FOSSology and Ninka

by:

Jon von Kampen

Doug Richardson

James Thompson

Table of Contents

[Project Charter 1](#_Toc379367999)

[System Service Request 2](#_Toc379368000)

[Minimal Required Development Environment 2](#_Toc379368001)

[Optimal Development Environment 2](#_Toc379368002)

[Stakeholders 3](#_Toc379368003)

[Communication Management Plan 4](#_Toc379368004)

[Distribution System 5](#_Toc379368005)

[Data Flow Diagram 6](#_Toc379368006)

[System Structure 7](#_Toc379368007)

[Copyright Declarations and License Choice 8](#_Toc379368008)

[FOSSology 8](#_Toc379368009)

[Ninka 8](#_Toc379368010)

[Original Artifact Attribution 8](#_Toc379368011)

[Original Documentation 8](#_Toc379368012)

[Original Software 8](#_Toc379368013)

[Licensing Conflicts 8](#_Toc379368014)

[Change Log 9](#_Toc379368015)

[Overall Document 9](#_Toc379368016)

[Project Charter 9](#_Toc379368017)

[System Service Request 9](#_Toc379368018)

[Stakeholders 9](#_Toc379368019)

[Communication Management Plan 9](#_Toc379368020)

[Distribution System 9](#_Toc379368021)

[Data Flow Diagram 10](#_Toc379368022)

[System Structure 10](#_Toc379368023)

[Copyright and License Declarations 10](#_Toc379368024)

This document is licensed under the Creative Commons Attribution 4.0 International license by copyright holders Doug Richardson, Jon von Kampen, and James Thompson.

# Project Charter

The purpose of this project is to develop a tool to generate SPDX documents that combine the outputs of FOSSology and Ninka. The tool will sequentially scan a piece of software, whether it is a file or package, using FOSSology and Ninka. The tool will collect the output given by both programs and compare and combine the output into one SPDX document. The result will give end users the licensing information that they need to determine how the scanned software may be used.

If we have completed the main tasks above, we will attempt to optimize FOSSology’s execution time to likewise increase the speed of our program. Also, if time permits, we will attempt to design a web based user interface.

# System Service Request

Our project will sequentially scan a given piece of software with FOSSology and Ninka and combine their output into a single SPDX document. In order to do this we will need the following tools.

## Minimal Required Development Environment

* A Linux based computer or virtual machine
  + If developers use independent systems, the Linux distribution should be standardized
* Apache 2.x
* MySQL
* Postgres 8.3 or higher
* PHP 5
* Perl with cpan and the Text::Template module

## Optimal Development Environment

A dedicated development server shared by all developers will save time and resources and prevent “overlap” issues. A dedicated server will also enable us to test our tool as a remotely accessible web application should we reach that point. The server will function as both a web server and a database to simplify collaboration on a single system.

The dedicated system should meet the minimal requirements above, but ideally will have sufficient processing power, memory, and disk storage to emulate real-world scale use of FOSSology, Ninka, and our tool.

It is a possible compromise to use a minimal or less-powerful system at the beginning of the project, then upgrade or move to more powerful hardware when the document generator portion of the tool is ready for full-scale testing.

# Stakeholders

The following groups/people are identified stakeholders in the project. These entities will develop and support the project, or may be called upon to volunteer advice and assistance, or may wish to use the completed project for research and business purposes.

* Our team: Doug Richardson, Jon von Kampen, James Thompson
* The University of Nebraska at Omaha
* Hewlett Packard Company (HP)
* The FOSSology team: Bob Gobeille, Mary Laser, Dong Ma, Yao-bin Shi, Raino Lintulampi, and Dan Stangel
* The Ninka team: Yuki Manabe and Daniel M. German

# Communication Management Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Communication | Who | When | How |
| Team Meeting | Doug, James, John | Twice a week. E-mail whenever needed. | In person during and after class. E-mail outside of meetings. |
| The FOSSOLOGY Team | Bob Gobeille, Mary Laser, Dong Ma, Yao-bin Shi, Raino Lintulampi, and Dan Stangel | Every other week and when important developments happen | E-mail |
| The Ninka Team | Yuki Manabe and Daniel M. German | Every other week and when important developments happen | E-mail |

