STANDARD OPERATING PROCEDURE 14
Warwick Clinical Trials Unit (WCTU) Computer System Development and Validation

Version:	5.0	Effective Date:	13 December 2022
Issue Date:	29 Nov 2022	Review Date:	13 December 2024
Author:	Ade Willis, Programming Team Manager, WCTU		
WCTU Reviewers:	WCTU Programming Team		
Sponsor Reviewers:	Mathew Gane, Research Governance & QA (Quality Assurance) Manager, Research & Impact Services (R&IS)		
WCTU approval:	Natalie Strickland, Head of Operations, WCTU		
Sponsor approval:	Jane Prewett, Deputy Director, Head of Research Governance (R&IS)		
Review Lead:	WCTU Programming Team		

Contents

Revision Chronology:	Effective date:	Reason for change:
Version 5.0	13 December 2022	Minor clarification updates to the system development procedures (4.3.1.1 to 4.3.1.9). Updates to the support and maintenance section to include handling requests for anonymising participants and allowing documented delegates to make requests (4.3.5) Added a new section for security maintenance (4.6)
Version 4.0	23 September 2020	Major update to the Software Development Process.
Version 3.1	10 October 2017	Updates to reflect the Programming Team's restructure and reference changes to supportive documents.
Version 3.0	24 November 2014	Major update to describe the new Software Development process and to amalgamate the Programming Team's guidelines into a single coherent procedure.
Version 2.2	07 January 2013	Update to remove the defunct guideline: G10-WCTU-PT-Standard Trial Objects and to remove the specified timelines for arranging system review meetings.
Version 2.1	10 February 2011	Update to acknowledge the Senior Project Manager and System Owner roles.
Version 2.0	21 January 2009	Re-write to incorporate new procedures for software development.
Version 1.0	March 2006	

STANDARD OPERATING PROCEDURE 14
WCTU Computer System Development and Validation

1. Purpose and Scope

The purpose of this Standard Operating Procedure (SOP) is to detail the computer system development and validation procedures for the Warwick Clinical Trials Unit's (WCTU) programming team. This SOP is applicable to the development of all clinical trial computer systems for trials that are sponsored by either the University of Warwick or by external organisations that have delegated computer system development responsibilities to the WCTU programming team.

2. Definitions

Computer Systems:	For the purpose of this SOP, a computer system is defined as the hardware, software, and associated documents that create, modify, maintain, archive, retrieve or transmit clinical trial digital information.
User Requirement Specification (URS):	A non-technical document written from a user's perspective detailing the requirements of the system.
Backlog Item:	Defines a distinct piece of functionality or a modification to existing functionality that is yet to be developed for a computer system.
Software Bug:	A fault in a computer system that causes incorrect or unexpected results or unintended behaviour.
Product Backlog:	A list of backlog items that are waiting to be developed as part of a sprint.
Sprint Backlog:	A list of product backlog items that have been assigned to a sprint.
Sprint:	An iteration of development work with a set time to complete the items listed in the sprint backlog.
Source Code Repository:	A system used to store computer code files and the history of changes made to those files.
Agile Software Development:	A general term for a set of frameworks and practices based on the Manifesto for Agile Software Development principles. See: https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/
Scrum Framework:	A software development agile framework that defines a number of tools and practices for managing the delivery of working software over the course of multiple iterations. See: https://www.agilealliance.org/glossary/scrum/
User Acceptance Testing (UAT):	A software testing process carried out by the user(s) of the system to ensure the requested requirements have been correctly implemented and the system is working as expected.
System Testing:	Testing conducted on complete components or the entire system to ensure compliance with the functional requirement specification.
Functional Requirement Specification (FRS)	A document used to define the functionality of a system i.e. what the system must do.

3. Background

The validation and qualification of computerised systems used in clinical trials is an EU legal framework requirement to ensure compliance with the principles of GCP as defined in the ICH Guideline for Good Clinical Practice (see https://www.ema.europa.eu/en/ich-e6-r2-good-clinical-practice).

This SOP outlines the software development and validation procedure that has been tailored specifically for the WCTU Programming Team to ensure best practice and regulatory compliance for the design, construction, validation, release, and subsequent amendments to clinical trial systems. The specified methods incorporate elements from agile development principles and the scrum framework to implement iterative work streams (sprints) for completing user requirements (backlog items).

4. Procedure

4.1 Responsibilities

WCTU Programming Team Manager:	• Oversee the development and delivery of the computer system(s) that are allocated to the WCTU programming team by a trial Chief Investigator (CI) as per the scope defined in the URS. • To ensure that systems are developed in accordance with recognised best practices and regulatory requirements.
Programming Team Members	• To carry out their respective roles in the design, development, testing, implementation, support, and maintenance of all systems.
System Owner (most commonly the Trial Manager/Coordinator):	• To review the software during development, provide feedback to the programming team, coordinate UAT and approve the system for release.
Statistician:	• To assist the system owner with reviewing the computer system, to define randomisation requirements, review and test the outcome of computer randomisation simulations and approve the randomisation algorithm for release.
ITS Server Provisioning:	• To provide hosting and backup services for all WCTU servers.

