

Software Development Proficiency

Proficiencies (weight)	Below Expectations	Meets Expectations
Testing (10)	Insufficient automated tests.	Automated unit testing code is adequate for the given project.
Naming (10)	Most names used are short and/or unhelpful.	Modules/functions/variables are clearly named to reflect their functionality (self-documenting code).
Data Modelling (10)	Program data is stored in individual variables without using appropriate data structures to organize it.	Custom classes and common data structures such as arrays and dictionaries are appropriately used to organize the program data.
Functional Decomposition (10)	The project has few functions or functions lack a clear, singular purpose.	Functions have a clear, singular purpose. Long functions are avoided.
Code Duplication (10)	The project has frequent instances of code duplication	Code duplication is generally avoided.
Code Modularity (10)	The code is barely decomposed into modules/classes.	The code is for the most part decomposed into suitable modules/classes with well-defined responsibilities.
Version Control (10)	Few commits are made, often addressing multiple concerns (not atomic).	Frequent commits are made, each addressing just one project concern (atomic).
Issue Management (5)	Issues are not used effectively.	Issues are used effectively to organize the work.
Formatting (5)	Contains many long lines of code or blocks of commented-out code which should have been deleted.	Code has consistent and appropriate indenting and spacing that reflects the logical structure of the code.
Project Structure (5)	The code consists of a few very large files containing disparate elements.	The code is comprised of separate, well-focused files, none of them too large.
Resource Separation (5)	Literal values are frequently used directly in the program.	Literal values are typically stored in constant variables instead of being used directly.
Inline Comments (5)	Code contains unhelpful inline comments, or complex code lacks explanatory inline comments.	Explanatory inline comments are used when and only when needed to clarify complex code segments.
Code Documentation (5)	Modules and functions are in general not documented.	Most modules and functions are adequately documented.