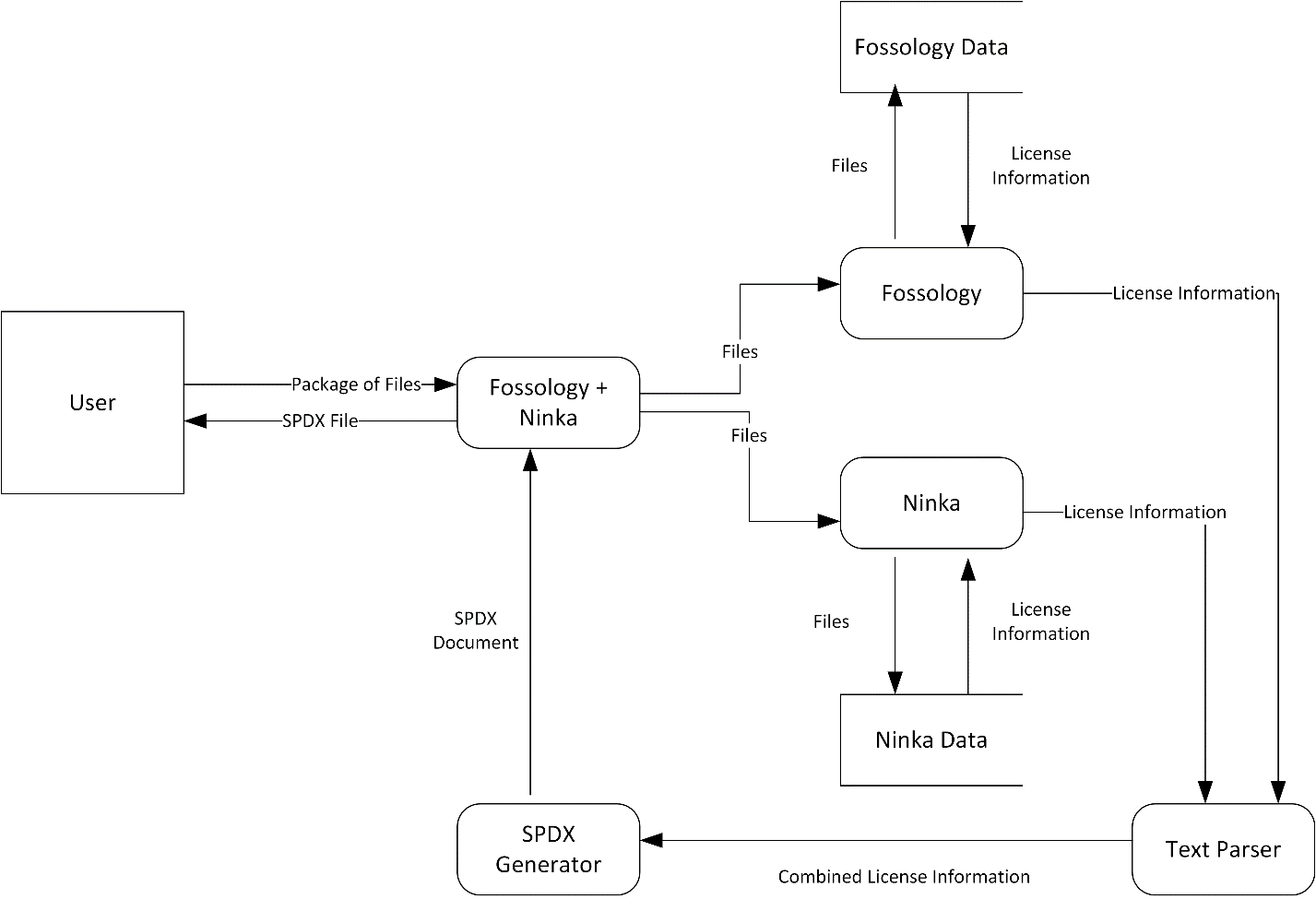
# Distribution System

The project will be distributed through a GitHub repository.

* **Name:** FOSSology-Ninka
* **Location:** [https://github.com/TheFinks/FOSSology-Ninka](https://github.com/TheFinks/Fossology-Ninka)
* **Ownership:** The repository is public, and is owned by Doug Richardson, Jon von Kampen, and James Thompson.

Programs and services may additionally be sent by e-mail at the request of a stakeholder or otherwise interested party. This method of distribution is at the sole discretion of the team (Doug Richardson, Jon von Kampen, and James Thompson) and should not be considered the primary means of distribution.

# Data Flow Diagram

****

# System Structure

The system will, by default, have all the structures and databases associated with FOSSology and Ninka. Refer to the documentation of those systems for more information.

On top of the structures built in to and required for the use of FOSSology and Ninka, the system is planned to have the following elements:

1. Text file output from FOSSology and Ninka on the command line (if such output is not already supported) (M)[[1]](#footnote-1)
2. A script to run FOSSology and Ninka on a given piece of software. (S)
   1. FOSSology and Ninka will probably be run in sequence because of anticipated resource constraints. Ideally, the script would run the tools in parallel.
3. A reader that will parse the output files and created a combined middle-document indicating the licenses concluded for a given piece of software (M).
4. An SPDX generator that will read the combined file and output it to an SPDX document (S).
5. A web-interface to do a tandem-scan of FOSSology and Ninka, and return an SPDX page (W).