4.2 When?

The software development process begins when a sprint has been scheduled by the project's Senior Project Manager (SPM) (see SOP 42 'Clinical Data Management System (CDMS) Planning & Maintenance'). Development work can only be performed as part of a scheduled sprint. A request for system support (see section 4.3.5) is actioned after the request has been submitted to the programming team's helpdesk.

4.3 How?

4.3.1 System Development

The sprint backlog lists all the backlog items and bug fixes that are to be completed within the agreed time. The programming team are responsible for creating the software based on the system requirements specified in the sprint backlog, documenting features and functionality in the Functional Requirement Specification (FRS) and testing that the system works as expected. All programming tasks should be documented against a relevant sprint backlog item. To help ensure the system meets the requirements, the system owner will work with the programming team during the sprint to review prototypes and provide feedback.

4.3.1.1 Documentation

The table below lists the documents that are created/updated by members of the programming team during a sprint. Documents that are based on templates have been noted. All templates can be

downloaded from the programming team's SharePoint site. Completed documents must be saved in the project's SharePoint folder after a sprint has finished.

Document	Description	When Created/Updated
Sprint Back Log	Lists the backlog items that have been assigned to the sprint and the related programming tasks.	Created before the start of every sprint. Updated during the sprint by the programming team to record progress until the sprint is finished.
FRS	Describes the functionality of all system features.	Can only be updated by members of the programming team during a sprint and before the System Release Approval Form has been approved.
Test Plan (applicable for System Testing and UAT)	A collection of test cases based on the sprint backlog.	Created as soon as the sprint backlog items are known and updated throughout the sprint. Cases can be added/amended during the test process as new test scenarios are discovered.
Randomisation Check List **(Template 14-T-1)**	Lists several checks to ensure the randomisation has been developed correctly and validated by the statistician.	Required whenever a randomisation system is created or modified. Must be completed and approved by the study statistician and a member of the programming team.
System Change Log **(Template 14-T-2)**	Lists all changes between the current and previous version of the system.	Created for every sprint after the initial version. The content of this document can be generated by a build tool or manually updated by a member of the programming team.
System Release Approval Form **(Template 14-T-3)**	Authorises a new version of a system to be released.	Created for every major or minor version. Not required for revisions or release candidates. This document must be completed and approved by a member of the programming team and the system owner.
System Release Check List **(Template 14-T-4)**	Lists system release validation checks to ensure the system is correctly implemented on the live servers.	Required for any live system release. Must be completed and approved by a member of the programming team.

4.3.1.2 Randomisation

The algorithm used to randomly allocate a participant to a trial arm must be tested and approved by the project's nominated statistician before the system is released. The programming team will work with the statistician to ensure the allocation is performing as per the requirements specified in the Protocol and URS. For new randomisation systems or for any change to an existing randomisation, the Randomisation Check list (Template 14-T-1) must be completed and approved by the study statistician and the member of the programming team that created the randomisation.

4.3.1.3 Audit Trails

Any system collecting clinical trial data must implement an automated audit trail that records actions for the creation, modification, and deletion of all clinical trial data records. The audit trail must be able to facilitate the reconstruction of the history of events relating to a record. At a minimum this must include a reference to the individual performing the action, what action was performed, a record of any modified data and the date and time the action occurred. Audit trails for all new systems must include the reason for any modification or deletion. All audit data must be made accessible to authorised end users, either as part of the system or as an interactive report that can be queried.

4.3.1.4 User Access Controls

Access to all systems should require authentication against an individually assigned username and password. For systems where this is not possible, additional security measures must be applied to ensure individuals accessing the system can be identified and all audit trails are attributed to an individual.

All systems should apply role-based authorisation controls to ensure users are only permitted to access the features of the system that are appropriate for their job role. An automated audit trail must record when roles are assigned and revoked.

4.3.1.5 Patient Identifiable Data (PID)

Special consideration must be taken for systems processing PID such as PID specific system access roles, device management controls and location-based access restrictions. When PID is processed, efforts should be made to reduce unnecessary disclosure.

4.3.1.6 Prototyping/Review:

During a sprint, the programming team will release prototypes to demonstrate the requested features of a system. The system owner and/or delegates should review these prototypes to provide feedback to ensure all requirements are present before any formal testing commences.

4.3.1.7 System Testing (Programming Team)

For this SOP, system testing is a broad term used to refer to any level of testing activity that is carried out by the programming team to ensure the system performs as intended. Testing must be conducted to verify that the requirements specified in the FRS work as intended. The scope of the test plan is typically defined by the items listed in the sprint backlog but may include additional regression tests if any other related features of the system have been affected. Tests can be carried out manually or automated using test automation software. A system test plan must have all the test cases resolved and be approved by the tester(s) before the sprint can be considered complete.

4.3.1.8 Source Control

A source control management system must be used to track changes to the system code. Source code repositories should include a master branch that contains the final commit for each released version. Any changes to a system should first be made in a development branch and then merged back to the master branch when the code is ready to be released. All commits should be annotated with a brief description of what changed since the last commit.

