

DIAGRAMMING IN SOFTWARE ENGINEERING

Álvaro García Miranda – UO289039

Marcos Machado Menéndez – UO238688

Óscar Abad López – UO258238

Introducción:

En este trabajo analizaremos algunos temas tratados en la entrevista concedida por Ashley Peacock al programa SE Radio sobre los diagramas en la ingeniería del software y su importancia.

Trataremos de exponer los temas tratados en la entrevista, dar el punto de vista de Ashley Peacock como persona importante del sector y con muchos años de experiencia, y aportar información general para contrastar el punto de vista de Ashley Peacock.

Introducción a los diagramas:

Para transmitir la información de forma eficaz se creó UML (Siglas para Lenguaje unificado de modelado), un estándar que marca ciertas reglas a la hora de crear un diagrama.

Dependiendo de la información que se quiera representar, podemos emplear un diagrama u otro ya que UML nos proporciona diversos diagramas, cada uno con sus reglas y restricciones. Algunos de los más importantes para Ashley Peacock son: el diagrama de clases, el diagrama de estados, el diagrama de secuencia y el diagrama de objetos.

Ashley Peacock también destaca la importancia de la estandarización en los diagramas para representar información que sea fácil de entender, pero considera que a veces dicha estandarización puede ser perjudicial ya que se pierde de vista el objetivo principal que es representar información de forma sencilla. Y según él a veces la estandarización solo complica el entendimiento de un diagrama. Es importante distinguir cuando debemos priorizar una u otra cosa.

Herramientas para creación de diagramas:

A la hora de crear nuestros diagramas tenemos diversas opciones, podemos crearlos mediante herramientas manuales, ya sea a mano, o con herramientas digitales como draw.io. También podemos crearlos

mediante herramientas automáticas como PlantUML o Mermaid. Ambos métodos tienen su utilidad, y veremos sus ventajas y sus desventajas.

Las herramientas manuales son una pesadilla de mantenibilidad ya que se puede producir inconsistencia entre un mismo diagrama almacenado en local y en la nube. Las herramientas automáticas solucionan dicho problema al implementar los diagramas desde archivos de código que se cargan directamente mediante lenguaje Markdown.

PlantUml está basado en Java y aunque es muy útil tiene el problema de que los usuarios que no tengan Java en sus dispositivos no podrán utilizarlo. Es la principal ventaja de Mermaid sobre PlantUML, que Mermaid está basado en JavaScript y se puede ejecutar en la web. Además, GitHub lo soporta de forma nativa.

Modelo C4:

El modelo C4, creado por Simon Brown, propone dividir la documentación de los diagramas de un proyecto en una jerarquía de 4 niveles. Los distintos niveles representan la profundidad y complejidad de la información que se pretende representar. El primer nivel es el diagrama de contexto (cuyo objetivo es que cualquier persona pueda entender el funcionamiento de la aplicación de forma superficial), el segundo es el de contenedores, el tercero es el de componentes y por último el código (el nivel más complejo de entender y solo para expertos en la materia y más concretamente en el funcionamiento de la propia aplicación).

Ashley Peacock destaca la utilidad de los dos primeros niveles del modelo C4 a la vez que descarta emplear los dos últimos ya que los considera demasiado complejos.

Whiteboarding:

Es una técnica más informal de crear diagramas, similar al brainstorming. Consiste en que un grupo plantee ideas sobre la información a representar en una pizarra común. Genera diagramas menos mantenibles, aunque puede ser una buena idea emplear esta técnica si pretendemos crear diagramas short-lived.

Co-Generación y Diagramas a partir de Código:

La co-generación consiste en crear código a partir de diagramas, es una técnica que no está muy desarrollada por el momento, ya que crea un esqueleto básico sin funcionalidad.

De forma inversa, muchos IDES permiten, ya sea de forma nativa o mediante plugins, crear diagramas de clases a partir de código ya escrito. Los IDES de bases de datos permiten crear diagramas Entidad-Relación.

Conclusión:

Los diagramas son una parte importante de la documentación de un proyecto. Nos permiten transmitir y entender información de forma mucho más compacta y rápida.

Son de gran utilidad tanto a la hora de planificar la creación de un proyecto nuevo, como para transmitir la funcionalidad de uno ya creado a nuevos integrantes o expresar su funcionamiento a usuarios finales del sistema u otros stakeholders.

Por otra parte, es importante emplear los diagramas correctos para transmitir la información, así como mantenerlos actualizados, un diagrama desactualizado puede suponer un peligro mayor que la inexistencia de dicho diagrama.