User Epic: Participate in NGDS

Version 0.1

*1/4/2013*

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Reason** |
| 0.1 | P. Bruschi | 1/4/2013 | Initial Draft Created based on NGDS Software Requirements Specification Version 2.7 |

Table of Contents

[User Epic: Participate in NGDS 1](#_Toc345071352)

[Version 0.1 1](#_Toc345071353)

[Version History 2](#_Toc345071354)

[Table of Contents 3](#_Toc345071355)

[1 User Epic 4](#_Toc345071356)

[2 Effort Estimate 6](#_Toc345071357)

# User Epic

The first user epic should be used to demonstrate minimal functionality of the NGDS. Assume that the NGDS system is installed centrally but has no “node-in-the-box” NGDS nodes registered. In order for an organization to participate in NGDS, it first must have data to share and must install NGDS software on a local host machine. Assume that the local node has the NGDS software installed and is “ready to go”.

|  |  |  |
| --- | --- | --- |
| Epic ID | | **Epic\_001** |
| Epic Name | | **Participate in NGDS Using Tier 1 File Uploads** |
| **Short Description** | | As an organization that wishes to participate in NGDS, we want to assure that our information can be easily uploaded to our local repository and found through the central NGDS User Interface. Therefore, our local Node Administrator will set up the local computers to host our local repository and then our Node Administrator will register our node with the central NGDS. Then our Data Submitter will upload an example file according to the Content Model (i.e. a Tier 1 file) into our local repository and create a resource. During the process of uploading the submitter must complete the metadata form. Our Data Steward will then make the resource public so that the metadata of the resource can be harvested by the central NGDS.  After the metadata is properly harvested centrally, we want to verify that the location of the resource can be found using the central NGDS user interface. Therefore we want to validate that an end user can locate the resource by browsing the map, by using keyword filtering, and by seeing the bounding box of the resource displayed on the map. The end user should be able to download the resource data from our repository to their local machine. |
| **Actors** | | Node Administrator, NGDS Administrator, Data Submitter, Data Steward, End User |
| Pre-Conditions | | The Node Administrator has set up the local computers to host the local “node-in-a-box” node. This probably means that the NGDS Software package has been downloaded and installed.  An example file containing Tier 1 data file exists. |
| Success End Conditions | | The file is successfully uploaded and stored in the local NGDS repository.  The metadata record for the provided file is successfully created.  The metadata remains “private”, until it is made public by a Data Steward.  After the metadata is made public, it is harvested by the central NGDS repository allowing an end user to search, view, and download the data through the NGDS UI. |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Node Admin/ NDGS Node Administrator | The local node administrator contacts the NGDS Administrator to register a new “node-in-the-box” installation. Assume that a Node Administer sends an email to the NGDS Administer to register the node (the URL) to be harvested. The CKAN Harvester has a UI or file to add the node. The harvester will periodically pull via the CSW protocol (e.g. once a day). Include use cases *Register Node* and *Register New Nodes into NGDS* |
