**U2: Gestión de Producto**

**¿Por qué creamos productos?**

El proyecto es el medio elegido para organizar personas, recursos, tiempo. Sin embargo, lo que hacen los desarrolladores es un producto, no un proyecto.

Con el paso del tiempo comenzó a surgir la necesidad de incorporar al equipo miembros cercanos al negocio, que no necesariamente tengan conocimiento técnico.

Al hacer software, no estamos buscando solo dinero, sino que buscamos implementarlo, mantenerlo, que el software sea utilizado y le sirva al cliente.

Para saber si el mismo es utilizado, y le agrega valor al cliente utilizamos diversas métricas, que determinan si un producto es exitoso o no, como, por ejemplo: cantidad de usuarios que instalan el producto, o cantidad de usuarios que dejan instalado el producto.

**¿Qué funcionalidades se utilizan?**

Al empezar a crear un producto, no se trata al detalle todo lo que se quiere que el producto haga. Si vamos a trabajar con un ciclo iterativo de desarrollo, la priorización cobra importancia. La idea es priorizar lo que siempre se usa, pero esto no es fácil de determinar.

Aquel que sabe qué funcionalidades son las más utilizadas es el cliente, el usuario, por ello es importante la presencia del negocio a la hora de priorizar.

También existen casos en donde el cliente percibe que, si el ciclo es iterativo, la funcionalidad que no fue desarrollada en las primeras iteraciones no será desarrollada nunca.

Definición de SW: Es un artefacto que vamos a construir para satisfacer una necesidad. Es un entregable, es el resultado de un proyecto.

**El producto y su versión**

La versión del producto está conformada por: Código, Manuales (de Usuario o de Configuración/Parametrización) y por la Base de Datos.

Esto es como mínimo, ya que puede tener muchas más cosas. Si el producto es de llave abierta, se debe entregar también la documentación (de arquitectura, de diseño, casos de prueba, etc.)

Al vender un producto, todo el código fuente es del comprador (por ley), aunque en el contrato se haya pactado otra cosa. El justificativo de esto, es que el producto está basado en el negocio del cliente. Pero esta legislación presenta varios problemas y es que no considera que el código puede incluir librerías propias y/o de terceros (en este último caso, si alguna falla, debemos hacernos responsables por la misma).

Aunque también hoy en día existen SaaS, PaaS, IaaS que cambian la concepción anterior. Ya que si nuestro producto es un SaaS solo entregaríamos los manuales y el acceso al ejecutable y BD. Aunque esto también conlleva otro problema y son los datos. Esto se debe a que si bien la infraestructura es de los desarrolladores, los datos producidos y almacenados mediante el uso del producto son del cliente.

**Evolución del producto de software**

**Diagrama

Descripción generada automáticamente**Que el producto sea funcional, es lo mínimo que se puede hacer, y hoy en día es considerado como básico, pero se espera mucho más de un producto.

Que el mismo sea confiable, es casi tan básico como que sea funcional.

El software hoy en día se busca que sea usable, en lo referido a la UX. Es decir, el usuario tiene que ser productivo, eficiente y estar cómodo usando el software.

La mayoría de los productos de software se quedan en este último escalón, ya que es muy complicado atravesarlo. Este escalón o línea marca un hito que la mayoría de los productos de software no pueden cumplir. Para cruzar este escalón el producto debe ser algo que nos mejore la vida.

**Explicación de la evolución de un producto de software**

No es necesario que tengamos claro al 100% que esperamos que el producto haga. Uno empieza queriendo algo que sea único. Es decir, el producto nuevo tiene una hipótesis de valor único.

La empresa o starup empieza la idea tratando de encontrar un nicho, un UVP (Unique Viable Product). Una vez encontrado existe una gran duda, y es la forma de monetizar el mismo. Si elegimos vender, la pregunta es, ¿venderá el producto? Si elegimos que sea gratuito, la pregunta es ¿será utilizado?

Para saber esto, necesito algo barato, rápido, que me permita salir al mercado y validar si el producto tendrá éxito. Por ello es que se cera el MVP (Minimum viable product). Dicho MVP también es una hipótesis. Salimos a buscar clientes y estos nos devolverán un feedback.

