



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Proyecto ShoggothViz

Metodología para la visualización evolutiva de código fuente de proyectos hospedados en repositorios abiertos de software

Desarrollado por: Joan Sebastian Lopez Riaño

Institución: Universidad Nacional de Colombia – Sede Bogotá

Programa: Maestría en Ingeniería de Sistemas y Computación – Modalidad en profundización

Fecha: Marzo de 2017

METODOLOGÍA

1. Estrategia General

Para la efectiva implementación del proyecto se optará por una metodología del tipo experimental, es decir, que se validará la propuesta mediante pruebas de control (experimentos) y comparaciones respecto a los antecedentes encontrados en el estado del arte. Para llevar a cabo lo definido en el objetivo principal (desarrollar e implementar una metodología para la visualización de métricas aplicadas al software) se definirán un conjunto de por lo menos tres propuestas de metáforas figurativas las cuales serán validadas mediante la construcción de una aplicación de software por cada una. Finalmente se espera que con los productos de cada propuesta (visualizaciones) poder realizar un estudio comparativo respecto a otras metáforas de visualización encontradas en la literatura, así como también se contempla la realización de un experimento controlado con la metáfora definitiva.

Los cuatro objetivos específicos se repartirán en tres fases de evolución del proyecto, las cuales se pueden resumir como : Definición de la metáfora figurativa de visualización, implementación de la metodología y finalmente validación de la misma.

2. Fases del proyecto

Fase I: Definición de la metáfora y las métricas de la visualización

Esta fase inicial abarca los objetivos específicos No.1 y No.2. Representa el esfuerzo inicial por comprender el estado del arte en el tema de visualizaciones figurativas de software, el estudio de la evolución del software y las distintas métricas a analizar, clasificándolas según su relevancia y/o utilidad en las etapas del desarrollo de un proyecto de código abierto. Después de llegar a unas conclusiones respecto al estado actual del campo de estudio se planea explorar por lo menos tres propuestas para la metáfora final a seleccionar, escogiendo aquella que cuente con las bases mas solidas que justifiquen su usabilidad y pertinencia. De igual manera se definirá el conjunto de métricas a extraer tanto de los artefactos de código fuente como de los repositorios abiertos de un conjunto delimitado de proyectos *open source*.

Esta fase será la mas importante pues con la construcción de una muy buena justificación para la metáfora definitiva se podrán obtener mejores resultados de los posteriores experimentos controlados.

Fase II: Implementación de la metodología en proyectos de muestra

Después definir la metáfora y las métricas que comprenderán la propuesta, se procederá a la implementación de dicha metodología mediante la construcción de una aplicación que tenga como

objetivo generar las visualizaciones sobre los artefactos de código de un proyecto *open source* y su respectivo repositorio o plataforma colaborativa.

Para la realización del prototipo se utilizara la metodología de desarrollo SCRUM simplificada, la cual, aplicada a un equipo unipersonal con *sprints* o iteraciones semanales permitirá agilizar la generación cada semana de nuevas visualizaciones que se acerquen mas al producto deseado.

Fase III: Validación de la metodología

Para esta fase final se planea realizar un análisis experimental que tenga como objetivo sustentar la justificación de la metáfora en comparación con otras propuestas encontradas en el estado del arte, explorando sus posibles aplicaciones y el aspecto de las distintas etapas de la construcción de un proyecto de software para las cuales para la cual pueda ser mas relevante.

Se contempla la realización de un experimento controlado para retro-alimentar la propuesta y que permita validar las principales características de la visualización asi como sus potenciales debilidades.

Fase IV (Final):

Esta fase se considera la etapa de divulgación pues contempla la publicación de un artículo científico que resuma el proceso de la desarrollo de la propuesta , así como los resultados obtenidos del experimento controlado y que exponga algunos ejemplos de la visualización aplicada.

3. Actividades por fase

A continuación se presentan las actividades y productos por cada objetivo específico:

Fase I: Definición de la metáfora y las métricas de la visualización	
OBJETIVO	<i>Investigar el estado del arte de trabajos relacionados con las metáforas figurativas de código fuente y sus relaciones con los avances en el mejoramiento de la calidad del software.</i>
ACTIVIDADES	<ol style="list-style-type: none">1. Definir un marco teórico mínimo para las temáticas de visualización de software.2. Realizar un filtrado inicial de referencias recientes en el campo de la visualización de software3. Realizar un segundo filtrado de referencias para seleccionar los proyectos relacionados con metáforas del código fuente4. Elaborar las conclusiones del estado del arte
PRODUCTOS	<ol style="list-style-type: none">1. Documento del marco teórico y estado del arte del proyecto
OBJETIVO	<i>Definir la metáfora a implementar y el conjunto de métricas a extraer de los artefactos de código fuente hospedados en repositorios abiertos o plataformas colaborativas.</i>
ACTIVIDADES	<ol style="list-style-type: none">1. Escoger y clasificar un conjunto inicial de métricas de software de acuerdo a las conclusiones del estado del arte.2. Escoger y clasificar un conjunto inicial de métricas de los repositorios de código abierto de acuerdo a las conclusiones del estado del arte.3. Proponer y desarrollar por lo menos tres ideas para la metáfora de representación de las métricas anteriormente seleccionadas.4. Diseñar ilustraciones de ejemplo sobre las visualizaciones propuestas aplicadas a métricas extraídas de algunos artefactos de código fuente.
PRODUCTOS	<ol style="list-style-type: none">1. Documento de la descripción de la metáfora a implementar.

	<ol style="list-style-type: none"> 2. Documento de la descripción de las otras posibles metáforas a implementar. 3. Presentación de la propuesta de la metáfora principal, utilidad y aplicaciones.
--	---

Fase II: Implementación de la metodología en proyectos de muestra	
OBJETIVO	<i>Desarrollar una aplicación de software que dados un conjunto de artefactos de código fuente (clases, scripts, etc) genere una visualización que represente las distintas métricas a analizar de dichos elementos.</i>
ACTIVIDADES	<ol style="list-style-type: none"> 1. Documentar los casos de uso y diagramas de flujo de la aplicación a desarrollar. 2. Escoger la tecnología a implementar en la propuesta de visualización y conexiones con los repositorios de código fuente. 3. Diseñar la arquitectura y el modelado de información para la generación de las visualizaciones. 4. Implementar el diseño (desarrollo de la aplicación).
PRODUCTOS	<ol style="list-style-type: none"> 1. Prototipo inicial de la aplicación que implemente la metodología propuesta.

Fase III: Validación de la metodología	
OBJETIVO	<i>Validar la aplicación desarrollada mediante el análisis de algunos proyectos tipo FLOSS, que servirán de casos de estudio, comparando la metodología con otras similares encontradas en el estado del arte y la realización de un experimento controlado</i>
ACTIVIDADES	<ol style="list-style-type: none"> 1. Seleccionar un grupo de tres proyectos como casos de estudio 2. Generar por cada proyecto una visualización (de ser posible) usando alguna de las metodologías cercanas (identificadas en el estado del arte). 3. Generar por cada proyecto una visualización usando la metodología propuesta. 4. Comparar los resultados obtenidos por las metodologías de referencia y la metodología propuesta. 5. Generar un listado de conclusiones respecto a la comparación del punto anterior.
PRODUCTOS	<ol style="list-style-type: none"> 1. Documento con la descripción del proceso de validación y de conclusiones respecto a posibles usos y aplicaciones de la metodología propuesta. 2. Documento que resuma las hipótesis y resultados del experimento controlado.

