Frederick National Laboratory
for Cancer Research

operated by ***Leidos Biomedical Research, Inc.***

Software Development Plan
for the
Center for Strategic Scientific Initiatives
Data Coordinating Center

Contents

Introduction 2
Overview: 2
Purpose 2
Definitions, Acronyms and Abbreviations 2
Agile Development Overview 3
Software Development Process 4
Requirements: 4
Programming Practices and Tools 6
Selected Technology Stack 6
Iterations 8
Iteration 1 – Setup and installations 9
Iteration 2 – Initiate work with the first data type 9
Iteration 3 – Continue work on the web portal 9
Customer Demonstration # 1 9
Iteration 4 – Upload interface, security scans 10
Iteration 5 – Query interface, move to DMZ 10
Customer Demonstration # 2 11
Iteration 6 – Normalize release process and schedule 11
Iteration 7 – Documentation/transition 11
Customer Demonstration # 3 12
Training 13
Appendix 1 13
Proposed Backlog Iteration Schedule 13

Table of Figures

Figure 1. The Agile Method 3
Figure 2. Proposed System ArchitectureArchitecture 7
Figure 3. Notional Server Architecture for Production 7

Introduction

Overview:

Data Repositories are important tools in cancer research, providing safe and sustainable locations to house data, providing access to input data for meta-analyses and allowing researchers to collaborate and share information across a common resource. The Center for Strategic Scientific Initiatives (CSSI) sponsors a diverse array of projects that generate data sets that vary in content and format yet are related across certain defining characteristics or metadata. Integrated management of the data sets across all sponsored projects will make the data more accessible and fundamentally more easily accessed and potentially reused by the cancer research community.

This development plan outlines the creation of a resource to comprise Data Coordinating Center (DCC) that will house and manage access to data created in a select set of CSSI projects as defined in the previously submitted project management plan (PMP).

Purpose

1. Provide a common location and web access to data from disparate data types including gene expression results from Next Generation Sequencing, microarray experiments, histopathological images, metabolomic data and proteomics data, allowing for easy access by multiple collaborators and researchers located at different geographic locations. Maintain flexibility to be able to handle new data types that are unspecified.
2. Store the data in one common location so that biological insights, which would otherwise be missed by having data in multiple locations, can be made.
3. Apply the information gained from one study to multiple studies and projects.
4. Allow users to search the metadata from each study to identify datasets of interest.
5. Allow for continuing annotation of existing data to enrich its scientific value,
6. Develop data storage and data mining modules that can be applied across studies thereby avoiding duplication of effort and saving costs.
7. Develop and/or adopt common vocabularies, data standards and ontologies for data representation, storage and comparison.

Definitions, Acronyms and Abbreviations

Backlog – Prioritized list of features (User stories) maintained on Jira

Customer – CSSI and their designees

Daily Review – daily team meeting that is held for the development team to coordinate tasks. A project lead (or designee) will also attend to take notes, share news, assist with problem solving, and keep updated on progress of the product.

ETL – Extract, Transform, Load - Extract data from homogeneous or heterogeneous data sources. Transforms the data for storing it in the proper format or structure for the purposes of querying and analysis, Loads it into the final target

Iteration – incremental development cycle that will be four weeks each

Jira – A software development and tracking product, developed by Atlassian that is used to track bugs, issues, and project management functions

MVP – Minimum Viable Product

Agile Development Overview

Agile software development is a set of methods in which requirements continuously evolve through collaboration between functional teams. The Agile method supports frequent deployments of new features and enhancements, thereby providing the opportunity for periodic demos and rapid customer feedback. This interactive approach incorporates the customer and enables flexibility, response to change and produces robust documentation. The continued interaction of the customer with the team ensures customer satisfaction and the continuous delivery of effective software. The Agile method breaks the tasks into small increments or Iterations that last two to four weeks each. Each Iteration involves planning, requirements analysis, and design, coding, and testing with a working product (not necessarily a final release) available at the end of the Iteration. Multiple Iterations are required to release a final product. The use of Iterations in this method allows risk to be minimized and bugs to be identified and resolved incrementally. However, this iterative method also means that the development of the software is a "work in progress" throughout the process. The interaction with the customer ensures that the customer is aware of progress can identify any changed needs or misinterpretations.

The main goal of Agile development is to deliver useful software to the customer and the CSSI DCC project team will work towards this goal. This is intended to be a fluid process as contracts and customer expectation and impact the project execution.

