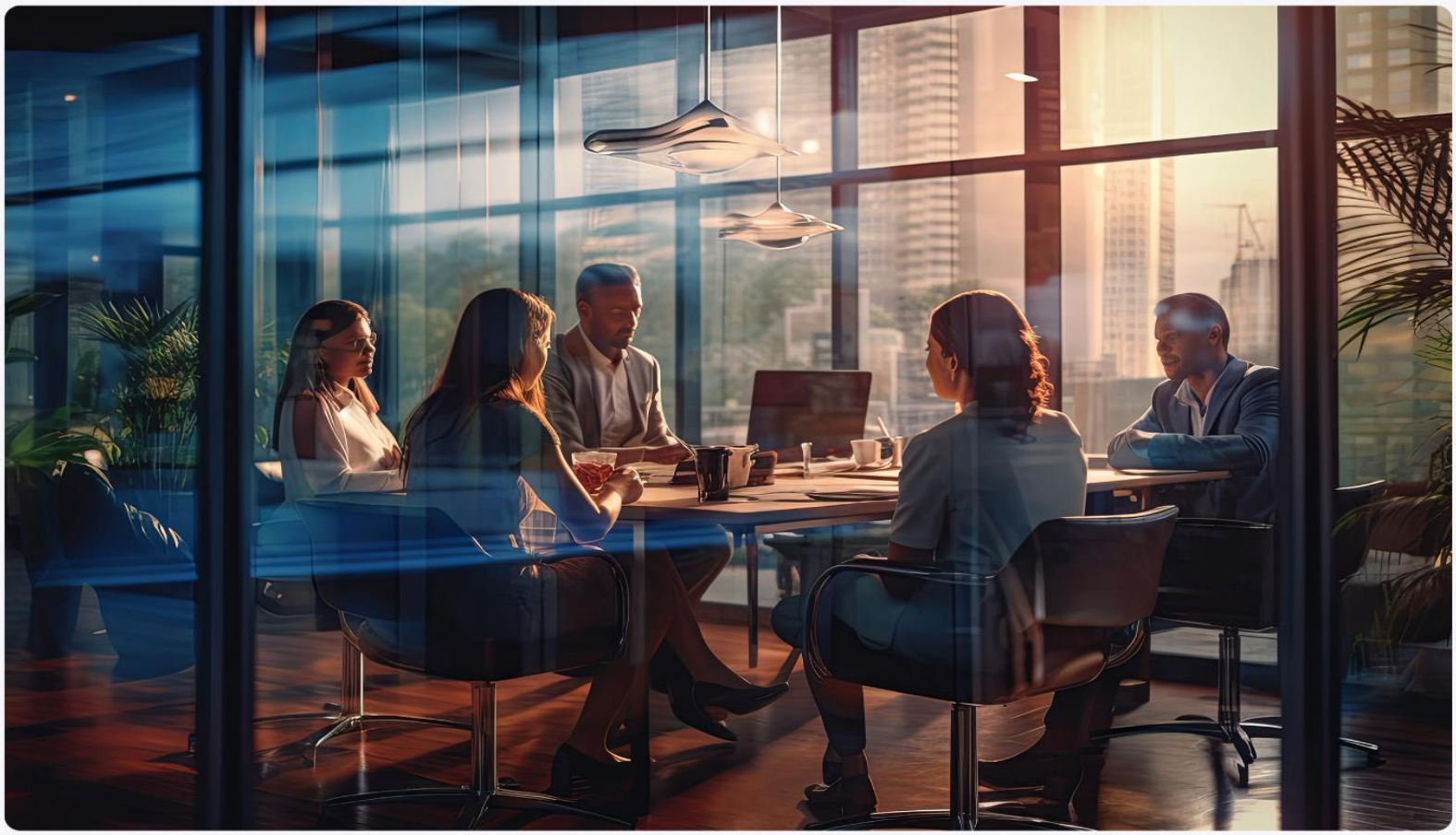

Policy for Secure SW Development and Maintenance

Standard Practice Document

Owner Name: Process Excellence Team(PEX)

Version: 2.0

neurealm

	Designation	Name
Prepared by	Sr .Manager – Process Excellence	Shalot Leely
Reviewed & Approved by	Director - PeX Team	Ambrish

