

Factores que inciden en el retraso de tiempos
de lanzamiento de proyectos comunitarios de
software libre mediante el uso análisis de redes
sociales: Caso practico Debian GNU/Linux
Protocolo

Wences Arana

14 de febrero de 2016

Índice general

Propósito del proyecto	2
Introducción	4
Antecedentes	6
Justificación	7
Marco Teórico	9
Planteamiento del problema	11
Objetivos	12
Metodología de investigación	14
Recursos	15
Resultados esperados	16
Asesor sugerido	17
Cronograma de trabajo	18

Propósito del proyecto

Los proyectos de software se definen como procesos de transformación de ideas hacia herramientas tecnológicas que resuelven necesidades de los usuarios. Y para llevar a cabo estos proyectos, las metodologías de gestión de proyectos sugieren la implementación de procesos ordenados con límites y objetivos definidos, con el fin de obtener ciclos de desarrollo y lanzamientos de Software eficientes y predecibles. No obstante, en la práctica proyectos de Software Libre han demostrado que la implementación de estos procesos no es una tarea trivial, dada su naturaleza orgánica y colaborativa.

Considerando que la medición de los factores que influyen en el atraso del cumplimiento de metas es de vital importancia para el área de Ingeniería de Software, y dada la inexistencia de un marco de referencia acerca de estos factores en el área de de Software Libre, se hace evidente la necesidad de determinar la existencia (si la hubiera) de factores que afectan los ciclos de lanzamiento.

Divisando la resolución de este problema, se propone el estudio de un proyecto de software existente, específicamente sistemas operativos libres desarrollados de manera colaborativa por voluntarios y otros actores interesados, siendo el elegido el proyecto Debian.

El proyecto Debian tiene como una de sus metas lanzar una nueva versión estable del sistema operativo en el momento en que ciertas metas se cumplen. Sin embargo, muchas de estas no se logran en un tiempo lo suficientemente rápido, teniendo como consecuencia la distribución de paquetes software más antiguos en relación a otras distribuciones GNU/Linux.

Poder predecir los tiempos de entrega, ayudará a proyectos grandes de software en la economización de recursos, debido a la detección a tiempo de cuellos de botella en desarrollo, colaboradores críticos en el proyecto,

mala comunicación entre los distintos equipos, entre otros. Ayudando así en la gestión del proyecto con la toma de decisiones relacionadas con los componentes que forman parte de su proyecto de software tanto lo social como lo técnico, lo tangible e intangible.

Introducción

En el área de administración de software una de las principales áreas de estudio es la entrega de lanzamientos en tiempo utilizando los recursos disponibles, si el proyecto de software no es entregado en el tiempo estipulado impactará directamente en el presupuesto del proyecto, la confianza en el equipo de desarrollo, y afectará la calidad en el entregable a utilizar. Poder predecir a tiempo los puntos que puedan atrasar el proyecto ayuda a entregar software sin impactar negativamente en los puntos mencionados anteriormente. En el caso del proyecto Debian GNU/Linux dicha problemática se ha visto evidenciada desde los inicios del proyecto, a pesar de no tener un esquema de tiempo estrictamente definido, se intenta lanzar una nueva versión estable del sistema operativo en el momento en que ciertas metas se cumplan, muchas de estas no se llegan a cumplir en un tiempo lo suficientemente rápido, causando que se distribuya software bastante estable pero más antiguo del que se presenta en otras distribuciones GNU/Linux.

Los proyectos de desarrollo de software libre tienen un componente humano colaborativo, el cual es construido a través de las aportaciones de código de cada uno de los desarrolladores, utilizando medios de comunicación como listas de correo y sistemas de registro de errores. Como puede notarse dichas relaciones dilucidan una red interna de comunicación entre cada uno de los desarrolladores, dichas redes puede ser estudiadas por medio del uso de técnicas provenientes de la teoría del análisis de redes sociales, pudiendo utilizar algoritmos y análisis de información que dichas teorías proveen para modelar de una mejor forma la red y poder clasificar la información determinando variables y factores que pueden apoyar nuestro estudio, unido a esto también podemos considerar el usos de teoría de redes sociales debido a que tiene como objetivo encontrar relaciones en eventos del proyecto que no parecen tener

incidencia unos con otros a primera vista, sin embargo al realizar un análisis profundo de estas, en algunos casos, pueden encontrarse desde relaciones de causa efecto entre eventos, actores, recursos, hasta la importancia de algunos de estos en el funcionamiento de todo el sistema en su conjunto, subgrupos relacionados, relaciones inesperadas, entre otras.