Uno de los resultados posibles, es que cada cliente potencial con el que interactúas te dice que, si bien la idea es buena, para poder usarlo necesita la funcionalidad X, y ese “X” es distinto para cada cliente o usuario. Eso quiere decir que el producto no encontró su nicho, por lo que la hipótesis planteada por el MVP fracasó.

Otro de los resultados posibles, es que las respuestas de los clientes potenciales son que, para usarlo, necesitan la funcionalidad X, y ese “X” es el mismo para todos los clientes. En ese caso, tiene sentido revisar la hipótesis de Cliente/Problema/Solución.

En ese momento podemos crear un segundo MVP, un ajuste o un pivot. Se construye este segundo MVP centrado en la nueva hipótesis basada en el aprendizaje generado por el anterior MVP. Supongamos que el segundo MVP es exitoso y está viendo una tracción real de los primeros usuarios. Se desea aumentar el crecimiento y buscar una penetración más profunda de sus primeros usuarios, además de atraer nuevos clientes. Para ello se seguirá investigando y validando, aprovechando el feedback recopilado para profundizar en las áreas que atraigan dicho potencial crecimiento.

De esta manera obtenemos una versión más robusta, y que es una aproximación al release o entregable.

Cuando se está encaminado con un determinado MVP, puede que la investigación encuentre áreas que necesiten un enfoque o una característica en particular, y que dicha característica pueda salir al mercado. Esto se conoce como MMF (Minimum Marketable Feature), y el objetivo de la misma es aportar valor. Supone una alta certeza (baja incertidumbre) de que existe valor en esa área o enfoque y que sabemos cual debe ser el producto para proporcionar ese valor.

La razón para dividir una característica grande en MMF más pequeños es principalmente el tiempo de comercialización (time to market) y la capacidad de aportar valor en muchas áreas. Cada MMF se puede identificar como una US.

Supongamos ahora que se quiere incorporar una nueva característica al producto, en un área con alto potencial, pero también con alta incertidumbre. Para disminuir la incertidumbre, planteamos una hipótesis, formulando un MVF (Minimum Viable Feature), es la característica mínima que aún puede ser viable para uso real y aprendizaje de los usuarios.

Si la MVF resulta exitosa, puede desarrollar más MMF en esa área para tomar ventaja (si eso tiene sentido). Si no es así, puede cambiar a otro enfoque hacia esa área de características, o en algún momento buscar una ruta de crecimiento alternativa. Esencialmente el MVF es una versión mini del MVP

Modelo Completo: El producto se cultiva en mercados inciertos al intentar varios MVP. Cuando se logra ajustar el producto en el mercado de productos, se combinan MMF y MVF según el nivel de incertidumbre del negocio en las áreas en las que se está enfocando.

Si bien los MVP / MMF / MVF son atómicos desde una perspectiva empresarial (no puede implementar y aprender de algo más pequeño) pueden ser bastante grandes desde la perspectiva de la implementación.

* MVF: Extraemos una única característica del MVP y la sacamos al mercado para validarla.
* MRF (Minimum Release Feature): Características mínimas que tiene que tener una reléase para salir al mercado. Es la evolución del MVP validado y comercializable.

Básicamente el MVP y el MVF son hipótesis que se deben validar ante situaciones de incertidumbre. El MRF y el MMF son entregables y comercializables.

**Relaciones entre MVP, MMF, MMP y MMR**

* **MVP**: Versión de un nuevo producto, creado con el menor esfuerzo posible. Dirigido a un subconjunto de clientes potenciales y utilizado para obtener aprendizaje validado. Es más cercano a un prototipo que a una versión real del producto.
* **MMF**: Es la pieza más pequeña de funcionalidad que puede ser liberada. Tiene valor tanto para la organización como para los usuarios. Es parte de un MMR o MMP.
* **MMP:** Es el primer reléase de un producto o MMR dirigido a los primeros usuarios. Está focalizado en características clave que satisfarán a dicho grupo.
* **MMR:** Release de un producto que tiene el conjunto de características más pequeño posible. El incremento más pequeño que ofrece un valor nuevo a los usuarios y satisface sus necesidades actuales. MMP = MMR1

**Aprendizaje Validado:** Es un método riguroso para demostrar progreso dentro de un ambiente de alta incertidumbre. Es el proceso de demostrar empíricamente que el equipo ha descubierto verdades valiosas acerca de un producto o empresa. Es más preciso, concreto y rápido que cualquier tipo de planeamiento o predicción.

