GATHERACT LLC

Software Development Lifecycle Policy

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 approval, planning, and life cycle development of GATHERACT LLC, software systems.

Policy

GATHERACT LLC, must establish and maintain processes for ensuring that its computer applications or systems follow an SDLC process which is consistent and repeatable.

Program and Project Management Standard

This SDLC process will include a Program and Project Management methodology that incorporates best practices and provides a standard structure for planning, managing and overseeing IT programs and projects over their entire life cycle. Best practices for Program and Project Management at minimum include the following documented components:

- at minimum one Project Sponsor
- defined project objectives and scope
- defined budget and timeline
- IT compliance & risk assessment (for projects subject to regulations) to include security requirements and related controls and which may include an outside security auditor to certify those controls.
- project approval by the Project Sponsor (and the Board of Directors if applicable)
- a Project Manager (depending on the project size and complexity may be part-time)
- a detailed project plan with project deliverables for projects with a development budget above 10 percent of the software development budget for the year, or for projects subject to regulations
- a process for adjusting objectives, scope, budget or timeline including approval by the Project Sponsor (and the Chief Risk Officer if applicable)
- acceptance of project deliverables by the Project Sponsor (and Board of Directors if applicable) prior to project closure. (This includes explicit sign-off of all IT compliance deliverables for projects subject to regulations.)

Software Development Phases and Approach Standard

A Software Development Project consists of a defined set of phases:

1. **Determine System Need Phase**
   The Determine System Need phase is the period of time in which an information system need is identified and the decision is made whether to commit the necessary resources to address the need.

2. **Define System Requirements Phase**
   The Define System Requirements phase is the period in which the User Requirements are broken down into more detailed requirements which can be used during designing and coding.

3. **Design System Component Standard**
   The Design System Components phase transforms requirements into specifications to guide the work of the Development phase. The decisions made in this phase address how the system will meet the functional, physical, interface, and data requirements. Design phase activities may be conducted in an iterative fashion, producing a system design that emphasizes the functional features of the system and technical detail.

4. **Build System Component Phase**
   The Build phase transforms the detailed, system design into complete coded software units and eventually, into an integrated product for release. Each software unit and subsequent integrated units are tested thoroughly. System documents that support installation and operations are also developed in this phase.

5. **Evaluate System Readiness Phase**
   This Evaluate phase is to ensure that the system as designed and built satisfies the requirements of the user. Whenever possible, independent testers measure the system's ability to perform the functions that are required by the customer and ensure an acceptable level of quality and performance. Once the phase is complete, it will be evident whether or not the system is ready for operation or redevelopment.

6. **System Deployment Phase**
   Deploy the System phase is the final phase of the development life cycle, when the system is released initially to a pilot site and then into the production environment. All necessary training for using the system is accomplished. The sequence of the phases depends on the software development approach taken. These approaches include but are not limited to:

   a. Waterfall Development
   b. Iterative Development
   c. Staged Delivery Development

   Based on the approach for and the size of the software development some of the phases can be combined. In Iterative Development there may be multiple Cycles (iterations) of the above phases before the final software is released.

7. **Program and Project Management Guidelines**
   In addition to the minimum Program and Project Management Standards, GATHERACT LLC should conform to the project management best practices as established in the *PMBOK Guide* issued by the Project Management Institute, the PRINCE2 methodology or similar.

SDLC Control Guidelines

The SDLC process will adhere to the following controls:

- Adequate procedures should be established to provide separation of duties in the origination and approval of source documents. This shall include but not be limited to separation of duties between Personnel assigned to the development/test environment and those assigned to the production environment.
- Modification of code or an emergency release will follow the change control standard.

