**Change Management Policy**

**Paradigm Software Technologies, Inc. DBA Nexelus**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Purpose**

This policy establishes Paradigm Software Technologies, Inc. DBA Nexelus processes to manage changes across the organization in a well-communicated, planned and predictable manner that minimizes unplanned outages and unforeseen system issues.

**Roles and Responsibilities**

Nexelus Product Owner, and Development Manager are responsible for establishing, with the approval from the CTO, and monitoring this policy.

**Policy**

This policy communicates Paradigm Software Technologies, Inc. DBA Nexelus management’s intent to implement IT-supported business processes in a way that minimizes risk and impact to Paradigm Software Technologies, Inc. DBA Nexelus and its operations. Paradigm Software Technologies, Inc. DBA Nexelus will manage all system and application changes subject to this policy (e.g. operating system, computing hardware, networks, applications, data centers) in accordance with the applicable change management procedures.

**Procedures**

The objective of this process is to manage the introduction of any change into production by ensuring that the correct procedures are being followed, proper documentation has been completed, proper testing has been performed, and proper approval is in place.

**Organizational Change Control**

The following procedures apply to all changes, including infrastructure, code, and networking changes, as well as the deployment of new hardware:

* A record of agreed authorization levels will be maintained.
* Changes are only submitted by authorized users.
* Controls and integrity procedures will be reviewed to ensure that they will not be compromised by the changes.
* All software, information, database entities and hardware that require amendment will be identified.
* Security critical code to minimize the likelihood of known security weaknesses will be identified and checked.
* Formal approval must be obtained for detailed proposals before work begins.
* Authorized users must accept changes prior to implementation.
* Changes will be implemented at a time that is least intrusive to business processes involved.
* Vendor-supplied software will be used without modification; in the event that a modification is necessary, the following will be evaluated:
  + Risk of compromising built-in controls and integrity processes
  + Vendor consent
  + Getting the modifications from vendor as standard updates
  + Impact of owning the responsibility for maintaining the program
  + Compatibility with other software in use.
* A technical review of applications will be conducted after changes to operating platforms (operating systems, databases and middleware platforms). The review will include:
  + Application control and integrity procedures to ensure that they have not been compromised by the operating platform changes.
  + Timely notification of operating platform changes to allow appropriate tests and reviews to take place before implementation.
  + Appropriate changes are made to the business continuity plans.
* Customers must be notified of major changes (new subprocess or, etc.) or changes that may lead to system downtime, at least <TIME PERIOD> prior to their deployment to the production environment.

**Planned Changes**

For planned changes, Paradigm Software Technologies, Inc. DBA Nexelus will:

* Plan the implementation and assign tasks, responsibilities, deadlines and resources;
* Implement changes according to the plan; and,
* Monitor the implementation to confirm that they are implemented according to the plan.

**Unplanned Changes**

For observed unintended changes, Paradigm Software Technologies, Inc. DBA Nexelus will:

* Review consequences of the changes;
* Evaluate the occurrence or potential for occurrence of any adverse effects;
* Plan and implement actions to mitigate any adverse effects as necessary;
* In the case of Emergency Changes (e.g., when a critical vulnerability is discovered and needs to be resolved immediately), an expedited process may be conducted.

**Software Development**

For software development, operating system applications and software changes will only be implemented after extensive and successful testing:

* The tests will cover:
  + Usability
  + Security
  + Effects on other systems
  + User- friendliness
  + Tests will be conducted on separate systems (test environment), and all corresponding program source libraries will also be updated, as appropriate.
* The operational software, applications and program libraries of Paradigm Software Technologies, Inc. DBA Nexelus will only be updated by trained administrators upon appropriate management authorization.
* Company operational systems will only hold approved executable code, not development code or compilers.
* A configuration control system will be used to keep control of all implemented software as well as the system documentation.
  + Previous versions of software will be retained as a contingency measure.
  + Old versions of software will be archived, together with all required information and parameters, procedures, configuration details and supporting software for as long as the data are retained in the archive.
* The utility programs of Paradigm Software Technologies, Inc. DBA Nexelus will only be accessible to a minimum practical number of authorized users in conjunction with Paradigm Software Technologies, Inc. DBA Nexelus’s *System Access Control Policy*. The use of utility programs will require:
  + Use of identification, authentication and authorization procedures;
  + Defining and documenting of authorization levels and ad-hoc use for utility programs;
  + Not making utility programs available to users who have access to applications on systems where segregation of duties is required;
  + Removing or disabling all unnecessary utility programs;
  + At a minimum, logical segregation of utility programs from application software. Where practical, segregating network communications for such programs from application traffic;
  + Limitation of the availability of utility programs (e.g. for the duration of an authorized change);
  + Logging of all use of utility programs.
* There will be a rollback strategy in place before changes are implemented.
* An audit log will be maintained of all updates to operational program libraries.
* All decisions to upgrade to a new version release must take into account:
  + Business requirements for the change
  + Security of the release, e.g. the introduction of new information security functionality or the number and severity of information security problems affecting this version.
* Change security measures will be in place, to include:
  + Branch protection rules (GitHub);
  + Review & approval from the security team in case of significant changes;
  + Changes deployed to production only by specifically authorized personnel with escalated privileges; and,
  + Post deployment QA testing to ensure the change is functioning as intended in the production environment.

**Supplier Services**

Management of changes to supplier services is covered in the *Vendor Management Policy* in the Policy Center.

**Configuration Management**

For configuration management, production and system network changes will only be implemented with properly conducted procedures:

* Microsoft DevOps are used to standardize and automate configuration management.
* No systems are deployed into Paradigm Software Technologies, Inc. DBA Nexelus environments without approval of the Paradigm Software Technologies, Inc. DBA Nexelus CIO.
* All changes to production systems, network devices, and firewalls are approved by the Paradigm Software Technologies, Inc. DBA Nexelus CIO before they are implemented to assure they comply with business and security requirements.
* All changes to production systems are tested before they are implemented in production.
* Implementation of approved changes are only performed by authorized personnel.
* Tooling to generate an up-to-date inventory of systems, including corresponding architecture diagrams for related products and services, is hosted on <SYSTEM NAME>.
* All frontend functionality (developer dashboards and portals) is separated from backend (database and app servers) systems by being deployed on separate servers or containers.
* All software and systems are tested using unit tests and end to end tests.
* All committed code is reviewed using pull requests to assure software code quality and proactively detect potential security issues in development.
* Paradigm Software Technologies, Inc. DBA Nexelus utilizes development and staging environments that mirror production to assure proper function.
* Paradigm Software Technologies, Inc. DBA Nexelus also deploys environments locally using Vagrant to assure functionality before moving to staging or production.
* All formal change requests require unique ID and authentication.
* Identities assigned to multiple persons (e.g. shared identities) are only permitted where they are necessary for business or operational reasons and are subject to dedicated approval and documentation.
* Clocks are continuously synchronized to an authoritative source across all systems using a single reference time source. Modifying time data on systems is restricted.

These procedures will be reflected in Paradigm Software Technologies, Inc. DBA Nexelus’s Configuration Management Plan (see Appendix A).

**APPENDIX A**

**Configuration Management Plan [Template]**

**1. General Information**

|  |
| --- |
| **Purpose** |
| *Describe the purpose of the Configuration Management Plan* |
| *Describe the scope of the Configuration Management Plan as it relates to the project.* |

|  |
| --- |
| **Scope** |
| *Describe the scope of the Configuration Management Plan as it relates to the project.* |

|  |  |
| --- | --- |
| **System Overview** | |
| *Provide a brief system overview description as a point of reference for the remainder of the document.  In addition, include the following:* | |
| **Responsible Organization** |  |
| **System Name/Title** |  |
| **System Code** |  |
| **System Category** | * *Major application: performs clearly defined functions for which there is a readily identifiable security consideration and need* * *General support system: provides general ADP or network support for a variety of users and applications* |
| **Operational Status** | * *Operational* * *Under development* * *Undergoing a major modification* |
| **System Environment/ Special Conditions** |  |

|  |
| --- |
| **Project References** |
| *Provide a list of the references that were used in preparation of this document.  Examples of references are:*     * *Previously developed documents relating to the project* * *Documentation concerning related projects* * *Paradigm Software Technologies, Inc. DBA Nexelus standard procedures documents* |