| 2 | Data Submitter | In the simplest case, the data submitter uploads a Tier 1 data file into the local repository. Include use cases *Upload New Files* and *Single File Upload.*  During the process of uploading the data submitter must complete the metadata form. The step is nearly satisfied by CKAN except the CKAN metadata form needs to be updated. We need a metadata form extension to meet the requirements. We might need to adjust the metadata model. This seems like a completely new extension. A possible value added is to generate a full text index of the uploaded document through SOLR parse and share that index via harvesting – but this is part of Epic\_002. Include *Create Metadata Record through Form*. |
| 3 | NGDS System | Include *Store File in Local NGDS Repository*  CKAN will make a URL for the file. By default, the URL “looks ugly”, but we might be able to change how the URL is generated. |
| 4 | NGDS System | A Tier 1 data file is “unstructured data” by NGDS and does not require a local template to create metadata. The NGDS   * Performs validation of metadata form based on content completeness. * Creates a data location URI and updates metadata form * Performs duplicate detection. This seems to be low priority and an extension will be needed to do this. * Validates, normalizes, converts geo-location attached to the data * Logs changes to metadata log file. CKAN has a logging mechanism that we will use. If we want something more, we will add more. * Provides a file upload success message.   There are CKAN APIs for triggering when data is uploaded. |
| 5 | Data Steward | Inspects the uploaded data and decides if the private information should be made public. If the metadata should be made public.  Include use case *Make Resource Public* |
| 6 | NGDS System | Bring public metadata to the central NGDS repository.  Include use case *Harvest Metadata*. |
| 7 | End User | Accesses the central NGDS user interface. Since the end user is browsing for information, the user does not need to be authenticated in the central NGDS database. |
| 8 | End User | Navigates to the Map Based Search page in the NGDS user interface. CKAN Spatial Extension should give a default map widget. The existing Map widget does not meet the specified UI.  Include use case *Map based Search* |
| 9 | End User | After locating a piece of metadata in the catalog, the user inspects the metadata record in more detail.  Include use case *View Metadata Record.* |
| 10 | End User | Selects data to deliver to user as a file. The NGDS retrieves the data file from the local node-in-the-box repository where it resides and downloads it onto the user’s chosen local directory in the specified format. CKAN has a link to the file to download.  Include use case *Download Data to a Local File* |
| Variants | | |
| Step | Actor | Description |
| 8a | End User | Utilizes keyword search to search against metadata keyword fields. CKAN seems to have this. Keyword based search is a simple Google-like search. Right now we will not make a dynamic search (enter and then click a button). We should also support a thesaurus in later phases. The issue is if the harvested data is searchable.  Include use case *Keyword Content-based Search* |
| 8b | End User | Given a collection of metadata records resulting from a search, extract bounding box extents and display on map. This appears to be difficult even though it is suppose to be part of the CKAN Spatial Extension but it does not appear to be working.  Include use case *Display Bounding Boxes for Search Results* |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
|  |  | |
|  |  | |