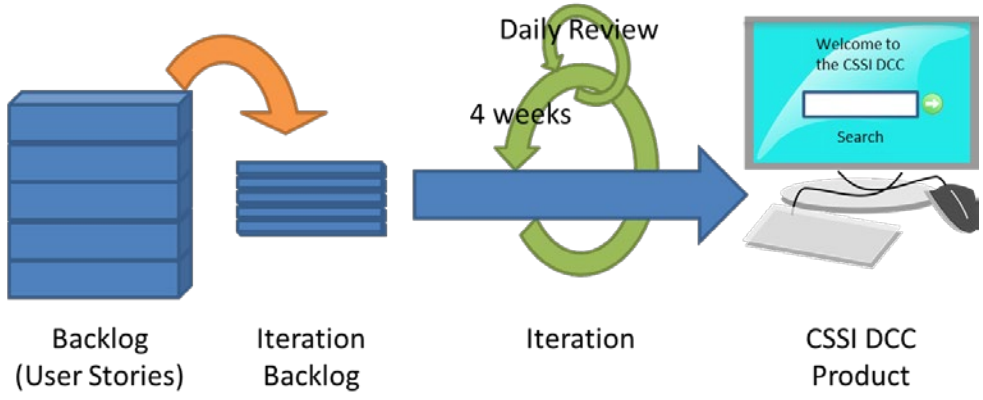

Figure 1. The Agile Method

Software Development Process

Requirements:

1. A standard flexible data coordination framework to house multiple subtypes of biologic data generated in support of cancer research funded or supported by the CSSI.
2. Investigators must be able to accession data into the store, provide a minimal set of metadata that describes the study that was performed, the type of experiment or assay performed, the input and output information whether observation or calculated and the pedigree and/or provenance of the input samples.
3. Information must be provided about any controls or standards used and the file types to be included in the data store in a standardized format.
4. Information submitted must be validated to ensure that it meets the criteria for the repository on several levels.
5. Researchers must be able to access data in the store either for inspection or for download. They should also be able to query across submissions for data with defined characteristics in which they are interested.
6. The technical architecture must be flexible, scalable and suited to distributed computing.
7. The delivered prototype should contain the features defined as the minimal viable product (MVP).

The project requires a data portal that not only meets the current needs from existing projects but will also be flexible and adaptable to the future projects and new data sets. The application will adapt modules from several existing FNLCR applications thereby embracing past successes, avoiding duplication and decreasing the time to results. Primarily two projects have been considered as adoptable frameworks, SysBioCube and the second generation of TCGA. The DCC will incorporate technical and design aspects from each of these. In addition, wherever possible, the ISA-Tab framework for metadata definition will be used as a governing standard to format data for submission and download. ISA-Tab is a well-accepted, well-defined set of templates and tools that allows researchers to fully characterize and describe data of all types. It also allows for the flexible extension of existing standards as novel formats and assay types are developed.

Development of the data portal will leverage the regular demonstrations of working software that will provide a platform for customer discussion and the refinement of functionality. The goal is to create and maintain a working system throughout development to which features can be added and matured over the entire period rather than to deliver a working system only at the end of the development period.

System and functional requirements will be tracked as work items in a Jira tracker so that progress can be monitored on an Iteration-by-Iteration basis and throughout overall development. The Agile method requires a well-groomed backlog of stories that have been elaborated by both technical staff and SMEs so that realistic estimates of the development needed can be made for each input story. Stories are added to Iterations only after they are elaborated sufficiently that the feature described can be fully implemented and tested. The definition of the completed feature must be in place for each story as

well. The number of Iterations required to complete a defined feature are determined by the technical estimation created for each story by the development team. Iterations are created by combining the number of stories based on the technical estimation that can reasonably be completed in that time period allowing for adjustment based on any new information that might be uncovered as a feature is created and the availability and technical skills of the team.

Software testing will be performed in conjunction with development so that metrics can be developed to ensure that the system performs as designed and meets the needs of the stakeholders. Web vulnerability scans will be performed using AppScan and Nessus early and often to identify vulnerabilities in system and software architecture and to ensure that it meets required federal standards for security and accessibility. Security and certification requirements will be addressed continuously across development to ensure continued authority to operate.