Document History

Version No.	Release date	Process Improvement Proposal Reference No.	Summary of changes	Prepared by	Approved by
1.00	11-03-14		First release	Sekar T	Sundaram
1.01	01-May-2017	NA	Changed the GAVS Logo	Blessy Sara Bobby	Sekar T
1.02	06-Nov-2017	NA	Updated with Zero incident logo	Blessy Sara Bobby	Sekar. T
1.03	19-Mar-2019	NA	Annual Policy review – No changes to policy for SW development and maintenance – ADM	Mesiya A	Sekar T
1.04	30-Apr-2020	NA	Annual Policy review – No changes to policy for SW development and maintenance – ADM	Rama Vani Periasamy	Sekar T
1.05	28 May 2020	NA	Updated with the policy details of development & test environment	Rama Vani Periasamy	Sekar T
1.06	30 Apr 2021	NA	Annual Policy Review - Incorporated policy and practices related secure development & environment	Rama Vani Periasamy	Sekar T
1.07	18 Apr 2022	NA	Annual Policy Review - Incorporated policy and practices related secure development & environment	Vijesh C	Sekar T
1.08	17 Apr 2023	NA	Annual Policy Review - Incorporated policy and practices considering new controls i.e. secure coding and data masking.	Vijesh C	Sekar T

1.09	20 Feb 2024	NA	Policy review – Incorporating policy and practices considering PCI -DSS 4.0 requirements	Shalot Leely M	Sekar T
2.0	2nd May 2025	NA	Aligning to Neurealm template and standard	Shalot Leely M	Ambrish

Statement of Confidentiality

This Neurealm Private Limited (formerly known as GAVS Technologies Private Limited) artefact and/or document and/or presentation is strictly confidential and it contains proprietary information intended only for recipients of Neurealm Private Limited (Neurealm). The recipient acknowledges and agrees that: (i) this artefact and/or document is not intended to be distributed (ii) the recipient does not have the right to implement, copy, reproduce, fax, print, publicly divulge, or further distribute it, in whole or in part in any form, without seeking the express written permission from Neurealm. Any unauthorized use of the contents of this artefact and/or document and/or presentation in any manner whatsoever, is strictly prohibited. The artefact and/or document and/or presentation represents Neurealm's current product offerings and best practices which are subject to change without notice. Please note that Neurealm collaborates in relation to some of its offerings.

All third-party trademarks used herein belong to their respective owners and may be protected by law. This artefact and/or document and/or presentation only refers to such trademarks under the doctrines of nominative and descriptive fair usage to illustrate and explain concepts without implying violation of any legal constraints. If any improper activity is suspected, all available information may be used by Neurealm for lawful purposes and to seek appropriate remedies. Neurealm complies with applicable privacy laws and regulations. Recipients are advised to handle the information contained in this Material in accordance with relevant privacy and data protection laws.

1.0 Introduction

The purpose of this policy document is to define the organizational expectations for planning and performing the processes (process document is called as standard practice) in the various life cycles of software development and maintenance. Neurelam Delivery Excellence team set these expectations visible to those members of the organization who are affected in general, senior management is responsible for establishing and communicating guiding principles, direction, and expectations for the organization.

2.0 Software Development Policy

Neurealm Processes are compiled with the CMMI DEV model. The following are mandated by Neurealm management:

2.1 The approved documented process of Neurealm shall demonstrate that the software:

1. SDLC discovery starts with defining security & compliance objectives of the project and selection of right development methodology (Agile, Waterfall etc.,)
2. Is developed in support of documented requirements including and not limited to functional, security, technical, regulatory agreed to by the customer and conformance to these requirements is validated, i.e. generally through reviews/inspections and formally developed system/acceptance tests
3. Secure developed and managed using project management plan (a.k.a software development plan) in conformance with the allocated requirements
4. Is developed using approved development processes with structured reviews and approval before proceeding into the next phase, with defined entry and exit criteria.
5. All software/ Application development should be done with secure authentication and logging.
6. Application Development needs to incorporate considering of information security issues during each stage of the software development lifecycle.
7. Is developed and maintained under documented plans for configuration management and change control, throughout the life cycle including installation and customer configuration
8. Is designed using Thread Modelling (Thread modelling consists of identifying probable attack scenarios and adding relevant countermeasures to the application design) and Secure design approach
9. Consider vulnerabilities in the third-party software and its components and plan for applying patches when necessary
10. Is developed using formalized analysis and design methods, generally resulting in hierarchical decomposition of design and functional or object orientation
11. Is developed utilizing current development standards (including security & OWASP Top 10 vulnerabilities) and conventions for requirements, design, coding, documentation and testing as per the project management plan (software development plan)
12. Is scanned using static application scanning tools (SAST) review and ensure potential weakness and errors are resolved.
13. It goes through an automated code or manual code review to flag / fix potential issues and make the code stable.
14. Has documentation for each development phase and deliverable of the product.