# Copyright Declarations and License Choice

## FOSSology

FOSSology is licensed under GPL v2 and LGPL v2.

## Ninka

Ninka is licensed under AGPL v3.

## Original Artifact Attribution

All documentation and software created by the team will be attributed to Doug Richardson, Jon von Kampen, and James Thompson.

## Original Documentation

All documentation (including this document) are/will be licensed under the Creative Commons Attribution 4.0 International License.

## Original Software

All original software created by our team for the purposes of this project will be either GPL v2 or LGPL v2, depending on whether or not it needs to access proprietary libraries. This will allow it to most closely conform to FOSSology and other related projects.

## Licensing Conflicts

GPLv2 and AGPL v3 (used by FOSSology and Ninka respectively) are incompatible licenses. We will seek advice on how to avoid license conflict while still managing to complete the system to the best of our ability. One possible solution is to license our software as “GPLv2/ LGPLv2 or any newer version,” as GPLv3 and AGPLv3 are compatible. This will be discussed at future meetings.

# Change Log

## Overall Document

|  |  |  |
| --- | --- | --- |
| Date: | Action | Who |
| 02/03/2014 | Created final report | James Thompson |
| 02/05/2014 | Proofread and formatted final report | Jon von Kampen |

## Project Charter

|  |  |  |
| --- | --- | --- |
| Date: | Action | Who |
| 01/27/2014 | Created Prototype | Doug Richardson |
| 01/29/2014 | Updated File content | Doug Richardson |
| 01/29/2014 | Uploaded to Google Drive | James Thompson |
| 02/02/2014 | Updated License declaration | Doug Richardson |
| 02/03/2014 | Updated and moved into final report | James Thompson |

## System Service Request

|  |  |  |
| --- | --- | --- |
| Date: | Action | Who |
| 01/27/2014 | Created Prototype | Doug Richardson |
| 01/29/2014 | Uploaded to Google Docs | James Thompson |
| 02/03/2014 | Updated License Notice and optimal environment requirements | Doug Richardson |
| 02/03/2014 | Moved into final report | James Thompson |

## Stakeholders

|  |  |  |
| --- | --- | --- |
| Date: | Action | Who |
| 01/27/2014 | Created Prototype | Doug Richardson |
| 02/02/2014 | Uploaded to google drive, updated file content. Updated License Declaration | Doug Richardson |
| 02/03/2014 | Moved into final report | James Thompson |

## Communication Management Plan

|  |  |  |
| --- | --- | --- |
| Date: | Action | Who |
| 01/27/2014 | Created Prototype | Doug Richardson |
| 01/29/2014 | Uploaded it to Google Docs | James Thompson |
| 02/02/2014 | Updated content to include FOSSOLOGY and Ninka commnuity. | Doug Richardson |
| 02/03/2014 | Moved into final report | James Thompson |

## Distribution System

|  |  |  |
| --- | --- | --- |
| Date: | Action | Who |
| 01/27/2014 | Created Prototype | Doug Richardson |
| 02/02/2014 | Uploaded to google drive, updated file content.  Updated License Declaration | Doug Richardson |
| 02/03/2014 | Moved into final report | James Thompson |

## Data Flow Diagram

|  |  |  |
| --- | --- | --- |
| Date: | Action | Who |
| 01/27/2014 | Created Prototype | Doug Richardson |
| 01/31/2014 | Created MS-Paint Prototype of the dataflow diagram | Doug Richardson |
| 01/31/2014 | Created VISIO document of the dataflow diagram | James Thompson |
| 02/04/2014 | Moved into final report | Jon von Kampen |

## System Structure

|  |  |  |
| --- | --- | --- |
| Date: | Action | Who |
| 01/27/2014 | Created Prototype | Doug Richardson |
| 02/02/2014 | Uploaded to google drive, updated file content. Updated License Declaration | Doug Richardson |
| 02/03/2014 | Moved into final report | James Thompson |

## Copyright and License Declarations

|  |  |  |
| --- | --- | --- |
| Date: | Action | Who |
| 01/27/2014 | Created Prototype | Doug Richardson |
| 02/02/2014 | Uploaded to google drive.  Included section on license conflicts and possible solutions.  Updated document license notice. | Doug Richardson |
| 02/03/2014 | Moved into final report | James Thompson |

1. Implementation of each element is prioritized according to the MoSCoW model: (M)ust-have, (S)hould-have, (C)could-have, and (W)ould-have. [↑](#footnote-ref-1)