|  |
| --- |
| **Acronyms & Abbreviations** |
| *If applicable, provide a list of the acronyms and abbreviations used in this document and the meaning of each.* |

|  |  |
| --- | --- |
| **Points of Contact** | |
| **Information** | *Provide a list of the points of organizational contact (POC) who may be needed by the document user for informational and troubleshooting purposes.  Include type of contact, contact name, department, telephone number, and e-mail address (if applicable).  Points of contact may include, but are not limited to, helpdesk POC, development/maintenance POC, and operations POC.* |
| **Coordination** | *Provide a list of organizations that require coordination between the project and its specific support function (e.g., installation coordination, security, etc.).  Include a schedule for coordination activities.* |

**2. Configuration Control**

|  |
| --- |
| **Change Control Board (CCB)** |
| *The Change Control Board (CCB) is a project-level, decision-making body that must approve or disapprove all change control requests before they can be implemented.  The CCB acts on those changes that would cause material or substantive changes to the system, including design specifications, budget (including lifecycle cost projections), the project schedule, and interface characteristics with other systems.*    *Describe the project CCB, its roles and responsibilities, and the membership.  The interaction between the CCB and management should also be presented in this section.  If the CCB is divided into separate organizations, such as a main CCB, a software management board, or a technical review board, indicate such in this section.  Identify the roles and responsibilities, participants, and interaction between each group, and management.*    *In addition, describe the CM organization as well as the relationship to other project entities and management management. Present the roles and responsibilities of each organization, and management area(s) within each organization, that will affect the CM function.* |

|  |  |
| --- | --- |
| **Configuration Items** | |
| *Configuration items (CI) are the products that are to be placed under configuration control.* | |
| **Management** | *Documentation describing the processes used to develop (or manage the development of) the system.* |
| **Technical** | *Documentation or baselines describing the system (e.g., Functional Requirements Document)* |
| **Software** | *Computer programs, operating systems and support tools, etc.* |
| **Data & Database** | *Files and records that exist apart from software, which access the contents of a database)* |
| **Network** | *As applicable* |
| **Hardware** | *Computer workstations, peripherals, servers and routers if applicable, etc.* |
| **Other** | *Other components that management may wish to include at its discretion.* |

|  |  |
| --- | --- |
| **Baseline Identification** | |
| *A baseline is a collection of information describing the technical characteristics of each CI.  Baselines serve as technical control points in the lifecycle for the evaluation of proposed changes to these technical characteristics.  The baseline and the approved changes or modifications provide a current description of the system.*    *Describe each system baseline, identified below, and the process by which it will be established and managed.  This should include, but is not limited to, the physical contents of the baseline, including the code being developed.  The physical contents may include hard copies of documentation and commercial off-the-shelf (COTS) software.  A graphic may also be created to depict where in the lifecycle process each baseline is generated and who becomes the responsible party of the identified baseline.* | |
| **Functional Baseline** | *The functional baseline, sometimes called the requirements baseline, is the main product of the Define System Phase and is managed in accordance with the Functional Requirements Document and Data Requirements Document.  Include a subsection for software and documentation, including design documentation, if applicable.*    *Describe where in the lifecycle the functional baseline will be established and the process by which it will be managed for this project.* |
| **Design Baseline** | *The design baseline reflects activities performed during the Design System Phase.  Its major component is a system/subsystem specification that defines the overall system design in terms of its subsystems, the allocation of requirements to subsystems and interfaces between subsystems and external systems.  The user acceptance evaluation criteria component of this baseline is defined in the Verification, Validation and Test (VV&T) Plan.  The user acceptance evaluation criteria are not a separate document but are a major element of the design baseline.  Include a subsection for software and documentation, including design documentation, if applicable.*    *Describe where in the lifecycle the design baseline will be established and the process by which it will be managed for this project.* |
| **Development Baseline** | *The development baseline, generated during the Build System Phase, defines the detailed structure of the system being implemented.  The development baseline’s major components are the generation of the computer programs (code) and the database.  Other components are the training documentation, user’s, operations, and maintenance documentation.  Include a subsection for software, documentation, etc., if applicable.*    *Describe where in the lifecycle the development baseline will be established and the process by which it will be managed for this project.* |
| **Product Baseline** | *The product baseline is established during the Evaluate System Phase.  The product baseline’s major component is the end system product as built by the developers.  This includes the following:*     * *Software* * *Design and specification documentation* * *Manuals (user, operations, maintenance, etc.)* * *Installation and conversion procedures*     *The product baseline is established after successful completion of the functional configuration audit (FCA), physical configuration audit (PCA) and associated system products and audit results presented at the Evaluate System review.  This baseline incorporates all changes needed to resolve problems detected during system acceptance and release testing and any discrepancies between the system, its requirements, and design documentation.*    *Describe where in the lifecycle the product baseline will be established and the process by which it will be managed for this project.* |