Specific documentation and deliverables will be tailored to suit the available time and resources for the project but adhering to the major stage gate transitions and review processes as required by CSSI. Each Iteration will be planned immediately prior to its commencement in conjunction with the project managers and the technical team, so that only an achievable number of stories are added to any Iteration. If needed, newly reprioritized work can be added to an Iteration but an equivalent amount of planned work must be removed in order to make consistent progress on the backlog of user stories. Defects or bugs uncovered during testing or development should be considered high priority to minimize the amount of technical debt that accumulates. Technical debt is any issue that cannot be resolved in the current Iteration with the available resources and technologies. An example of technical debt is a defect that is not considered severe enough to block a planned release or for which a viable work-around exists. It is good practice to perform an Iteration Retrospective that reviews the immediate past Iteration to uncover areas that the team can improve to facilitate development. However, the timeline for development is aggressive so that it may not be time-efficient to do a retrospective for each Iteration. Iterations conclude with a review of the working code for the team and periodically for the customer. Not all review meetings need to include the customer but it is important to frequently review with the customer to ensure that the final delivered product matches the envisioned design and functionality and to validate that the business requirements to be supported by the system have not shifted away from the original needs. Monthly reviews (after each four week Iteration) will be held with CSSI during the normally scheduled meeting time. Daily reviews within the development team will ensure that issues blocking development are addressed in a timely manner so that progress does not stall. As needed, internal meetings to discuss stories that need additional elaboration will be required to answer questions or clarify details. The technical team will document any work items that need clarification on the work item tracker (JIRA).

Programming Practices and Tools

The DCC team will be using Agile programming practices and tools including iterative planning/development and continuous integration.

Programming Practice	Tool	Tool Description
Iterative Planning and Development	Atlassian Jira	Jira is a proprietary issue tracking product, developed by Atlassian. It provides bug tracking, issue tracking, and project management functions.
Continuous Integration	Jenkins	Jenkins is an open source continuous integration tool written in Java. Software engineers check in when their individual task/issue/change is completed and then perform a merge.

Selected Technology Stack

The selection below represents a solid and implementable approach to several architectural aspects. It takes into account leading open source technologies as well as leveraging existing NCI/FNLCR assets.

