Number: **CRI.POL-004**	**Title: Clinical Research Informatics Software Development Life Cycle (SDLC) Policy**	
	Effective Date: June 15, 2022	**Page 1 of**

Supersedes Version dated: N/A	REQUIRED APPROVALS BELOW	
PDG-CISIL Approval Date: May 17, 2022		
CRI Technical Director Approval:	DocuSigned by: Bill Sanus FB86108144CF4B1...	Date: 7/26/2022
CRI Scientific Director Approval:	DocuSigned by: Meredith Zozus CE16226E6FE3443...	Date: 8/9/2022

1.0 Purpose

The mission of Clinical Research Informatics (CRI) at UTHSA is to support knowledge generation toward improvement in human health through provision of data, information systems, and related services to clinical investigators and research teams. Accomplishing this mission requires ensuring the following manifesto:

(1) Data are handled in a manner compliant with applicable laws, regulations, and other requirements including research contracts and that privacy and confidentiality are maintained,

(2) Data are available when needed for use in human subject protection and institutional oversight,

(3) Data are capable of supporting study conclusions and other intended decision-making,

(4) Data are documented sufficiently to support reuse, research reproducibility, and replication, and

(5) Software and other computer programs developed or offered by CRI function as intended.

The CRI System Development Life Cycle (SDLC) Policy establishes the minimum requirements and responsibilities for software and system development. It provides division leaders, project managers, technical staff, and project stakeholders guidance and the implementation standards for software and system development. Programming is a complex endeavor and susceptible to failure unless undertaken with a deliberate and systematic methodology. It is crucial that system developers, customers, and all levels of CRI technical management adhere to the CRI SDLC. To be successful, all application and programming projects must follow a well-defined development lifecycle. A software application typically undergoes several development lifecycles, corresponding to its creation and subsequent upgrades. Each such development lifecycle constitutes a project. Such projects continue until the underlying technology

ages to the point where it is no longer economical to invest in upgrades and the application is considered for either continued as-is operation or retirement.

The CRI SDLC is a methodology for implementing an application project by following a sequence of standard steps and techniques and acts as a collection of resources designed to support the approval, planning, and programming life cycle development of CRI developed software and systems. This, in combination with sound Project Management, improves the ability of CRI application projects to deliver as expected, on time, and within budget.

The goals of this SDLC are to:

- Establish the SDLC process that will be standardize, document, and control CRI software and system development and management
- Deliver quality systems which meet or exceed customer expectations when promised and within cost estimates.
- To provide a framework for developing quality systems using an identifiable, measurable, and repeatable process.
- Identify and assign the roles and responsibilities of all involved parties, including functional and technical managers, throughout the system development life cycle.
- Ensure that system and project risks are detected early, requirements are well defined, and subsequently satisfied.

The SDLC policy will achieve these goals by:

- Establishing appropriate levels of management authority to provide timely direction, coordination, control, review, and approval of the system development project.
- Ensuring project development staff accountability.
- Documenting requirements and maintaining trace ability of those requirements throughout the development and implementation process.
- Ensuring that projects are developed within the current and planned information technology infrastructure.
- Identifying project risks early

2.0 Scope

This policy applies to all programming-related endeavors for which data are managed by Clinical Research Informatics (CRI) at the University of Texas Health Science Center San Antonio (UTHSA). The SDLC spans from origination through delivery or system decommissioning.

3.0 Responsibility

3.1 The CRI Directors are responsible for maintaining and implementing this policy and ensuring the CRI SDLC SOPs are sufficient to provide sufficient control over software and system development and management processes.

3.2 CRI's faculty, staff and subcontractors performing programming tasks are responsible for adhering to CRI policies, SOPs, and Work Instructions (WIs) and will notify a CRI Director when adherence will not be likely or has lapsed.

3.3 CRI's Faculty, staff and subcontractors performing programming tasks are responsible for adhering to study- and program-specific procedures documented in SOPs and will notify CRI Technical management when adherence will not be likely or has lapsed.