The code used for each released version must be retrievable from the master branch by applying tags (or an equivalent) that specifies the system's version number. The final code for a released version should also be annotated with a release note that briefly describes any new features, updates, or bug fixes. The release note should be applied when merging a completed development branch back into the master.

4.3.1.9 System Versioning

Version numbers should consist of three numbers separated by dots to represent major, minor and revision. An additional release candidate alphanumeric string can be appended to the version number to indicate incremental builds prior to a version's final release e.g., 1.00 rc1 (1st pre-release build), 1.00 rc2 (2nd pre-release build), 1.00 (final released version). A sprint will typically result in a new version of a system, either a release candidate (a pre-release build) or the final released version.

What constitutes a major or minor increment can be subjective but as a general guide a major version is used when new features are added or a significant change in functionality has occurred, a minor version is used for amending existing functionality. For example, adding a new form would result in a major version, changing the validation rules on a small number of existing questions would be a minor version. A revision version should only be used to fix a feature that is not performing as expected e.g. software bugs, cosmetic User Interface issues, typos, documentation discrepancies etc.

All major, minor and revision releases must be merged back into the master branch and tagged accordingly (see 4.3.1.7 source control). Release candidate versions should not be merged into the master branch but should be tagged as a release candidate version in the development branch.

4.3.2 User Acceptance Testing (UAT)

The UAT for a specific feature can start after the following actions have been completed:

- The FRS must have been updated and made available to the system owner.
- Application code should be fully developed.
- System testing should be largely complete i.e. no showstoppers or major defects that would prohibit the UAT process, only cosmetic errors should be permitted.
- The UAT environment must be ready. This should emulate the live environment.

The system owner or delegate(s) is responsible for performing the tests and completing the UAT test plan. This process will typically require a number of test cycles to identity issues, have the programming team fix and update the system and for the system owner or delegates(s) to retest. The person completing the first cycle of testing should be the same person that completes any retests. The UAT test plan must have all the test cases resolved before the sprint can be considered complete.

4.3.3 System Release

A system can only be released after all sprints have been completed. For all major and minor releases, the system owner must complete the System Release Approval form. A revision release does not require approval from the system owner, but they should be notified why a revision is required (if they have not already been informed) and when the revision will be released. All system releases must have a system release check list completed.

4.3.4 System Migration

A system migration may be required if there is a change in the hosting infrastructure e.g. upgrading to a new server or if the data are being transferred to a new system e.g. a legacy system is being upgraded for interoperability reasons. In both cases a migration plan must be created to ensure there are no unintended alterations to the data. The migration plan needs to be tailored to each project but must include a migration test plan that can demonstrate that the system's data integrity has been maintained after the system has been successfully migrated.

4.3.5 Support and Maintenance

4.3.5.1 Responding to Helpdesk Requests

The programming team use the Service Now platform for users to log support and maintenance requests via a helpdesk. All requests must be made by either a system owner, a documented delegate or an appropriate senior manager e.g., a senior project manager or a senior member of the operations team. All helpdesk requests related to an active clinical trial should be linked to the trial team's group email address to ensure all helpdesk correspondences are copied to the trial team members.

All data clean, user access, report creation, anonymisation and technical support requests must be logged on the helpdesk before being actioned. The programming team member who completes the request must resolve the helpdesk call.

4.3.5.2 Data Modifications

Any scripts created to change data must be saved and reference the helpdesk incident number. If the system's audit trail supports logging the reason why the data was changed then the incident number must be recorded as part of the audit record.

4.3.5.3 Ad hoc Report Development

All reports must be tested and approved by the system owner or delegate(s) before being made available on the live report server.

4.3.5.4 Anonymising Participants

System owners or delegate(s) can request that identifiable data is permanently removed from a database. Requests to anonymise data must specify all the field(s) that contain the identifiable data to be removed. When this request is actioned, the data in the specified field(s) must be deleted. Identifiable data stored in the audit trail must also be deleted and a flag must be added to each audit record to indicate the audit data has been anonymised.

4.3.6 Security Maintenance

All software used and created by the programming team must be regularly reviewed and updated to ensure patches and/or other security or maintenance measures are applied in a timely manner. Refer to IS07: Systems Management Policy (warwick.ac.uk) and IS09: Software Management Policy (warwick.ac.uk)

4.3.7 System Archiving

Clinical trial data must be retained in a format that ensures the long-term reliability, retrieval, and accurate reproducibility of the original data. Notification to archive a system must be sent to the programming team's helpdesk by the system owner or an appropriate senior manager. The archiving process, order of archiving events and timelines is tailored to each system but typically involves

preventing any further updates to the database, revoking access to the system, removing the online application, and removing reports. Additional requirements may also involve the redaction of data from the archived database e.g., anonymising participants.

List of abbreviations

CI	Chief Investigator
FRS	Functional Requirement Specification
GCP	Good Clinical Practice
ICH	International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use
ITS	IT Services
PID	Personal Identifiable Data
QA	Quality Assurance
R&IS	Research & Impact Services
SOP	Standard Operating Procedures
UAT	User Acceptance Testing
URS	User Requirements Specification
WCTU	Warwick Clinical Trials Unit