Persistence: Oracle, MongoDB, NFS file storage, Cleversafe object storage

Application framework: PHP/Python, JavaScript

Server operating system: Ubuntu 14.04 LTS/ClevOS

CI software: Jenkins

QA testing frameworks: Minitest unit tests, Cucumber, Selenium

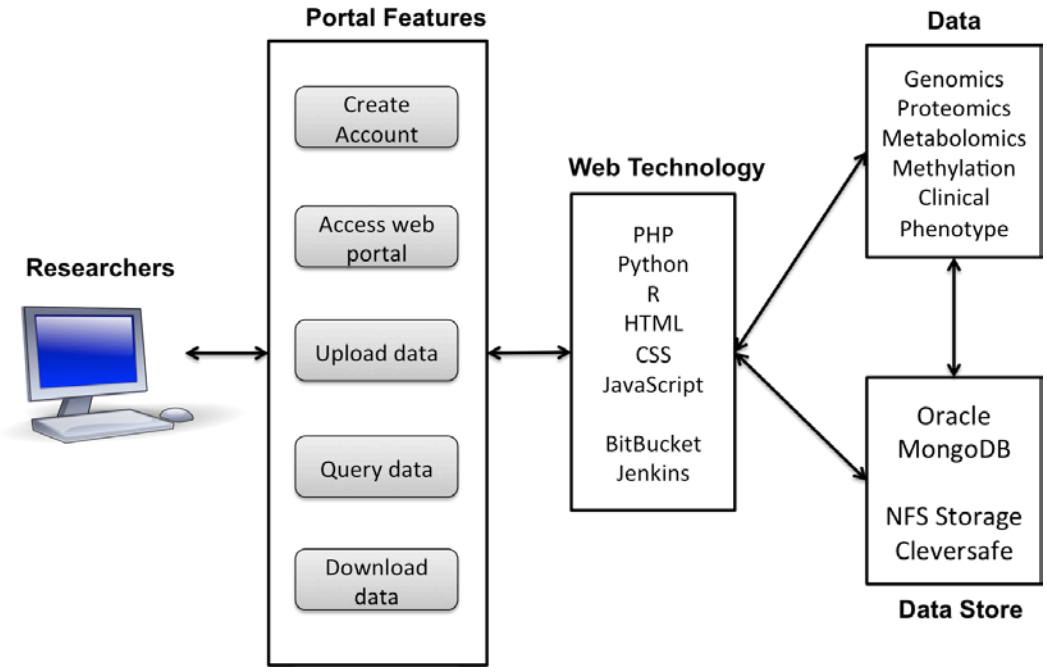

Figure 2. Proposed System Architecture

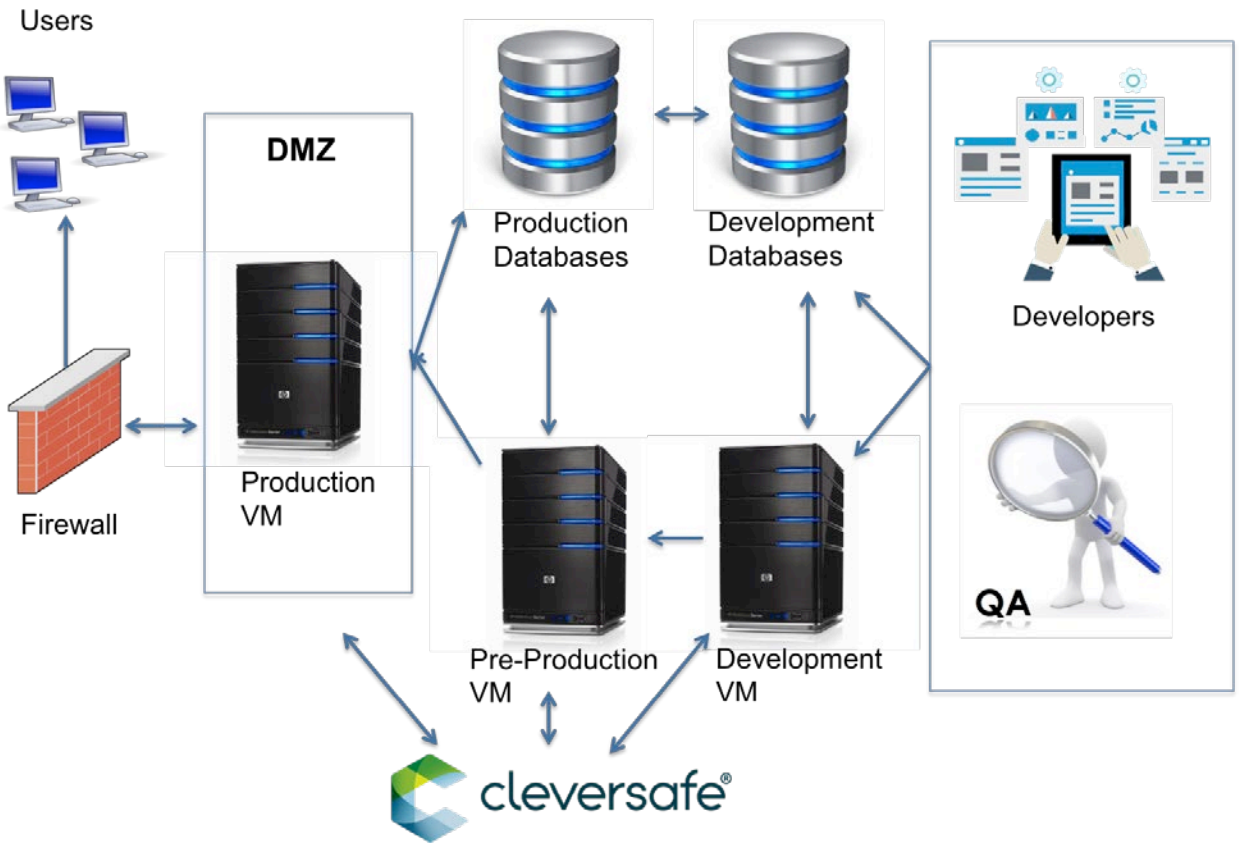

Figure 3. Notional Server Architecture for Production

The technology mix addresses and provides support for the following project requirements:

- Support Agile software development methodologies, continuous integration, automated QA and deployment
- Proven programming language and framework with wide adoption and community support
- Dual database approach with relational data integrity, persistence with transactions, plus flexibility of a document database for future querying and data warehousing activities
- Fine-grained role-based user security and privileges
- Robust and modern system security with low attack surfaces and fewer vectors for intrusion
- Lightweight yet heavy-duty application framework that allows compartmentalization of code and concerns into services, controllers, views and API interfaces
- Ability to create single-page and classic MVC applications
- Ability to serve up multiple data views, i.e. JSON, XML or server-side rendered webpages based on business need
- Ability to test and deploy subcomponents of the application without having to redeploy large binaries

The CSSI DCC will be evaluated for positioning within the Science DMZ. While this may not be appropriate for the pilot, the DCC will be built with the requirements for the Science DMZ taken into consideration.

