

Análisis de tiempos de lanzamiento de proyectos
comunitarios de software libre mediante análisis de
redes sociales: Caso practico Debian GNU/Linux
Protocolo

Wences Arana

8 de diciembre de 2015

Índice general

Propósito del proyecto

Todo proyecto de software inicia como una idea general hasta llegar a ser un sistema íntegro y funcional, para lograrlo conlleva un proceso ordenado con límites definidos en cuestiones de tiempo y cumplimiento de metas. Dicho cumplimiento de metas debe ser dentro de un marco tolerable de tiempo, esto con el fin de tener un proceso eficiente de desarrollo de software, un mejor manejo de costos de producción, calidad del mismo entre otros. Aunque varios proyectos de software libre como en el caso del Proyecto Debian no cuentan con dicha rigidez de tiempo si se desea que el software obtenido para lanzar de manera estable deba contar con un estándar de calidad alto.

Por esto mismo la medición de los factores que influyen en el atraso del cumplimiento de metas se hace de vital importancia para el área de la ingeniería de software teniendo como finalidad lograr crear un proceso eficiente de desarrollo replicable y dentro de los recursos disponibles o proyectados. Debemos por lo tanto plantear la realización de un análisis por medio de un estudio de los datos disponibles y crear un modelo replicable en proyectos de software con características similares, en nuestro caso se hará el estudio en base a sistemas operativos libres desarrollados de manera colaborativa por voluntarios y otros actores interesados, tomando en cuenta factores relacionados con la unidad mínima de software denominada paquete que conforma una red de estos que a su vez conforman el sistema operativo en su totalidad, además de las redes que existen en las relaciones sociales de los participantes en el desarrollo de dicho software. Se utilizará como base de este estudio a el Proyecto Debian debido a que provee esta infraestructura, además de tener actualmente algunos problemas de retraso en entregas de lanzamientos de su rama estable, dándonos el escenario ideal para realizar el estudio.

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 y 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 confianza en el entregable a utilizar. Poder predecir a tiempo los puntos que puedan atrasar el proyecto ayudaría a entregar el software sin impactar negativamente en los puntos mencionados anteriormente. En el caso del proyecto Debian GNU/Linux dicha problemática se ha visto evidenciada a través del tiempo, 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 lograr en un tiempo lo suficientemente rápido, de tal forma que se distribuya software más antiguo del que se presenta en otras distribuciones GNU/Linux.

Para poder llegar a determinar las causas que afectan en el lanzamiento de tiempo se decide utilizar la teoría de análisis de redes, esto debido a que tiene como objetivo encontrar relaciones en eventos de la naturaleza que no parecen tener incidencia unos con otros a primera vista, sin embargo al realizar un análisis profundo de sus relaciones, 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 upstream, cantidad de parches aplicados por parte de upstream 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 la propuesta por el muestreo teórico o “Grounded Theory”, bastante utilizada cuando se realizan estudios sociales, como es en teste caso además de ser un bien definido de actores conocidos.

Antecedentes

Los proyectos de software libre tienen por lo general estructuras colaborativas comunitarias, 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 bugs¹. 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 ². 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 ³. 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.

¹Crowston, Kevin, and James Howison. "The social structure of free and open source software development." First Monday 10.2 (2005).

²Ryan, Sharon, and Rory V. O'Connor. "Development of a team measure for tacit knowledge in software development teams." Journal of Systems and Software 82.2 (2009): 229-240.

³Ehrlich, Kate, Giuseppe Valetto, and Mary Helander. "Seeing inside: Using social network analysis to understand patterns of collaboration and coordination in global software teams." Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on. IEEE, 2007.

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