**Software Development Life Cycle (SDLC) Policy**

**Nexelus**

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

**Purpose**

This policy defines the high-level requirements for providing business program managers, business project managers, technical project managers, and other program and project stakeholders guidance to support the change management aligned with the Information Security Program. This includes change management for both infrastructure and software development.

**Roles and Responsibilities**

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

**Policy**

Nexelus' must establish and maintain processes for ensuring that its computer applications or systems follow an SDLC process which is consistent, repeatable and maintains information security at every stage.

**Process**

***Software Installation and Change on Operational Systems***

* Operating system applications and software will only be implemented after extensive and successful testing. The tests will cover:
  + Usability
  + Scalability
  + 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 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 archive.
* 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 consider:
  + 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 Control Procedures***

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.
* Track change developed by.
* Track change approved by QA.
* Track deployment approved by Manager.
* Track change deployed by.
* The change has been developed and deployed by different persons and segregation of duty is in place.
* Where applicable, UAT is approved by client.
* 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.
* Application changes will be implemented at a time that is least intrusive to business processes involved.
* BI/Reports, which do not require users to wait for updates and replacing application binaries, will be updated based on customer requirements.
* 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.

***Critical Change Control Procedures***

In case of a showstopper issue that hampers a business process to be completed on production environment, the Critical Change Control Procedure overrides the standard Change Control Process. The following procedures apply to all urgent/critical changes, including infrastructure, code, and networking changes:

* Product Owner, Systems/Network Manager or Development Manager will escalate the issue to be treated as Critical Change and authorize/direct relevant resources to work directly on the issue.
* CTO or CEO will be informed as soon as possible.
* Code changes will need to be approved by relevant authorized individuals other than developer(s).
* Change is reviewed directly by the Manager or QA.
* Change will be deployed on production by release manager with the approval of the CTO or the CEO. If both are not available, then the Product Owner or the Development Manager can authorize the deployment. In the absence of the Release Manager, the General Manager or the Development Manager may authorize another resource to access the server and deploy the change.
* If applicable, support will inform the set of clients impacted by this issue, about the fix.
* Log for critical changes will be maintained on Microsoft DevOps.