Additionally, due to the time required to have the Cleversafe® system available, the pilot CSSI DCC will likely be set up locally on an NFS mount. The pilot data is not of significant size and will take up minimal space on the local system. The DCC will be transitioned to the Cleversafe® system when it is ready for use. This will diminish the risk of possibly being required to re-upload data once it's fully functional. SysBioCube is currently on an NFS system and this will also facilitate the development of the pilot DCC.

Iterations

The ability to adopt and modify an existing repository framework will ensure a successful prototype by building off a known data model and user interface. Customizing and refactoring a platform rather than establishing one *de novo* will jump-start development and reduce the overall time and risk to delivery.

Development will ramp up over the initial two months as the technical team is established allowing for the formation of high-functioning development team. Additional documentation effort will also be undertaken during this period to ensure that the requirements have been adequately described for development to begin. Detailed functional gap analysis between existing behavior of similar features in SysBioCube and the desired functionality for the DCC will be undertaken to determine plan for

customization. The technical team will determine the precise Iteration schedule based on available resources. Iterations will be adjusted and changed throughout the development process. The following is the planned schedule of activities. *This is an estimate based on our expectations as of January 31, but consistent with the Agile process this will be continuously updated.*

Iteration 1 – Setup and installations

The first Iteration will be the primary planning phase for the development cycle. The project team will be identified and trained, gap analysis of existing software will be performed, and documentation outlining the database will be developed.

- Setup virtual servers for development, pre-production and production
- Test the web framework and functionality
- Establish initial database schemas, users and database access controls
- Training for new programmers on existing resources to understand their roles and project requirements
- Continuing technical documentation
  - requirements elaboration
  - systems and activity diagrams
  - Concept of Operations documentation
- Functional and Gap analysis of SysBioCube

Iteration 2 – Initiate work with the first data type

Gap analysis from Iteration 1 will be reviewed and used to elaborate technical documentation for the development of the system. Initial coding and customization will begin during this Iteration.

- Finalize Architectural specification
  - Initiate security and design reviews where needed
- Review initial gap analysis findings and develop customization details
- Establish metadata templates and naming conventions
  - Determine the ISA-Tab templates required
- Begin customization of the web portal
- Begin development of the user access module for data uploads and downloads
- Begin development of the data explorer module
- Begin development of the query module

Iteration 3 – Continue work on the web portal

The third Iteration will encompass the continued creation and refinement of the web portal where users will interact with the DCC.

- Continued maturation of the user access module for data uploads and downloads
- Continued maturation of the data explorer module
- Continued maturation of the query module
- Develop code to parse and load initial data into the database (ETL)
- Create additional database tables, schemas, collections

Customer Demonstration # 1

Presented during a regularly scheduled meeting in May 2016

The first demo of the DCC portal will be scheduled for the end of the 3rd month of development to allow sufficient time to complete additional technical and systems documentation as well as establish development environments and support systems such as controlled code repositories and to adapt SysBioCube to serve as the foundation for the DCC application.

Expected completed milestones:

- Backend
    - Virtual servers with minimal configuration to start development
    - Development, test, production environment for the project
    - Database schemas, tables, views will be created
    - User roles and permissions for the first data type
    - ISA-Tab templates for the first data type
    - Parsers and scripts for the first data type
    - Complete ETL for the first data type
- Web Interface
    - Home page with minimal information
    - Login interface for accessing the data
    - Demo page for data uploads (non-functioning)
    - Client side validation of uploaded data
    - Validation checks for data formats
    - Pipeline for uploaded data to be added to the OS and database
- Initial data loading- single data type
    - Demonstration of unrestricted data sharing
    - Demonstration of restricted data sharing

Iteration 4 – Upload interface, security scans

The upload and validation features will be created and refined. Milestone releases will be periodically designated so that the product can be moved to each more advanced tier in a consistent and efficient process.

