¿Qué debería saber una Product Manager sobre Arquitectura de Software?

Perla Velasco-Elizondo

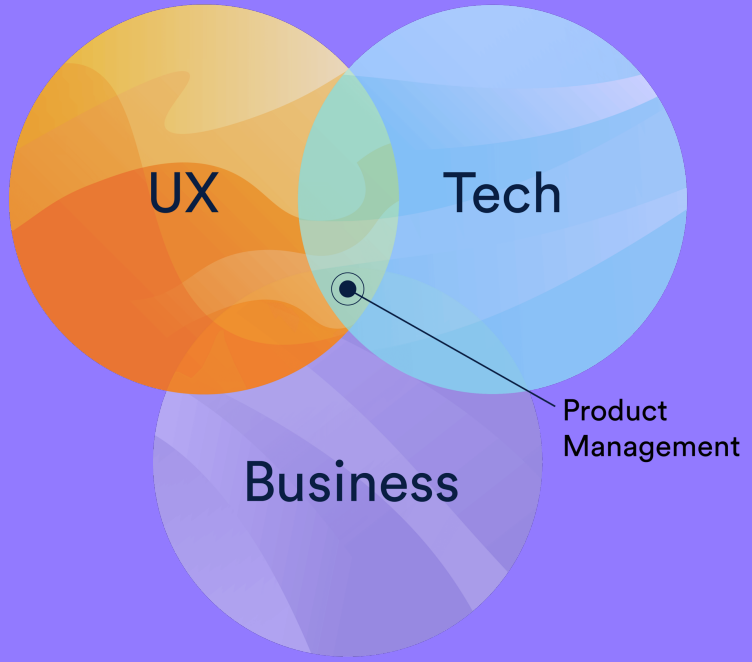
#Devday4w

Product Management

Tiene que ver con identificar las necesidades de los consumidores y satisfacerlas a través del desarrollo y entrega de productos.

Esto incluye la parte de investigación previa al desarrollo del producto, las estrategias y el plan de marketing posteriores al mismo.

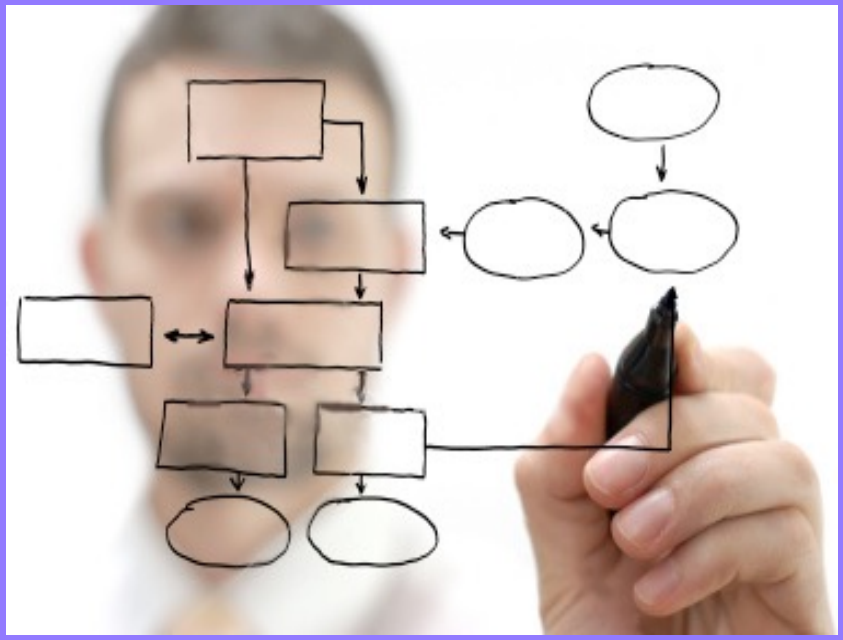
Estas actividades son realizadas por un rol denominado Product Manager (PM)



DEV DAY 4 Women

Software Architecture

Tiene que ver con la toma de decisiones sobre la estructuración general de un sistema, involucrando no solo sus requerimientos funcionales sino también requerimientos técnicos, de negocio y de calidad.



DEV DAY 4 Women

Una buena
arquitectura de software
es fundamental para
el éxito de un
producto de software

DEV DAY 4 Women

Atributos de Calidad

Son requerimientos que especifican atributos que pueden usarse para juzgar la operación de un producto de software, en lugar de sus comportamientos específicos, ej. disponibilidad, portabilidad, desempeño o seguridad o facilidad de prueba.

Una mala arquitectura de software puede inhibir el logro de los atributos de calidad.