Utilizando dicha teoría se pretende analizar el proyecto colaborativo de software libre Debian y su sistema operativo Debian GNU/Linux con el objetivo de encontrar los factores que inciden en el tiempo de lanzamiento de versiones estables de este. Para lograr este objetivo se manejarán varios enfoques para medir dicho impacto y podemos definirlos como: utilizar como principales nodos la unidad mínima funcional de software en el proyecto denominada paquete, la figura de desarrollador de software siendo en este caso tanto un desarrollador Debian como un desarrollador original del software que da origen al paquete, como relaciones entre nodos se tomarán en cuenta, dependencias de construcción entre paquetes, dependencias en tiempo de ejecución de los paquetes, bugs detectados en los paquetes que interfieran en la relación entre ellos, cantidad de bugs reportados de parte de un desarrollador Debian a un desarrollador del software que da origen al paquete, cantidad de parches aplicados por parte de los desarrolladores originales del software enviados por un desarrollador Debian, cantidad de correos entre desarrolladores. Con esta información modelada se podrá analizar por medio de algoritmos existentes de teoría de análisis de redes sociales, en caso de poder ser aplicables, de lo contrario se creará el adecuado para poder realizar de manera apropiada los datos y de esta forma poder encontrar, patrones de centralidad, incidencia de clanes, nodos de mayor importancia, etcétera, y pudiéndose interpretar los valores y hacer un análisis del impacto en el tiempo de desarrollo de una nueva versión estable del proyecto de software a otra. Se utilizará como metodología de investigación por medio de un diseño no experimental *ex post facto* de tipo descriptivo, debido a que se hará un análisis de información de hechos y no se experimentará directamente con ellos.

Antecedentes

Los proyectos de software libre tienen por lo general estructuras colaborativas comunitarias[3], ,se han realizado estudios acerca de los patrones de comunicación, debido a que investigar las estructuras sociales es una forma útil de entender las practicas de los equipos realizándolo por medio de realizar un análisis de comunicación entre desarrolladores en un proyecto de software utilizando análisis de redes sociales de los correos entre los colaboradores que resuelven bugs[1]. Es importante además de que forma se transmite el conocimiento entre todos los miembros del grupo se ha podido determinar por medio de la utilización de la misma técnica que el conocimiento tácito del proyecto se lleva a cabo de persona a persona de manera directa o por un intermediario en posición dominante de tal forma que pueden determinarse flujos de comunicación entre personas[4].Otros autores han tocado el tema de la forma en que la comunicación fluye entre todo el equipo de desarrollo, además de mapearlo a los artefactos de software que estos programan [2]. Como se puede observar ya existen estudios que han abordado el tema utilizando la técnica de SNA aplicada a comunidades desarrolladoras de software, desde el punto de vista de comunicación, colaboración además de poder relacionarlas con el software en sí. Con estos estudios como base podremos modelar de una mejor forma nuestro estudio y siendo el elemento diferenciador, poder relacionar los resultados al impacto directo con el tiempo de lanzamiento del sistema operativo y relacionándolo con la antigüedad del software incluido dentro de este, y encontrando además los cuellos de botella y problemas específicamente para nuestro caso de estudio la distribución Debian GNU/Linux.

Justificación

En la actualidad proyectos colaborativos de software tienen problemas al tratar de poder entregar nuevas versiones estables de sus programas, debido a múltiples razones, tanto en la parte técnica de elección de herramientas y metodologías de desarrollo, así como en su parte social, respecto a la relación entre todos los desarrolladores que forman parte del proyecto. La investigación busca poder encontrar las principales causas de estos retrasos por medio de un análisis en el aspecto social respecto a la comunicación de los desarrolladores, y las relaciones existentes entre los paquetes de software que conforman a todo el sistema operativo que produce un proyecto colaborativo de software libre.