- Secure programming standards should be followed. Secure code training should be provided to <COMPANY NAME>, Inc's developers.
- All software deployed on Corporate or Hosted infrastructure must prevent security issues including but not limited to those covered by SAN and OWASP.
- Code changes are reviewed by individuals other than the originating code author and by individuals who are knowledgeable in code review techniques and secure coding practices.
- Overrides of edit checks, approvals, and changes to confirmed transactions should be appropriately authorized, documented, and reviewed.
- Application development activity should be separated from the production and test environments. The extent of separation, logical or physical, is recommended to be appropriate to the risk of the business application or be in line with customer contractual requirements. The level of separation that is necessary between production, development, and test environments should be evaluated and controls established to secure that separation.
- All changes to production environments should strictly follow change control procedures, including human approval of all changes, granted by an authorized owner of that environment. Automated updates should be disallowed without such approval.
- Active production environments should not be re-used as test environments. Inactive and/or decommissioned production environments should not be used as test environments unless all private data has been removed. Test environments should not be re-used as production environments without going through a decommissioning and recommissioning process that cleans all remnants of test data, tools, etc.
- Individuals who are responsible for supporting or writing code for an internet-facing application, or internal application that utilizes web technology and handles customer information, should complete **annual** security training specific to secure coding practices. For individuals supporting or writing code for an internet-facing application, training should also include topics specific to internet threats. The individual should complete the training prior to writing or supporting the code. The training must include OWASP secure development principles as well as OWASP top 10 vulnerability awareness for the most recent year available.
- Separate duties between Personnel assigned to the development/test environment and those assigned to production environment.
- Custom accounts and user IDs and/or passwords should be removed from applications before applications become active or are released to customers.
- Production data should not be used in testing or development environment.
- Security controls that are in place for the production copy in the test system should be production quality (e.g. mirroring the production controls over the data).
- When conducting quality assurance (QA) testing prior to the release of a new feature requiring user input where constraints on user input may be reasonably understood, feature acceptance tests must include testing of edge and boundary cases.
- All major release candidates must be scanned for vulnerabilities using a black box vulnerability scanner prior to release in production. Any high or critical vulnerabilities must be remediated or marked as mitigated prior to the approval of the release candidate for release to production.

For situations demonstrating that testing needs to use production data, the requirements are the following:

- The Information Resource Owner will provide approval before production data can be used for testing purposes.

- Wherever possible, the production data should be tokenized or anonymized instead of using production data.
- Testing and parallel runs should use a separate copy of production data and the test location or destination should be acceptable (e.g. loading confidential production data to a laptop for testing is not acceptable).
- The data should not be extracted, handled, or used by the test process in a manner that subjects the data to unauthorized disclosure.
- The data should be accessed on a need-to-know basis.
- Normal test activities should not use production data. In cases where test activity requires access to production data, access to production data should be restricted to only those individuals who have a documented business need. Only the information with the documented business need should be accessible by those users.
- Production data used for testing should be securely erased upon completion of testing.
- Test data and accounts will be removed before being placed into production.
- Restricted/Protected Information will be encrypted according to the Encryption Standard while at rest or in transit.
- Error message must be handled securely and they must not leak sensitive information.
- Periodic static code analysis by an automated tool will be performed against internally developed code prior the testing using production data.
- Dynamic application vulnerability scanning should be performed before the product is deployed with production data and/or in a production environment. When source code is available, static code analysis should be performed before the product is deployed with production data and/or in a production environment. Issues found should be remediated unless an exception is approved by the IT Security.

Secure Coding Policy

This policy defines the high-level requirements for verifying that GATHERACT LLC applications and products interact with information only in a secure manner. The actual information and the integrity and availability of the data must be protected at all times.

Multi-Threading Standard

Multi-threaded applications must be written so that they are thread-safe.

Buffer Overruns Standard

- Applications partially or fully written in languages that allow developers to directly manage memory and/or to directly access memory locations must be written to avoid buffer overruns.
- Custom array accessing, custom string management, and other custom memory management implementations written by GATHERACT LLC must not be used without a business justification.
- Automatic memory management mechanisms must be used wherever possible.
- Automatic string management mechanisms must be used wherever possible.
- Custom routines should be modularized and abstracted so that developers using the custom management routine do not need to know anything about the custom management in order to use it correctly and safely.
- Array access mechanisms must be written so that they return an error message instead of allowing numeric overflow or underflow.

- Numeric overflow and underflow must be checked with one of the following two mechanisms.
    - Check that available "space" in the numeric field is greater than the amount of space that would be added to the variable.
    - Institute a hard coded limit to the maximum value at which a numeric field may be added, and a commensurate hard coded numeric value that is allowed to be added to the field.
        - Variables should be manually set to NULL after freeing them.
        - Variables in re-entrant functions should be checked if they are null before using them.
        - Automatic memory management mechanisms should be used wherever possible.
        - Automatic string management mechanisms should be used wherever possible.

Cryptography Standard

- Custom cryptography is not allowed. This applies to algorithms, modes, secure protocols, secure random number generation, and all other uses of cryptography.
- All cryptographic algorithms, modes, secure protocols, secure random number generation and all other cryptographic tools used by GATHERACT LLC products must be chosen from the list of algorithms approved by Information Security Department.
- Products must maximize configurability of cryptographic algorithms, modes, protocols, libraries, etc., so that when new discoveries force GATHERACT LLC to change, GATHERACT LLC is able to do so with minimal product impact.
- Encrypted data must include an identifier that indicates the algorithm and mode used to encrypt the data, so that it is easy to seamlessly migrate from an algorithm and/or mode to a new one.

Mobile Local Storage Standard

Storing Restricted/Protected Information on a mobile device requires Compliance officer approval.

