**Event Storming**

1. Introducción. \_

En el desarrollo de software existen diferentes técnicas para obtener los requisitos funcionales. Muchas veces cuesta recopilar toda esa información y plasmarla en un diagrama de estados, diagrama de flujos, diagramas UML, un simple listado que nos tocará refinar y volver a refinar día a día con los expertos de negocio, product owner, etc. Es costoso, sí, porque muchas veces no tenemos una imagen precisa y clara de cómo funciona un negocio.

Event storming intenta simplificar este proceso con diferentes mecanísmos que involucra a las personas adecuadas en el proceso para obtener de una forma visual el funcionamiento de un sistema de negocio.

1. ¿Qué es evento storming?

Event storming es una técnica para descubrir el comportamiento de un negocio o los requerimientos del mismo. Es decir, permite recopilar eventos que se producen, los actores involucrados, servicios de terceros implicados y comandos o acciones para ejecutar las reglas de negocio.

Lo más destacado es: event storming proporciona un entendimiento común y un lenguaje ubicuo. Es decir, todo problema lo podemos plasmar en diferentes esquemas o modelos y todos los miembros son capaces de entenderlo. En ningún momento entra en detalles técnicos, esa es la gran ventaja.

1. Integrantes

Es un ejercicio donde pueden participar todo el mundo, es decir, los stackeholders, expertos en negocio, programadores, arquitectos, expertos en UI, etc. Aunque es recomendable acotar el número de participantes entre 6 y 8 personas.

1. ¿Cuándo usar event storming?

Todos los sistemas tienen una complejidad diferente. Cuando la dificultad es muy alta, es cuando se recomienda aplicar esta técnica. Ademas funciona muy bien cuando desconocemos totalmente del tema u problema para asi poder adquirir un contexto mas amplico en un par de sesiones.

1. ¿Por qué usar event storming?

Por que event storming se apoya en el aprendizaje como grupo, fomentando la colaboración y la comunicación dentro del equipo de trabajo y no solo el enfocarse en la accesivilidad u la cordinación del equipo. Dando paso a participar, opinar, aportar ideas y el compartir las visiones diferentes, lo que nos lleva a una mejor comprención de los procesos, todo esto con el fin de llegar al mismo objetivo, el cual es satisfacer las nesecidades de un negocio.

**Procedimiento event storming**

• Domain event:

El primero paso es descubrir los eventos más importantes. Los eventos deben de empezar con un sustantivo + verbo.

• Command:

Después descubrimos los comandos o acciones, es decir, que acción es necesaria para lanzar dicho evento.

• Rol:

El siguiente paso es descubrir quien desemboca la acción, puede ser un usuario o un sistema externo.

• External system:

Ahora toca descurbir sistemas externos, por ejemplo, el envío de email se hace a través de un servicio que esta fuera de nuestro dominio.

• Business rule:

Analizar posibles reglas de negocio, en este ejemplo, el cliente marca el requirimiento de una cita automática cuando es creado.

• Aggregate:

Ahora toca representar los agregados, recueden, los agregados representa el estado de un sistema.

• View:

Podemos añadir énfasis visual en el alta de un cliente. Pero insisto, este paso no es tan importante.

• Organizamos los agregados en diferentes contextos. El objetivo es definir límites entre ellos. En resumidas cuentas es agrupar las responsabilidades de los agregados. Por ejemplo, en caso de Administration puede haber una gestión de usuarios y operadores, y en caso de Calendar una gestión de reservas, reuniones y tareas. Podemos ir añadiendo más y más agregados en base a las necesidades del negocio y cada uno estará clasificado en un contexto.

**RECOMENDACIONES DE EVENT STORMING**

1. **Involucra a las partes interesadas clave:**

Asegúrate de que las personas adecuadas estén presentes durante las sesiones de Event Storming. Esto incluye a desarrolladores, expertos en dominios y cualquier otra persona que tenga conocimiento valioso sobre el sistema.

1. **Define un objetivo claro:**

Antes de comenzar con la sesión de Event Storming, establece un objetivo claro. Pregúntate a ti mismo qué problema específico estás tratando de resolver o qué parte del sistema estás diseñando.

1. **Utiliza papeles y tarjetas de colores:**

Asigna colores a los diferentes tipos de eventos, comandos, agregados, etc. Esto hace que sea más fácil visualizar y entender el flujo de eventos durante la sesión.

1. **Itera y refina:**

Event Storming no es un proceso estático. Itera sobre los resultados, refina los eventos y ajusta el diseño según sea necesario. La retroalimentación de las partes interesadas es crucial.

1. **Fomenta la colaboración y la comunicación:**

Event Storming es una herramienta colaborativa. Asegúrate de que todos los participantes estén comprometidos y alienta la comunicación abierta. La idea es aprovechar la sabiduría colectiva del equipo.

1. **Incluye el tiempo en el modelo:**

Event Storming no solo trata sobre eventos instantáneos, sino también sobre eventos a lo largo del tiempo. Asegúrate de incluir el tiempo en el modelo para comprender cómo evoluciona el sistema con el tiempo.

1. **No te preocupes demasiado por la formalidad al principio:**

Al principio, no te preocupes demasiado por seguir todas las reglas de manera estricta. El objetivo principal es obtener una comprensión compartida del sistema. Puedes refinar y formalizar más tarde.

1. **Considera la escalabilidad y la tolerancia a fallos:**

Diseña tu sistema pensando en la escalabilidad y la tolerancia a fallos desde el principio. Event Storming puede ayudarte a identificar puntos críticos en tu diseño.

1. **Documenta el conocimiento adquirido:**

Event Storming es una herramienta de descubrimiento poderosa. Documenta el conocimiento adquirido durante las sesiones para que sea accesible para todos los miembros del equipo.

1. **Aplica principios de Event Sourcing:**

Si decides implementar Event Sourcing, asegúrate de comprender sus principios y cómo se integran con el diseño general de tu sistema.

**CONCLUSION**

Event Storming facilita la comprensión y diseño efectivo de sistemas complejos, permitiendo que los equipos tomen decisiones informadas sobre la arquitectura y el diseño de software. Es una herramienta valiosa para la colaboración y la toma de decisiones en el desarrollo de software.