15. Undergoes system/acceptance testing by individuals or organizations not directly involved in the design or implementation of the product being tested. Preferably all such independent test shall go through Dynamic application scanner tools (DAST) by simulating hacker attacks at runtime.
16. Undergoes a Penetration testing preferably using third-party team of security professionals to simulate possible attacks.
17. Demonstrate through metric analysis, achievement of annual business goals and Total Customer Satisfaction as the overriding objective.
18. Decisions are taken as per established decision process by identifying alternatives and evaluating alternatives.
19. All software development work shall exhibit a separation between production, development, and test environments. The development and staging data shall not have any production data in them.
20. Incident response plan describing the procedures the incident team must follow to address any security breaches that might occur.
21. Periodic security check shall be performed on all the environment mentioned in the project plan.
22. Data retention and data disposal policy shall be adopted at the end of life of the application / project.
23. Data masking shall be implemented wherever applicable, in accordance with both client and project requirements, to ensure the protection of sensitive information during the development, testing, and maintenance phases. This mitigates the risk of unauthorized exposure, preserves data privacy, and ensures compliance with relevant data privacy regulations as well as the specific needs of the client and project.

2.2 Each project must use an optimizing process as per Business Management System

2.3 Software being developed as per contract, as well as other third-party software, must be verified for conformance to requirements, and shall be developed using a process with demonstrated quality capability and repeatability that conforms to this policy. The subcontractor selection and management shall be according to a documented standard practice applicable for supplier management.

2.4 **Project Planning:** Every project will be allocated with a SPOC from QMG (PPQA aspect) and the roles as part of an overall charter to lead and teach in software quality practices, shall include the following responsibilities:

1. Jointly approve (with project management) a PPQA plan for every software project
2. Audit the development process for proper execution in conformance with appropriate sections of the processes and service / software delivery plan
3. Participate in the assessment and metric tracking of internal deliverables, as well as in the measurement, analysis, and feedback of overall software quality and customer satisfaction.
4. Perform, Test, witness, or audit, and attend selected life-cycle reviews, to assure appropriate software verification and validation.
5. Participate in configuration management (specifically change control) and continuous improvement process.

6. Provide appropriate visibility in situations of nonconformance.

7. Coordinate defect prevention activities organization wide.
8. Escalate the deviation of the development from the quality processes to the President Technology for approval for release.

2.5 Gathering Requirements & Analysis: Each business is to establish goals for software quality, project process performance and create metrics which support the tracking and assessment of the progress made towards those goals. The measurements shall be analysed to establish and maintain process capability baselines for the organization process and the results shall be used to improve the organization process. The measurements shall be used only to evaluate project's performance and not individual's performance.

2.6 Implementation: Each project is to be developed using established, effective project management techniques having allocated requirements as the basis and, including but not limited to:

1. size, effort, and schedule estimation and tracking
2. work breakdown structures and dependency analysis
3. Project tracking of all life-cycle deliverables and activities (documents, reviews, etc.)
4. an integrated project management plan (IPMP) compatible with the standard process
5. provision of necessary software support, process and PPQA resources in addition to development
6. identification and management of risks, and contingency planning
7. planning for training and defect prevention activities

Any major deviation from the standard software process require approval from SEPG/senior management.

2.7 Code Review:

1. Code reviews ensure code is developed according to secure coding guidelines.
2. Code reviews look for both existing and emerging software vulnerabilities.
    - Reviewed by individuals other than the originating code author, and who are knowledgeable about code-review techniques and secure coding practices.
    - Reviewed and approved by management prior to release.
3. 3. Process for maintaining inventory of Certificates for the web pages and Trusted Keys used for transmission of CHD.

4. Process for maintaining an inventory of Cipher Suites including at least the following:
    - Protocol/ Suite
    - Description
    - Strengths
    - Weaknesses
    - Usage/Application

5. Process for maintaining an inventory reflecting all the bespoke and custom software applications along with their version number, application owner and functionality/description.

6. If manual code reviews are performed for bespoke and custom software prior to release to production, code changes are:
    - Reviewed by individuals other than the originating code author, and who are

knowledgeable about code-review techniques and secure coding practices.

• Reviewed and approved by management prior to release.

7. Security vulnerabilities are identified and managed as follows:

• New security vulnerabilities are identified using industry-recognized sources for security vulnerability information, including alerts from international and national computer emergency response teams (CERTs).