4.0 References

4.1 UT System: UTS 165 Information Resources Use and Security Policy

4.2 UT Health San Antonio: HOP 5.8 Information Security

4.3 UT Health San Antonio: HOP 5.11 Project Delivery

4.4 UT Health San Antonio: HOP 5.8.25: Systems Development Life Cycle Policy

4.5 Title 21 CFR Part 11, Electronic Records, Electronic Signatures

4.6 NIST 800-53/NIST 800-53A, Security and Privacy Controls for Information Systems and Organizations

4.7 Software Engineering Institute, Capability Maturity Model and Agile Frameworks

4.8 GAMP 5 Guide: Compliant GxP Computerized Systems

4.9 Clinical Research Informatics Policies

4.9.1 CRI.POL-001: Quality Management System

4.9.2 CRI.POL-002: Human Resources Management Lifecycle

4.9.3 CRI.POL-003: Data Management Lifecycle

4.9.4 CRI.POL-005: Project Management Lifecycle

5.0 Definitions

Term	Definition
Auditability	Whether adherence to a process can be independently and objectively verified.
CRI	Clinical Research Informatics
CRIS	Clinical Research Informatics Specialist
Project Sponsor	A person or group who provides resources and support for the project, program or portfolio and is accountable for enabling success.
QMS	Quality Management System: a formal system that documents the structure, processes, roles, responsibilities and procedures required to achieve effective quality management (ASQ Glossary)
Qualified Designee	An individual designated by an individual in the responsible role and directly overseen by the individual in the responsible role or equally qualified by training, education, or experience according to the Training SOP to perform a responsibility of the designating role.
Software	Any computer code written to perform a task or related set of tasks

Term	Definition
Software Application	A software application is any program, or group of programs, that is designed for the end user to perform a task or related tasks.
SOP	Standard Operating Procedure: text that communicates an organization's requirements for a process including what will be done (process tasks), when (their sequence and timing), and by whom (the institutional roles responsible for performing and overseeing the tasks). In regulated industries, SOPs also commonly specify the documentation generated by a process (process documentation), and how the quality of a process will be controlled.
Substantive changes	Changes to tasks, their sequence, timing, documentation, or the roles designated as responsible for tasks.
Unexpected quality problem	Any deviation from the fitness for use of data or computer programming that (1) were not anticipated, i.e., for which detection and control mechanisms were not planned or do not exist, or that (2) occur with a greater severity or frequency than anticipated in such plans.
UTHSA	University of Texas Health Science Center San Antonio
Work Instructions	Work instructions refer to documented guidelines that clarify how to perform assignments. They provide precise descriptions of task-related steps to reduce potential setbacks, damages, and inconveniences. Work instructions are also known as work guides, job aids, or user manuals.

6.0 Policy

Projects enter the CRI SDLC process via the procedures set forth in CRI.POL-005 Project Life Cycle Management. Once the project enters the SDLC process, it goes through eight phases, during which defined products and documents are created, reviewed, refined, and approved. Not every project will require that all phases be subsequently executed and may be tailored to accommodate the unique aspects of projects. All project documentation will be managed within the CRI Project Tracking System following the guidelines within CRI.POL-005 Project Life Cycle Management.

Phase One: Definition

Technical projects are defined based on the required deliverables which in turn defines the primary programming teams and development methodologies necessary for completion. Technical projects are defined as one or more of the following:

- **Software Development**:
  - Projects that deliver or enhance software applications.
  - Project Team: Application Development
- **Clinical Research Data Management**:

  - Projects that require mixture of technology and components to enable a strategic usage (collection, management, interaction, and interface) of Clinical/Personal Health Information and related data.
  - Project Team: Warehouse Programmers
- **Research Database Support**:
  - Projects that build, manage, or maintain databases or database objects that interface or interact with CRI Applications, Research, and non-clinical Information.
  - Project Team: Database Administrators
- **Systems Management**:
  - Projects related to the installation, management, and maintenance of hardware and software, services, networking, and infrastructure.
  - Project Team: System Administrators

Phase Two: Categorization

Once a project has been through the Project Definition Phase, it will be categorized by the following criteria:

- **Data Confidentiality**: Data are classified according to HOP Data Classification Guidelines (HOP 5.8.21)
  - **Confidential/High Risk**: Information or data is classified as Confidential if it must be protected from unauthorized disclosure or public release based on state or federal law or regulation, and by applicable legal agreement to the extent permitted by law.
  - **Controlled:** The Controlled classification applies to information or data that is not generally created for or made available for public consumption but may be subject to release to the public through request via the Texas Public Information Act or similar State or Federal law.
  - **Published**: Published information or data includes all data made available to the public through posting to public websites, distribution through email, social media, print publications or other media outlets. Information that is widely available to the public through publications, pamphlets, web content, and other distribution methods
- **Programming Rigor**:
  - High: Projects that warrant the implementation of the full quality management system based on client requirement, potential harm resulting from data retrieved, collected, or managed, or local, state, or federal guidelines
    - Example: Industry sponsored clinical trials for submission to the FDA; data used for clinical decisions
  - Medium: Projects supporting research or operations that don't necessitate "High" programming rigor but may still support or manage Confidential/High Risk data.
    - Example: local investigator-initiated studies, prepatory to research operations, support, and management systems
  - Low: Projects with lower accuracy needs or long-term operational expectations
    - Example: feasibility queries, one-off reports, or single use programs

- **Project Oversight**:
    - The type and extent of project oversight is determined by the project definition, project team, data confidentiality, and programming rigor.
        - Example: Projects with high rigor and high-risk clinical information may be managed by
            - Clinical Research Informatics Specialist: Assures investigator timelines met, data are accurate, data delivery meets investigator requirements.
            - Programming Team Lead: Assures code meets all applicable standards, version control requirements met, the correct resources are assigned.
            - Post Award Manager: Assures projects are delivered within resource constraints and sponsor requirements.