Mobile Data Caching Standard

Restricted/Protected Information must not be permanently cached on the mobile applications device. This includes, but is not limited to: caches, cookies, preferences, webkits, information that could later be retrieved by the mobile application's device backup software, etc.

Bulk Download Standard

- Bulk download of Restricted/Protected Information to mobile devices is not allowed.
- Restricted/Protected Information sent to the mobile application's device must be limited to the minimum confidential and/or proprietary data necessary for the function of the application or for the function of the application's interface.

Mobile Application Geolocation Standard

The Information Security department must assess and approve mobile applications utilizing the geolocation (GPS) functionality prior to application deployment.

Mobile Code Obfuscation Standard

Interpreted client side code (e.g. database variables, database table names, etc.) must be obfuscated. Passwords, usernames, and database connection strings must not be present in any code.

Web Security Guidelines

The following application countermeasures should be implemented for all web applications.

- Clickjacking - Web applications should provide defense against malicious attempts at encapsulating web pages into a frameset.
- Cookies –
    - Cookies should have the domain restricted to the most discreet host and path feasible.
    - Session cookies should not allow a cookie to be replayed to prevent JavaScript from accessing the cookie.
    - Cookies used for HTTPS sessions should have the "secure" attribute set. Session cookies should have the "HttpOnly" attribute set to ensure that the browser only sends a cookie over a secure connection where supported by the underlying technology stack.
    - Cookies should not have an "expires" attribute set.
    - Except for the session identifier, cookies should not contain clear text sensitive information.
- Session Cookies –
    - Web applications should generate a new session token/identifier when switching from unauthenticated to authenticated sessions.
    - Session cookies should be transient, not persistent cookies.
    - For client certificate protected sites, the secure session token should be correlated to the client certificate for critical business functions (e.g. changes to the system of record).
- Comments in returned content (e.g. HTML, JSP, client side JARs, Active-X controls, Flash shared objects etc.) should be minimized.
- Error messages should not return unsanitized data, such as raw URL requests or responses from external sites. Such error pages are common vectors for Cross-Site- Scripting attacks.
- Applications should disable browser and proxy caching if processing Restricted/Protected Information.
- Applications should disable the browser "Auto completion" features on forms containing Restricted/Protected Information.
- Restricted/Protected Information should not be sent via URL parameters. The information should be sent using POST parameters, including HTML FORM inputs and URL redirection.
- The HTTP TRACE method should be disabled in production.
- Cross-Site Request Forgery (CSRF/XSRF)
- Applications should use transient authentication methods or include a secret.
- Applications should not use persistent authentication methods alone (e.g. a cookie or HTTP authentication) as a defense to Cross-Site Request Forgery (CSRF/XSRF).
- Applications should not rely on hidden fields, URL parameters, cookie values, HTTP headers (such as HTTP_REFERER), or other obscurity techniques as the basis for authorization decisions.

Mobile Local Storage Guidelines

- Mobile applications' local storage of credentials and other Restricted/Protected Information should be encrypted.

- Access to local data containing Restricted/Protected Information should be authenticated.
- Mobile applications should not store usernames, passwords, encryption keys, challenge question answers, activation codes, database connection strings or any other confidential information in plaintext.

Bluetooth Guidelines

- The mobile application should not modify Bluetooth settings.
- Bluetooth pairing should be done in the device OS settings and not through the mobile application. The mobile application should not place the device into Bluetooth discovery mode.

Mobile Code Signing Guidelines

All mobile code should be approved by the Information Security Department before it can be released into production or uploaded to any marketplace application store. The following controls should be in place for controlling the cryptographic keys or access to the release/distribution process:

- Any mobile applications released should be signed by a certificate represented by GATHERACT LLC or a third-party certificate with keys or credentials officially assigned to channel partner or GATHERACT LLC. By using a GATHERACT LLC-owned signing key, customers will know that mobile (iOS, Android, etc.) applications really came from GATHERACT LLC and can be trusted.
- Keys or signing credentials should not be controlled by third party vendors or have access to the app store release/distribution process.
- The signing party should ensure that the code be signed is the valid source code.
- The signing party should ensure that the key is properly secured.
- The signing party should verify that the appropriate security reviews have been completed.

Applicable Standards

Applicable Standards from the HITRUST Common Security Framework

- 09.b – Change Management
- 09.c – Segregation of Duties

Applicable Standards from the HIPAA Security Rule

- 164.308(a)(5)(i) - Security Awareness and Training
- 164.310 (c) – Workstation Security
- 164.310 (d)(1) – Device and Media Controls

Version History

Number	Published	Author	Description
1	4/4/2022	Joe Crop	Initial creation