DEV DAY 4 Women

¿Qué debe saber/entender una PM?

External quality	Brief description
Availability	The extent to which the system's services are available when and where they are needed
Installability	How easy it is to correctly install, uninstall, and reinstall the application
Integrity	The extent to which the system protects against data inaccuracy and loss
Interoperability	How easily the system can interconnect and exchange data with other systems or components
Performance	How quickly and predictably the system responds to user inputs or other events
Reliability	How long the system runs before experiencing a failure
Robustness	How well the system responds to unexpected operating conditions
Safety	How well the system protects against injury or damage
Security	How well the system protects against unauthorized access to the application and its data
Usability	How easy it is for people to learn, remember, and use the system

Internal quality	Brief description
Efficiency	How efficiently the system uses computer resources
Modifiability	How easy it is to maintain, change, enhance, and restructure the system
Portability	How easily the system can be made to work in other operating environments
Reusability	To what extent components can be used in other systems
Scalability	How easily the system can grow to handle more users, transactions, servers, or other extensions
Verifiability	How readily developers and testers can confirm that the software was implemented correctly

DEV DAY 4 Women

¿Qué debe saber/entender una PM?

1. Disponibilidad: ¿Pueden los usuarios acceder a nuestro producto en el momento en que lo necesitan?
2. Desempeño: ¿Pueden los usuarios realizar las operaciones de nuestro producto en los tiempos que lo necesitan?

3. Escalabilidad: ¿ Pueden nuestros ingenieros escalar nuestro producto a medida que crece nuestra base de usuarios?
4. Mantenibilidad: ¿Pueden nuestros ingenieros modificar fácilmente nuestro producto a medida que evoluciona la tecnología?
5. Trazabilidad: ¿ Pueden nuestros ingenieros tener un registro de las operaciones realizadas en nuestro producto?

DEV DAY 4 Women

¿Qué debe saber/entender una PM?

1. Cuáles son los atributos de calidad fundamentales satisfacer las necesidades de los usuarios de nuestro producto.

2. Cuales de esos atributos de calidad involucran un alto nivel de dificultad técnica.

DEV DAY 4 Women

Principios de Diseño de Software

Son una serie de reglas y recomendaciones específicas que los ingenieros de software deben seguir durante el diseño y desarrollo de un producto de software si quieren escribir un diseño e implementación limpia, comprensible y fácil de mantener.

Una mala arquitectura de software puede ser el resultado de no atender principios de diseño.

DEV DAY 4 Women

¿Qué debe saber/entender una PM?

1. **Cohesión** tiene que ver con cómo se organizan las distintos componentes de un sistema de software como las clases, servicios o subsistemas.
   Alta cohesión significa tener todos los datos y la funcionalidad estrechamente relacionados en un componente con un rol bien definido.

2. **Acoplamiento** tiene que ver con la interdependencia entre los distintas componente de un sistema de software.
   Bajo acoplamiento significa menos interdependencia entre los componentes. Los cambios realizados en un componente no deberían requerir cambios sustanciales en otros componente.

DEV DAY 4 Women

Patrones de Arquitectura de Software

Son estructuras de sistemas de software de probada eficacia en ciertos dominios de aplicación.

Estas estructuras contienen componentes de varios tipos, relaciones entre ellos y propiedades de ambos.

Una mala arquitectura de software puede ser el resultado de no usar patrones de arquitectura de software.

DEV DAY 4 Women

¿Qué debe saber/entender una PM?

1. Tener una comprensión básica de componentes clásicos patrones como cliente-servidor, capas, modelo-vista-controlador, microservicios, etc. No es necesario conocerlos todos, o tener un amplio conocimiento.

2. Entender la razones por qué se ha, o no elegido, un patrón arquitectónico para el desarrollo del producto de software

3. Si se ha elegido uno, entender las desventajas que tiene el uso del patrón arquitectónico elegido.

Deuda Técnica

Es el costo del retrabajo adicional que deben pagar los ingenieros cuando por elegir una solución fácil (limitada) ahora en lugar de usar un mejor enfoque de diseño y desarrollo que llevaría más tiempo.

DEV DAY 4 Women

¿Qué debe saber/entender una PM?

1. Costos de desarrollo. La velocidad de desarrollo se hace lenta.
2. Costos operativos. Tiempo perdido en la solución de problemas.
3. Costos de oportunidad. Tiempo perdido en lanzar una funcionalidad que podría haber mejorado el negocio.
4. Costos comerciales. Interrupciones que conducen a la pérdida de ingresos

¡Gracias!

perla@velasco-elizondo.net

DEV DAY 4 Women