Phase Three: Requirements

The Requirements Phase supports the allocation and derivation of the project requirements down to the system and data elements representing the lowest level of the design. This phase formally defines the detailed functional user requirements based on the high-level requirements identified during Pre-Award operations (within the Project Life Cycle Process).

The requirements are defined in this phase to a level of detail sufficient for system and software design to proceed based on the results of the SDLC Categorization Phase.

Requirements shall:

- Accurately define a system that meets end-users' operational mission requirements within specified cost and schedule constraints.
- Sufficiently link the needs of the end users to the data, system, and system elements that, in turn, enable data and system elements to be designed and developed.
- Provide insight into the interactions among various functions to achieve a set of balanced requirements based on user objectives.
- Develop a quantifiable set of performance and operational requirements by defining the functional boundaries of the system in terms of the behavior and properties to be provided.
- Define implementation constraints (stakeholder requirements or solution limitations).
- Define any exceptional security needs.

Requirements shall be written as:

- Necessary
- Unique
- Unambiguous
- Complete
- Consistent
- Singular
- Measurable/quantifiable
- Verifiable (e.g., Testable)

- Technically feasible/achievable/obtainable
- Traceable through test results
- Attributable to a stakeholder
- Able to be validated
- Operationally effective

The resultant system requirements are addressed at technical reviews and audits throughout the acquisition life cycle and captured in applicable program and system design and engineering technical documentation.

Phase Four: Design

The primary objective of the Design Phase is to transform the requirements identified in the Requirements Phase into a project specification. The specification shall include the design of the software or application, address any network requirements, and define any user, system, or data interfaces and is designed to:

1. Satisfy the project's system, functional, and data requirements
2. Communicate project needs in sufficient accuracy and detail to all project teams
3. Address and mitigate security risks
4. Gain necessary approval for progressing to the Development Phase

The design document shall address the following, as needed, based on the project's requirements:

1. Required data source(s)
2. Potential risks and defining mitigating features.
3. Security risk assessments.
4. Developing a conversion plan to migrate current data to the new system.
5. Determining the operating environment.
6. Defining major subsystems and their inputs and outputs
7. Allocating processes to resources
8. Contingency plans
9. Training (user, administrator, etc.)
10. Operating parameters
11. Maintenance mechanisms and schedules

Phase Five: Development

During this phase, the system or software is programmed, developed, or otherwise constructed. Testing is often described within the Development Phase; however, within CRI, they are described in two separate phases as projects that are categorized as "Data Confidentiality – Confidential/High Risk" and "Programming Rigor – High" dictate that extensive code review, system testing, and operational validation must occur significantly increasing resource requirements.

Development methodology is driven by how the project is defined in the Definition Phase (that determines the CRI programming environment) and Categorization Phase (this determines the

programming team or teams) with specifics detailed in corresponding CRI SOPs and CRI Work Instructions.

While the selection and determination of environment and team may dictate unique development features or requirements. All software and system development shares the following:

- Team or Project lead ensuring policy and procedures followed and all requirements met
- Accountability to the Post Award Manager (Project Lifecycle Management) to track time and effort
- Adherence to CRI programming conventions and Source code version control requirements

Phase Six: Testing

Testing may account for a significant investment of resources by programming, operational, and CRI technical management staff. Within this phase the review and testing of individual elements (units) for quality, accuracy, and usability feed back into the Development Phase until the service, software, or application are ready for the Acceptance Phase. When required testing plans and procedures may be written and managed by stakeholders outside of the CRI team

All software written by CRI personnel is both open to or required to undergo CRI Management/Team Lead/Peer code review and acceptance based on requirements, CRI SOPs, and CRI Work Instructions.

Phase Seven: Acceptance

This phase is initiated after the system has been tested and reviewed as complete by CRI and the project sponsor. The delivered software, system, or data is compared to performance and quality objectives established during Definition and Requirements Phases. Implementation may include user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes, etc., until all those items defined through the Pre-Award process have been delivered. This phase continues until the system is operating in production in accordance with the defined user requirements.

Phase Eight: Operations and Maintenance

The system operation is ongoing based on project specifications. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue if the system responds to the project's needs. When modifications are identified, the system may reenter the Definition Phase.

7.0 Deviations

7.1 Deviations from this policy

7.1.1 Deviations from this policy are handled according to section 7.0 in the CRI QMS Policy (CRI.POL-001).

8.0 Review and Revision of This Policy

8.1 Review and revision of this policy is handled according to section 9.0 in the CRI QMS Policy (CRI.POL-001).

9.0 Attachments

none

10.0 Revision History

Version No.	Revision Date	Description of Revision
0.0	07/09/2020	This is a draft procedure for trial use.
1.0	05/17/2022	This is the approved policy for dissemination
2.0	06/01/2022	Effective date changed