- Functional upload interface
- Complete validation pipeline for the data and metadata
- Security scans
- Testing and validation

Iteration 5 – Query interface, move to DMZ

The initial shift to the network area known as the Demilitarized Zone (DMZ) is required prior to exposing any application beyond the NCI/ FNLCR firewall is controlled by authority outside of the technical team, additional documentation may need to be developed in order for this transition to be approved.

- Functional query and search interface
- Bug fixes for new and existing features
- Submit for DMZ transition approvals
- Site to be made available out of the firewall

- Functional testing by QA

Customer Demonstration # 2

Presented during a regularly scheduled meeting in July 2016

This will be the first live presentation of a production DCC portal. Customer feedback during the demo will be used to adjust features as needed before the final release of the pilot.

Expected completed milestones:

- Backend
  - Validators for the initial data type
  - Scripts to enable version control in the OS and database
  - Integrated upload and validation processes to enable a smooth flow between the web interface and the backend database
  - Unit tests for the main features
  - Submit for Nessus and Appscan security scans
  - Review security reports and fix any high or medium level security issues
  - Submit for DMZ exception and follow all suggested protocols for enabling the site outside the firewall
- Web Interface
  - Data upload page for the selected data type
  - Client side validation and success/error messages after data uploads
  - Multiple file uploads for the web interface
  - Simple query interface to search data
  - JavaScript enabled interactive filtering of the data
  - Individual downloads of the selected data

Iteration 6 – Normalize release process and schedule

The planned sixth Iteration will cover validation and testing, which will assist with release planning.

- Iterative ETL and validation of remaining data types from the project
- Continued Functional Testing by QA
- Planning for User Acceptance Training (UAT) and user training
- Release Planning and periodic releases to production tier

Iteration 7 – Documentation/transition

The final Iteration will focus on release of the final product as well as finalizing all technical documentation.

- Last production release of the prototype
- Finalize all documentation
- UAT and initial user training for the prototype
- Lessons learned, document any changes needed for the next phase

Customer Demonstration # 3

Presented during a regularly scheduled meeting in September 2016

This will be the final demo of the pilot. Suggestions and feedback will be used for phase 3 development

Expected completed milestones:

- Backend
  - Consistently add other projects and data
  - Develop ISA-Tab templates for the data
  - Create validators for the data as they are added
  - Improve search and query interface based on the new data
  - Unit tests for the new features
  - Download interface to enable downloads of all data
  - Documentation on the available features
  - Periodic Nessus and Appscan security scans
  - Review security reports and fix any high or medium level security issues
- Web Interface
  - Functional data download interface
  - Upload, download, search and filter on all available data
  - Multiple user accounts and access
  - Documentation and FAQ pages are enabled
  - A functional pilot interface to move into phase 3

Training

The following stakeholders have the option to be trained on the use of the CSSI DCC when the pilot is released.

Name	Title	Affiliations
Rachana Agarwal	Technical Project Manager	FNLCR
Michelle Berny-Lang	Project Manager	CSSI
Braulio Cabral	Director , CBIIT Technical Operations Support	FNLCR
Jim Cherry	Assistant Project Officer	OSO
Jack Collins	Director, ABCC	FNLCR
Sharon Gaheen	TPM II, Genome Data Commons	FNLCR
Dan Gallahan	Deputy Director	DCB
Emily Greenspan	Program Director	CSSI
Sean Hanlon	Program Director	DCB (PSON)
Dianna Kelly	IT Program Manager	OSO
Chris Kinsinger	Program Manager	CSSI
Nastaran Kuhn	Program Director	DCB (PSON)
Jerry Lee	Deputy Director	CSSI
Joan Pontius	Bioinformatics Analyst IV	FNLCR
Henry Rodriguez	Director	CSSI
Dinah Singer	Director	DCB
Eric Stahlberg	Bioinformatics Scientist III	FNLCR
Kathy Terlesky	Director, Project Management Operations	FNLCR
Greg Warth	ABCC	FNLCR
Gordon Whiteley	Director, Clinical Proteomics	FNLCR

Appendix 1

Proposed Backlog Iteration Schedule

Primary Category	User	User Story	Rationale	Iteration
Administration	System Admin	want analysts and developers of the coordinating center to have access to a minimum of disk storage, CPU and access to the database and files of the coordinating center	analysts and developers of the data coordinating center can efficiently run scripts, programs and analysis of production data and pending data.	1

