. Additional | A substantial number of technical requirements are described, but the technical requirements could be broken down into more detailed/specific needs. | Identifies few requirements, or only identifies vague requirements without providing justification. | Fails to identify customer requirements, or incorrectly includes user actions or customer requirements as technical requirements, |
| **Hardware Considerations** | | Identifies possible hardware needs, and explains why the need exists as related to the prompt. | Provides possible hardware considerations, but they do not describe why they are applicable to the prompt (if the need is not obvious). | Makes no comment on potential hardware considerations<br>OR<br>Does not justify why there are no/minimal hardware concerns. |
| **Security Considerations** | | Discusses the potential security concerns of the system, and provides justifications for why they are a concern and potential ways of limiting/resolving them. | Discusses potential concerns, but they are not described in the context of the prompt<br>OR<br>They provide no ideas/examples of how to limit/resolve the concerns. | Fails to recognize any security concerns,<br>OR<br>Listed concerns not relevant to the prompt.<br>OR<br>Does not justify why there are no security concerns. |

Max Score: 13

Tic-Tac-Toe

| Criteria | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| **Customer Requirements** | Identifies important customer requirements, such as those necessary to perform the key function of the product. | Identifies a good number of meaningful customer requirements, but fails to list key requirement OR Is not detailed in describing key requirements. | Identifies few requirements, or only identifies vague requirements without providing justification. | Fails to identify customer requirements, or incorrectly includes user actions or technical requirements as customer requirements. |
| **Potential User Actions** | Identifies a substantial number of potential user actions (approx. 5+). | Most user actions are covered, but the listed actions could be broken down into more specific actions. | Identifies few user actions, only covering a small sections of possible actions OR provides only generic actions for any system. | Fails to identify any user actions, or incorrectly includes customer or technical requirements as user actions. |
| **Technical Requirements** | Identifies a substantial number of technical requirements (approx. 8+) for the software. Additional | A substantial number of technical requirements are described, but the technical requirements could be broken down into more detailed/specific needs. | Identifies few requirements, or only identifies vague requirements without providing justification. | Fails to identify customer requirements, or incorrectly includes user actions or customer requirements as technical requirements, |
| **Hardware Considerations** | | Identifies possible hardware needs, and explains why the need exists as related to the prompt. | Provides possible hardware considerations, but they do not describe why they are applicable to the prompt (if the need is not obvious). | Makes no comment on potential hardware considerations OR Does not justify why there are no/minimal hardware concerns. |
| **Security Considerations** | | Discusses the potential security concerns of the system, and provides justifications for why they are a concern and potential ways of limiting/resolving them. | Discusses potential concerns, but they are not described in the context of the prompt OR They provide no ideas/examples of how to limit/resolve the concerns. | Fails to recognize any security concerns, OR Listed concerns not relevant to the prompt. OR Does not justify why there are no security concerns. |

Max Score: 13

Testing and Quality Assurance

| Criteria | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| **How Testing is Handled** | If a particular methodology is used, the apprentice describes the process in detail with correct use of industry terminology<br><br>If no particular methodology is used, the apprentice described how they test code in detail. | Described an approach to testing with mostly correct industry terminology, and includes how they would fit into the QA process as well as it's value. | Can describe an approach to code and software testing, but cannot describe why it is valuable or how they are a part of it. | Has no understanding of the value of software testing, or has no demonstrable experience with the process, that they can describe. |
| **Recognizing Platform Support Issues** | The apprentice both identifies the possible issues, as well as clearly describing why those issues occur in the system they choose | Identifies possible platform support, but is vague or provides incomplete descriptions for why it applies to their chosen development scenario. | Can describe platform support issues, but does not describe why or when they occur in any way. | Cannot identify common platform support issues for any software development scenario. |
| **Resolving/Managing Cross-Platform Issues** | | Answers includes possible ways of resolving common cross-platform issues, dependent on the scenario they chose. | Identifies methods for resolving/managing cross-platform issues, but does not explain how it solves the issue or why | Proposes no methods for resolving cross-platform issues, specific or otherwise. |

Max Score: 8

# Communicating Technical Ideas

## Technical Concept Explanation

| Criteria | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| **Use of Technical Terminology (and minimizing 'jargon')** | Minimizes the use of 'jargon', and defines any technical terminology they do use in simplified terms first. AND Communicates using analogies, examples, and other rhetorical devices. | Minimal use of jargon, and most jargon is described and explained when it comes upt OR Minimal examples, analogies etc. hinder the reader's ability to interpret the concept | Includes some jargon without explanation or description, or described technical terminology with other technical terms. | Does not define technical terminology they use. OR Defines terms in a fundamentally incorrect way. |
| **Correct Description of Technical Concept** | Describes the major principle or key features of the selected technical concept correctly AND Avoids unnecessary details that may confuse others. | Correctly describes the core concept, but includes some small details that may be incorrect without affecting the overall description. | Mostly correct description of the core concept chose, but key aspects of it are explained incorrectly OR Key aspects of the concept are missing entirely, leaving a 'hole' in the understanding of the reader | Describes the technical concept entirely incorrectly. |

Max Score: 6

## Software Development Lifecycle

| Criteria | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| **Definition of the Software Development Lifecycle** | Describes their workplace methodology in detail AND Describes how that methodology works for a developer in the day-to-day | Described a chosen methodology accurately | Provides a basic definition of a methodology, but does not go beyond the textbook definition to describe how the process works. | Cannot identify or define any methodologies used for managing the software development process. |
| **Strengths of Weaknesses of the Process** | If aware of or reflects on several strengths and weaknesses of software development methodologies, as they relate to the developers, the clients, and the product itself. Describes these pros/cons clearly. | Identifies strengths and weaknesses, but does not look at the overall process. For example, they may focus only on the issues a methodology causes for the developers. | Identifies a strength or weakness, but does not make clear why the methodology in question causes that benefit/concern. | Identifies no strengths/weaknesses. |

Max Score: 6