# Effort Estimate

The following table provides an effort estimation of the epic based on using CKAN as the platform. This first epic requires much investigation of CKAN in order to implement. Further note that not all aspects of the epic have been estimated. In particular, displaying maps and map based searches are not estimated and may be pushed off to later.

The table shows a set of tasks to be done, a description, and an effort estimate in days. The table also indicates what step or steps of the epic from which we derived the task. Our estimate is based upon analyzing the “out-of-the-box” CKAN UI in order to determine the gaps between what is offered out-of-the-box versus what the epic describes.

Some of the biggest risks become more obvious:

* CKAN Harvester needs to be better understood. Several tasks depend upon the capabilities of the CKAN Harvester and how to make it do what we want.
* CKAN terminology versus NGDS terminology needs to be understood so we can find the gaps. The preference is to use the NGDS terminology that comes from the NGDS requirements.
* Implementation of Maps is put off to later.
* The focus of the early sprints should on demonstrating how well the out-of-the-box CKAN UI supports the Anthrotech UI Design. We think we can come up a UI that will look nice but we do not know how close we will come to the NGDS UI design. One of the first steps is to build some static pages. Our focus is to see if we need to start from scratch and build our own UI, which will be more expensive to implement.

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Epic Step(s) | Description | Effort |
| Create Web Page for Allowing a NGDS Administrator to add a Node-in-a-box | 1 | * Investigate and implement ability to add a user as an NGDS Node Administrator (a basic privileged user who can access certain functionality such as adding a node-in-the-box URL). * Create a web page in the CKAN UI to allow the NGDS Administrator to add the node-in-the-box URL. * Investigate the CKAN Harvester. How do we tell CKAN what to harvest? * Utilize the CKAN UI as needed but use NGDS terminology for terms shown in the UI. * Extra Credit: Provide the ability to remove a node for harvesting. Investigate how already harvested information is cleaned up. | 3 -5 days |
| Upload a Resource | 2,3,4 | * No SOLR parsing and indexing. * Assume the simplest form of a resource. CKAN calls this a “collection”. * Investigate the CKAN “license”. This comes with CKAN yet, there is no NGDS requirement for this. * Investigate how CKAN creates a data location URI and updates metadata form. We can generate the URI ourselves. We plan to leave it as is and let CKAN do it. * Create reasonable example test files for uploading tier 1 data. This could be based on existing files from GTDA. * Implement the ability to upload a tier 1 resource data file. * Enhance CKAN’s metadata capability. For example, add Geolocation information to metadata. We are fine with the spatial location in the default CKAN form. However, we want to add the ability to specify the location on a map later. * Add Validation of metadata. Validate that required data has been entered. We expect that the CKAN metadata form comes with validation, but we need to check. * Investigate CKAN logging and log user activity. * Store uploaded information locally. This data is marked as “private” and will not be harvested. | 9 - 11 days |
| Make Resource Public | 5 | * Investigate CKAN’s concept of “Make Resource Active” and see how this might be used to implement the NGDS concept of making Resources Private and Public. For example, in CKAN making resources “Active” makes them visible. This is not quite the same for NGDS because a Data Submitter is a different role than a Data Steward. * Implement a UI page for a data steward to find and publish a resource. For each resource that data steward wishes to publish, make it visible to the harvester. * log user activity (investigation performed in Upload Resource step) * Extra Credit: Investigate how a public resource can be made private. | 10 days |
| Harvest Meta Data | 6 | * Investigate the CKAN Harvester. Learn about configuring the CKAN Harvester (e.g. can it be scheduled? Do we need to extend it?) We assume that the CKAN Harvester works automatically and data is “pulled” from the central NGDS node from Node-in-the-box. The CKAN Harvester should provide most of the functionality. But we should find out exactly what CKAN does with the harvested content. It might be that the harvested metadata is not the searchable through the CKAN default search capability. A transform might be needed to put the data into the correct CKAN database table. * Distinguish between "published" and "unpublished" data so that only published data can be harvested. This should be done in coordination with the “Make Resource Public” task. * Create a Central NGDS Node and at least one Node-in-the-box to demonstrate harvester. Test various scenarios to understand the CKAN harvester. Demonstrate that data is harvested | 10 days |
| Find, View, and retrieve Data | 7, 9, 8a 10 | * No Map Based Search. Just the Library Search at first. Establish a URL that can be access publically (Investigate if CKAN supports an “anonymous search” or does the user need to login?) The user should land on a page to allow for the library search. Also keep in mind that a user login capability will be needed later. Code should be easily extended later for this capability. * At first we should create some static pages in the CKAN UI to show feasibility of the design and show how close we can come to the Anthrotech Design. This is critical decision to be made if the team should abandon the CKAN UI completely and make a new UI with CKAN as the back end. * Investigate and update CKAN CSS. * Allow the NGDS Administrator to configure the URL. This might be done through the UI. * Construct the Library Search page to look similar to the Anthrotech Library Search Page. Allow the user to perform a library (keyword) based search of the metadata and view the results. Support the download of resources to a local computer. | 15 days |
| Design Documentation | N/A | * Provide Functional Description of the release. * Describe the architecture, component design, and concepts. * Describe design decisions that were made and potential impact on future releases. Many of these decisions will be made as a result of the investigations made during the above tasks. * Trace design to requirements * Provide a user guide describing the UI * Provide an installation guide for creating a node (i.e. the central NGDS node and a node-in-the-box). | 15 days |
| System Testing | N/A | * Design Test Harness and test strategy for running tests and reporting results * Develop System Test Cases based on the Epic functionality. * Develop Non-Functional Test Cases (e.g. Performance, usability). * Trace Test Cases to functional and non-functional requirements. * Run Test Cases and report results. | 15 days |