Primary Category	User	User Story	Rationale	Iteration
Documentation	Everyone	Know the definition of terms and vocabulary being used at the DCC including: data types, metadata, ontologies, copy number variation etc. and ensure that they are being used consistently.	plan to use those controlled vocabularies that will allow my data set to be appropriately labled and become integrated with other studies -- commonly used terms will not be taken out of context and mis-used.	1
Documentation	Everyone	The user manual to include: the definition of terms used at the DCC to be organized in a glossary; description of submission, access and QC including how datasets were validated; navigating the portal	I can look up terms easily because they are sorted alphabetically perform my own QA and understand the QA that was performed on the datasets	1
Documentation	Submitter	The user manual to include: documentation describing submission, access and quality control at the DCC; file formats and specifications; example submissions	my submission can be efficient and consistent with the setup at the DCC my upload will pass validation	1
documentation	Everyone	have access to definitions for terms, and have these terms be consistent with their use within the DCC	minimize ambiguity in communications	1
Documentation	End User	to be provided a concise overview of the DCC	can quickly decide if the DCC overlaps with my interests	1
Administration	System Admin	Create users and user groups	so that I can assign different permissions to them	2
Administration	System Admin	have access control for the uploaded documents	assign different levels of access to all entities involved in DCC	2
Administration	System Admin	create roles and permissions	so that I can separate out access based on logged user	2
Upload	submitter	have a formal way to represent a machine readable data standard that describes an experiment (For example as provided by the use of ISAtab)	describe the steps of how the data was determined, the experiment was performed and how each sample was processed	1
Documentation	Everyone	help section with Tutorials/training manuals, Glossary, FAQ, policies and regulations etc.	maneuver the DCC	3
Documentation	Everyone	have access to an online User Guide	have one central location where I can keep up to date on file format specifications, upload procedures as well as how to access data set submissions	3

Primary Category	User	User Story	Rationale	Iteration
Upload	End User	want text files to be in tsv rather than csv format	so that fields can include commas without hindering the parsing of data	3
Upload	End User	files in data submissions that represent text only should be in machine readable format (text file) rather than Word or PDF, and files in data submissions that represent columns of values should be submitted as tab delimited text files, instead of excel spreadsheets.	use UNIX and parsing scripts such as Perl,Python and R to access the data and text, and at the same time, can always use excel to open the file	3
Upload	End User	no empty spaces, brackets, parenthesis and special characters in directory and file names	use UNIX or other appropriate platform to access the data	3
Upload	System Admin	have a user interface and documentation that passes 508 compliance	pass government requirements for accessibility	3
Upload	submitter	want to verify a checksum or md5 file for my file uploads	I can confirm that my upload was successful	3
Upload	submitter	upload data sets using a web interface	don't need to know UNIX to deposit data	3
Upload	submitter	upload one or more than one file in a single submission	so that I can associate all files to a submission	4
Upload	submitter	have version control assigned to my upload	communicate with collaborators about which revision of data they are using	4
Upload	submitter	be able to submit the metadata for my submission as a flat file included in the submission. The DCC has software that parses the metadata from the flat file.	submit a data set without having to fill out online forms for each entry, and so that I can avoid errors in filling out forms	4
Upload	Everyone	verify that the format and contents of my submission are valid after upload but before making public (positive acceptance)	any invalid submissions will not be made public or completely processed	4
Portal	End User	have a search function across data sets	to allow me to find relationships in the data	5
Portal	End User	ability to search data by typing in keywords or metadata terms.	obtain reports or data for further analysis	5
Portal	End User	be able to download data sets based on what metadata they are mapped to- such as cell line name or ID	retrieve those data sets that match my research interests	5

Primary Category	User	User Story	Rationale	Iteration
Portal	End User	be able to browse metadata and have access to files that match my selections	be able to download data sets that match my research interests	5
Documentation	End User	want to be able to report bugs or errors	so that they can be remedied	6
Portal	End User	want a checksum or md5 file for my file downloads	confirm that my download was successful by comparing the size of my download with expected size of download	6
Upload	submitter	verify(test) that the format and contents of my submission are valid before upload, for example, through a distributed validator that I can run on my computer before submission	know of any inconsistencies of my submission such as inclusion of unintentional files, misnamed files, format problems, deviations from controlled vocabularies before I go to the trouble to upload them.	6