El estudio brindará una forma de poder medir por medio de lo anteriormente expuesto formas de poder encontrar factores que pueden retrasar el lanzamiento de un software, para poder detectarlos a tiempo y poder enfocarse en la solución de dichos inconvenientes.

Se ha decidido elegir un proyecto comunitario de software libre como Debian GNU/Linux debido a factores sociales y técnicos, sociales por que la estructura organizacional de la comunidad de desarrolladores es pública y abierta lo cual es una ventaja para poder obtener información de relaciones entre toda la comunidad. Desde el punto de vista técnico, al ser un proyecto que desarrolla un sistema operativo de software libre permite poder tener acceso a toda la información referente a los enlaces y relaciones que existen entre las unidades de software que forman al sistema operativo en general, desde relaciones de dependencias directas hasta relaciones indirectas en el funcionamiento integral del sistema operativo. Utilizando al proyecto Debian podremos describir los problemas específicos que tiene de tal forma que pueda servir al área de ingeniería de software.

El trabajo va dirigido a toda persona que necesite modelos para poder determinar que puntos modificar en su organización de manera técnica y social de tal forma que pueda mejorar tiempos de lanzamiento y obteniendo calidad en un tiempo razonable con una mayor anticipación.

Marco Teórico

1. Ingeniería de software

- a)* Definición
- b)* Recurso humano
- c)* Recurso de software
- d)* Etapas del proceso de desarrollo de software
- e)* Ciclo de vida de desarrollo
- f)* Gestión del proyecto
- g)* Aseguramiento de calidad
- h)* Tiempos de lanzamiento
- i)* Estimación de tiempos y metas

2. Proyectos colaborativos de software libre

- a)* Definición
- b)* Proyecto Debian
- c)* Estructura organizacional
- d)* Infraestructura técnica
- e)* Versionamiento del sistema operativo Debian GNU/Linux
- f)* Modelo de lanzamiento de distribución estable
- g)* Problemas relacionados a lanzamientos de versiones estables

3. Análisis de redes sociales

- a)* Definición
- b)* Centralidad, cuellos de botella
- c)* Clicas, clusters y componentes
- d)* Triadas

4. Caso de estudio

- a)* Clasificación de información
- b)* Selección de información
- c)* Métodos de recolección de datos
- d)* Recolección de información
- e)* Métodos de análisis

5. Análisis de resultado

- a)* Técnicas de presentación de información
- b)* Gráficas
- c)* Resultados
- d)* Notas Finales

Planteamiento del problema

Alcances

- Modelar una red de paquetes de software, relacionados por medio de las dependencias entre ellos, para analizar la importancia de cada uno
- Modelar una red de colaboradores dentro del proyecto (desarrolladores Debian) y entre los colaboradores externos e internos (Desarrolladores Debian y desarrolladores originales de software)
- Analizar las redes que pudieron determinarse y aplicar algoritmos de análisis de redes sociales
- Analizar los datos y encontrar los paquetes con más errores críticos para lanzamiento (release critical) y determinar si dichos paquetes tienen un grado alto, en la red y si esto afecta en los tiempos de lanzamiento de la distribución
- Analizar los datos referentes a las redes de colaboradores en el proyecto para determinar los colaboradores con más conexiones

Límites

- Se utilizarán los algoritmos existentes de teoría de redes sociales para encontrar las características importantes de las redes, no se pretenderá en este estudio crear nuevos algoritmos de análisis

Objetivos

Objetivo General

- Determinar los principales factores sociales y técnicos que influyen en el retraso de los ciclos de lanzamiento en un proyecto de software colaborativo, utilizando la teoría de análisis de redes sociales.