|  |  |
| --- | --- |
| **Roles & Responsibilities** | |
| **<Role 1>** | *Responsibilities* |
| **<Role 2>** | *Responsibilities* |
| **+** |  |

**3. Training**

|  |
| --- |
| **Training Approach** |
| *Provide information regarding the content and scheduling of CM training to be conducted for all personnel supporting the project.  Train project personnel, including those assigned responsibility for performing CM activities, in the objectives, procedures, and methods for performing their CM-related duties.  Examples of training include the following:*     * *Role, responsibility, and authority of the CM personnel;* * *CM standards, procedures, and methods;* * *CM tools and their capabilities; and* * *Data measurement, analysis, and reporting.* |

**4. Change Control Process**

|  |  |
| --- | --- |
| **Change Classification** | |
| *Describe how change classifications will be determined and assigned in terms of the level of severity of their impact.  Selection factors may include:*     * *Criticality* * *Interface requirements* * *Change sensitivity* * *Schedule* * *Ownership* * *Scope and complexity* | |
| **<Classification 1>** | *Description* |
| **+** |  |

|  |  |
| --- | --- |
| **Change Control Forms** | |
| *Document the flow that generated change control forms will follow from initiation through approval or disapproval.  Additionally, describe the forms that may be included in the change control process such as:*     * *Needs Statement* * *Requirements Change*     *Include sample forms in this plan.  These forms may include, but not be limited to, problem reports, system change requests, impact analysis reports, and change authorization notices.* | |
| **Form Flow** | *Describe* |
| **Form Types** | *List and Describe* |

|  |
| --- |
| **Problem Resolution Tracking** |
| *Describe the process used to log project problem requests and initiate resolution.* |

|  |
| --- |
| **Measurements** |
| *Define the measurements used to determine the status of CM activities, the effectiveness of CM processes, and the stability of controlled baseline deliverables.* |

|  |  |
| --- | --- |
| **Configuration Status Accounting** | |
| *All CM activities are recorded, stored, and reported by the CSA function.  The CSA function is a discipline that provides managers with feedback to determine whether decisions of the CCB are being implemented as directed.  As approved changes are executed, the CSA function records and files data concerning the appropriately modified software, hardware, and documentation.  The CSA function is responsible for identifying and issuing the most current approved versions of the CM-controlled items to project participants.*    *Identify the format and contents of the status summary reports that will be produced by the CSA function, and include them in an appendix to this plan.  Describe how the audit trail will be kept that identifies all changes implemented on approved baseline deliverables.  Examples may include using hard copy, diskettes (hard or compact), or a COTS tool.*    *Outline the processes and describe how captured information will be used to accomplish functions such as assuring that the software meets the design intent, contractual requirements are satisfied, and testing is performed in accordance with test plans.* | |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Configuration Management Libraries** | |
| *For each library (development, pilot, production, etc.), describe the organization of the CM library, including the multiple divisions of the library (the technical support library that stores the project development and production deliverables, the configuration library that contains records kept in support of the CCB, and the reference library consisting of technical documents that are either government-produced or COTS).  Each library type should be discussed in a separate subsection.* | |
| **Development** | *Description* |
| **Production** | *Description* |
| **+** |  |

|  |
| --- |
| **Release Management** |
| *Discuss the means by which the release of all project CIs will be managed.* |

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Editor** | **Approver** | **Description of Changes** | **Format** |
| 1.0 | September 13, 2022 | Compliance Analyst | CTO | Initial Creation | Electronic |