Toxikon Corporation

Bedford MA 01730

**STANDARD OPERATING PROCEDURE**

Reference Number: \*\*\*\*\*\*

Revision Number: 000

**Coding Standard**

1. **PURPOSE**

The purpose of this SOP is to describe the coding standards for internally developed code at Toxikon. These standards are flexible depending on the language and project being assessed, but they should be followed as much as possible.

1. **MATERIALS**

* PC or laptop.
* Text Editor, preferably one with syntax highlighting support for the used programming languages.
* Any other language or project dependent software for creation of code.
* StyleCop (optional).

1. **PROCEDURES**

General Guidelines

These principles apply to all coding projects developed in house, ranging from large web applications to small visual basic macros. These rules are staples of good code.

The Principle of Least Astonishment: When making decisions regarding how to implement features, choose the most obvious solution when possible, resulting in the least amount of astonishment or surprise. This applies to both UI/UX choices as well as back-end language and implementation choices.

Don’t Repeat Yourself (DRY code): Whenever possible, prevent repetition of reusable code. Repetitive code is difficult to manage and can lead to bugs when edits are made. This, however, should be balanced with the Rule of Three: If you need something once, build it. If you need something twice, pay attention. If you need it a third time, abstract it.

You Won’t Need It: Always solve the immediate overall problem at hand, while avoiding stipulating future problems. This will reduce the amount of time you spend writing unnecessary code. This also applies when developing front-end UI/UX features early in a project. You will find that you waste time trying to write this code if you do it too early in a project.

Repetition of Style

At Toxikon, applications tend to evolve, with users opting for a newer or more independent system to track or calculate data, reports, or requests. When producing an application that has a “predecessor,” try to replicate the flow of the previous application as much as possible to reduce the learning curve.

This design principle is also important when starting work on an existing body of code. The coding style consistency is most important within a project. It doesn’t matter if every project has the brackets each on a line of their own as long as the brackets within a single project as uniform. Global uniformity is desired, but not always possible. Due to variation in each language used to create these applications, developers are rarely able to have unique coding style throughout every application. Focus on keeping the style consistent and intuitive within a project and the code will be much more manageable.

User Interface Best Practices

When developing user interfaces for Toxikon applications, keep in mind the application’s predecessor. The flow of the app should be intuitive, and the easiest way to do that is to model if off the application that previously performed its role if such an application exists. Keeping the interface simple, with as many “borrowed” elements from other Toxikon apps, is the best approach for usability. Taking advantage of other trending UI/UX elements is also highly encouraged.

Documentation of Code and Changes

This step is incredibly important for use of your code by others. The more you write down explaining the design choices that you make, what changes you make to your code and when, the better. It is best practice to keep this documentation in your project folder, performing version control on it as well.

Statement of Purpose: When first starting a project, write down with as much detail what the end product should be. Depending on the project, this may be difficult, but determining what your end goal is

Development Notes: Throughout your project, writing down notes in a centralized location in a consistent format will allow for yourself and others to look back and see why certain choices were made. Revise this frequently to make sure that the notes reflect the current state of the application. It will save you time in the long run.

Change Records: After an application is initially deployed, it is frequently revised to reflect new desires and needs of the users. These changes need to be documented, and version control needs to be performed. Refer to *SOP 15.3.2.2 Change Management Procedures* for more information on how to document changes.

Commenting Code: This form of documentation, although least formal, can provide the greatest amount of utility for the effort. Simply commenting code, particularly blocks of algorithmically tricky code, can immensely improve quality of edits. Keep comments concise yet descriptive, and try to establish a convention for all comments.

1. **EVALUATION**

The evaluation of the adherence to the above coding standards will be defined by SOP 15.x.x.x Code Review Process.

1. **RECORDS**
   1. Data Recording Requirements

All activities must be recorded at the time of occurrence. Data collection forms and/or SOP related work forms and/or electronic data collection must be completed, and signed by technically competent and responsible personnel, as per Study Director SOP 1.5.21, current revision.

* 1. Archivable Data

All raw data, including ancillary data (e.g., equipment calibration forms, etc.) are to be archived according to SOPs 1.5.5, 1.5.5.1, and 1.5.5.4, current revisions, if applicable.

Records of archive activities are maintained in Lotus Notes or Matrix Gemini (SOP 15.3.9 and 15.3.17, current revisions), or other Software Systems, where applicable.

1. **SAFETY PRECAUTIONS AND PERSONAL PROTECTIVE EQUIPMENT (PPE) REQUIREMENTS**

The standards contained within Toxikon's Chemical Hygiene Plan apply, unless there are project or substance related specialized precautions, which would be provided in writing to the Study Director and staff for additional exposure assessment and training.

**Location:** Not applicable

Risk Level: Not applicable

Required PPE: Not applicable

Special Precautions: General safety precautions pertaining to data integrity and use of IT resources apply.

1. **REFERENCES**
   1. Technical Guidelines

Not applicable

* 1. Supplementary Standards

ISO/IEC 17025:2005, General Requirements for the Competence of Testing and Calibration Laboratories.

* 1. Published Literature

Microsoft C# Coding Conventions (C# Programming Guide)-

https://msdn.microsoft.com/en-us/library/ff926074.aspx

1. **FORMS / ATTACHMENTS**
   1. There are no specific forms in this SOP.
2. **RESPONSIBLE / ALTERNATE PERSONNEL**
   1. Responsible Parties: The parties responsible for the use and maintenance of this standard are as follows:

Manager, IT

* 1. Alternate Parties: The parties alternately responsible for the use and maintenance of this policy are as follows:

HR/Senior Management

1. **CROSS REFERENCED SOPs**

1.5.5 Storage and Retrieval of Records and Data

1.5.5.1 Quality Records

1.5.5.4 Storage, Retrieval, and Archive Management of Electronic Study Records

1.5.21 Study Director

SOP 15.3.2.2 Change Management

15.3.9 Workflow Databases

15.3.17 Matrix Gemini Laboratory Information Management System