• Vulnerabilities are assigned a risk ranking based on industry best practices and consideration of potential impact.

• Risk rankings identify, at a minimum, all vulnerabilities considered to be a high-risk or critical to the environment.

• Vulnerabilities for bespoke and custom, and third-party software (for example operating systems and databases) are covered.

8. Roles and functions are separated between production and pre-production environments to provide accountability such that only reviewed and approved changes are deployed.

9. Software engineering techniques or other methods are defined and in use by software development personnel to prevent or mitigate common software attacks and related vulnerabilities in bespoke and custom software, including but not limited to the following:

• Injection attacks, including SQL, LDAP, XPath, or other command, parameter, object, fault, or injection-type flaws.

• Attacks on data and data structures, including attempts to manipulate buffers, pointers, input data, or shared data.

• Attacks on cryptography usage, including attempts to exploit weak, insecure, or inappropriate cryptographic implementations, algorithms, cipher suites, or modes of operation.

• Attacks on business logic, including attempts to abuse or bypass application features and functionalities through the manipulation of APIs, communication protocols and channels, client-side functionality, or other system/application functions and resources. This includes cross-site scripting (XSS) and cross-site request forgery (CSRF).

• Attacks on access control mechanisms, including attempts to bypass or abuse identification, authentication, or authorization mechanisms, or attempts to exploit weaknesses in the implementation of such mechanisms.

• Attacks via any “high-risk” vulnerabilities identified in the vulnerability identification process, as defined in Requirement 6.3.1.

2.8 All software development shall demonstrate continual attention to process and quality improvement through cause/effect analysis.

2.9 Every group developing software shall have a capability improvement program in place including training, reuse of existing software (designed for this purpose), and technology acquisition, process improvement.

3.0 Process Improvement Policy

3.1 An organizational process improvement process shall evaluate the current processes and

technologies against emerging standards and identify triggers for significant process changes.

3.2 The roles of organization Process group (SEPG) as part of overall charter to initiate and track organization wide process improvement activities shall include the following responsibilities:

1. create, manage and own Software engineering and business process policies.
2. create, track and facilitate Process improvement roadmap proactively.
3. create and sustain Process awareness through seminars, workshops, training etc.,
4. Facilitate, track and manage Process Improvement Requests.
5. facilitate, review and track Technology roadmap and its implementation.
6. facilitate, track and review software engineering training plan structure.
7. Benchmark other organizations and external facilities to bring in applicable practices.

EPG shall have a representative from Senior Management and members from each functional group (Project Management, Engineering, Support, and Process) and a few delivery and project managers as permanent members and few project leaders on rotational basis.

3.3 The roles of software engineering technology group as part of overall charter to lead and provide support for new processes, tools and technologies shall include the following responsibilities software engineering need analysis, collation, and roadmap creation.

1. providing assistance and technology induction for business impact improvement, in particular for Productivity and Quality improvement
2. Process automation
3. create, manage, and own software process and assets.
4. create, implement, and manage software engineering training program.

4.0 Training Policy

4.1 Organizational training group shall

- Gather and analyze training needs.
- Develop tactical and strategic training plans.
- Organize training programs, analyze feedback, and assess training effectiveness.

4.2 Every software professional must undergo the following courses during the Induction Training Program

- Company HR Policies
- Organizational process trainings
- Security Awareness training

4.3 Every member of ADM must undergo the relevant training programs as per the identified training needs that are documented in the Master list of role-based training programs.

Getting started with secure development:

Ready to take your first steps toward secure software development? Here is the quick inputs:

- Review popular SDL methodologies and choose the one that suits you best. Do so at the beginning of your project. The cost of delay is high: the earlier you find potential security issues, the cheaper it is to fix them.
- "Mind the gap"—match your current security practices against the list of SDL activities and identify the gaps.
- Read case studies / lessons learned on SDL implementation in projects similar to yours. Consider their successful moves and learn from their mistakes.
- Come up with a list of practices to cover the gaps. Prioritize them and add activities that improve security to your project's roadmap.
- Get buy-in from management, gauge your resources, and check whether you are going to need to outsource.
- Train your team on application security and relevant regulations to improve awareness of possible threats.
- "Shift left" by implementing each security check as early as possible in the development lifecycle. This will save you a lot of resources, as the price of fixing security issues grows drastically with time.
- Automate everything you can. Take advantage of static code scanners from the very beginning of coding. Add dynamic scanning and testing tools as soon as you have a stable build.
- Do not hesitate to hire outside experts. Arrange for security audits, since an outside point of view might identify a threat you failed to notice.