Objetivos Específicos

- Analizar el grado de la interrelación entre los desarrolladores de la distribución Debian GNU/Linux para determinar los principales colaboradores del proyecto tomando como medida la cantidad de código realizado y errores detectados y/o corregidos.
- Evaluar la importancia de los paquetes de software por medio de su grado en la red de paquetes en cuanto a sus dependencias y el impacto de tener un error crítico.
- Enumerar los paquetes de la distribución respecto a su número de dependencias y su posibilidad de retrasar el lanzamiento.
- Determinar los desarrolladores más activos dentro del proyecto, tomando como medida la cantidad de código aportado, bugs reportados, participación en lista de correo y mayor relación con otros desarrolladores.
- Analizar el impacto de perder la colaboración de desarrolladores activos dentro del proyecto en el tiempo de lanzamiento de la distribución.

- Examinar si la comodidad de las relaciones de colaboración entre los distintos desarrolladores que participan en la construcción de la distribución, tanto internos como externos impacta en el tiempo de lanzamiento de la distribución.
- Analizar el efecto de un bug RC en distintos paquetes y determinar si existe una correlación entre un paquete importante de la distribución con este tipo de bugs y el tiempo de lanzamiento de la distribución.

Metodología de investigación

La investigación será de tipo cuantitativo, por medio de un diseño no experimental ex post facto de tipo declarativo. Se decidió utilizar este enfoque debido a que se hará una recolección de datos de sucesos del pasado (correos electrónicos de colaboradores de la distribución, paquetes de software que conformaban a la distribución en un tiempo determinado del pasado, cantidad de errores que poseía un paquete en un periodo de tiempo determinado en el pasado). Por lo tanto no se puede realizar un experimento debido a que los datos son de hechos que ya sucedieron antes. Se opta por el modelo descriptivo y de análisis de relaciones de influencia.

Para el análisis de los datos ex post facto se utilizará la metodología de análisis de redes sociales, por la estructura de la información que puede ser agrupada en forma de redes y aplicando algoritmos de dicha metodología facilitará la elaboración de factores de correlación, que se acoplan a los buscados por nuestra investigación.

Se buscara describir dichos factores de correlación en el desarrollo de la distribución Debian GNU/Linux versión jessie en su periodo de desarrollo en el periodo de años 2013-2015.

Recursos

- Base de datos de errores del Proyecto Debian (Debian UDD)
- Base de datos de los paquetes del Proyecto Debian
- Base de datos de la lista de correo de errores y de los desarrolladores
- Computadora con el poder de procesamiento para analizar más de 30,000 paquetes interrelacionados
- Acceso a información del proyecto
- Investigador
- Asesor
- Al menos 2 desarrolladores Debian

Resultados esperados

Poder determinar la existencia de redes entre los distintos participantes de un proyecto colaborativo de software, desde el punto de vista social, es decir, la parte humana del proyecto (desarrolladores) y la parte técnica conformada por los paquetes. De existir las redes poder utilizarlas para analizar y encontrar y validar los posibles factores que afectan en el lanzamiento de una nueva versión estable. De tal forma que se esperara obtener:

- Mapear una red de paquetes unidos por sus relaciones de dependencias
- Mapear una red de colaboradores dentro del proyecto
- Obtener en ambas redes valores de centralidad
- Por medio de los datos obtenidos a partir del análisis de las redes, si es posible definir valores de centralidad, existencia de clusters, clicas, entre otros.
- Determinar importancia de los paquetes por medio de análisis de centralidad
- Determinar paquetes críticos en base a su cantidad de errores e importancia en la red
- Determinar colaboradores mas activos en cuanto a errores reportados y corregidos
- Determinar colaboradores mas activos respecto a su comunicación a través de los medios de comunicación del proyecto

Asesor sugerido

Ing. Msc. Víctor Leonel Orozco López

Cronograma de trabajo

Bibliografía

- [1] Kevin Crowston and James Howison. The social structure of free and open source software development. *First Monday*, 10(2), 2005.
- [2] Kate Ehrlich, Giuseppe Valetto, and Mary Helander. Seeing inside: Using social network analysis to understand patterns of collaboration and coordination in global software teams. In *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pages 297–298. IEEE, 2007.
- [3] Eric Raymond. *The cathedral & the bazaar musings on Linux and open source by an accidental revolutionary*. O'Reilly, Beijing, 1999.
- [4] Sharon Ryan and Rory V O'Connor. Development of a team measure for tacit knowledge in software development teams. *Journal of Systems and Software*, 82(2):229–240, 2009.