**Minimum Viable Product (MVP)**

Es un concepto de Lean Startup que enfatiza el impacto del aprendizaje en el desarrollo de nuevos productos. Una premisa clave detrás de la idea de MVP es que usted produce un producto real que puede ofrecer a los clientes observar su comportamiento real con el producto o servicio.

Definición: Versión de un producto que permite a un equipo recopilar la cantidad máxima de aprendizaje validado sobre clientes con el menor esfuerzo. Este aprendizaje validado viene en forma de si sus clientes realmente comprarán su producto.

Es la versión del producto que me permite un ciclo completo de build-experiment-learn con un mínimo esfuerzo. Carece de muchas funcionalidades que serán esenciales en un futuro.

Ver lo que la gente realmente hace con respecto a un producto es mucho más confiable que preguntarle a la gente que harían. Tiene el valor suficiente para que las personas estén dispuestas a usarlo o comprarlo inicialmente.

Necesitamos feedback de potenciales usuarios para saber si invertir o no. Eso es incompatible con la idea de hacer un producto entero. Demuestra suficiente beneficio futuro para retener a los primeros usuarios.

**Minimum Viable Feature (MVF)**

MVF es una versión mini del MVP. Es una característica a pequeña escala que se puede construir e implementar rápidamente, utilizando recursos mínimos, para una población objetivo para probar la utilidad y adopción de la característica.

Un MVF debe proporcionar un valor claro a los usuarios y ser fácil de usar. MVF requiere recursos mínimos, los estándares de calidad de la industria y la producción deben guiar el diseño y confiabilidad.

El grupo de usuarios para un MFV, son los primeros en adoptar, los clientes leales que han compartido conocimientos anteriormente o los miembros de una junta asesora de clientes. Es decir, usuarios flexibles y tolerantes.

**Errores Comunes**

* Confundir a un MVP, que se enfoca en el aprendizaje con un MMF o un MMP, ya que estos se enfocan en ganar(se comercializan). El riesgo de esto es entregar algo sin considerar si es lo correcto que satisface las necesidades del cliente.
* Enfatizar la parte mínima del MVP con exclusión de la parte viable. El producto entregado no es de calidad suficiente para proporcionar una evaluación precisa de si los clientes utilizarán el producto.
* Entregar lo que consideran un MVP, y luego no hacer más cambios a ese producto, independientemente de los comentarios que reciban al respecto.

**Valor vs Desperdicio**

Lean es un framework mucho mas viejo que agile, y posee 7 principios, dentro de los cuales el 1ero y el más importante es “Eliminar desperdicios”. Esto me permite gastar menos y ser más productivos.

Ahora la pregunta es ¿qué es desperdicio?, la respuesta es simple, todo aquello que no genera valor es desperdicio. A eso apuntamos con nuestra necesidad de validación, es la razón por la que queremos saber si nuestro producto les dará valor a nuestros clientes.

Todo aquello que no contribuye al aprendizaje de qué es de valor para el cliente, es desperdicio. Por eso es importante experimentar, que el cliente tenga contacto con el producto, porque podemos enterarnos de qué cosas le agregan valor, qué cosas le sirven, y asi nos evitamos esfuerzos en desarrollar cosas que no sabemos si le serán de utilidad o no.

La productividad de un startup no puede medirse en términos de cuánto se construye cada día, por el contrario, se debe medir la productividad en términos del valor que generamos al cliente por día.

**Build-Experiment-Learn Feedback Loop**

Diagrama

Descripción generada automáticamenteEl éxito no es entregar un producto, el éxito se trata de entregar un producto que el cliente usará.

La forma de hacerlo es alinear los esfuerzos continuamente hacia las necesidades reales de los clientes.

Este feedback loop permite descubrir las necesidades del cliente y alinearlas metodológicamente.

La actividad fundamental de una startup es convertir ideas en productos, medir o experimentar como responden los clientes, y aprender de esto para decidir si pivotar o perseverar.

El método Lean Startup se basa en este ciclo de feedback. Mediante este proceso, podemos aprender cuando es tiempo de realizar un cambio importante (pivotear) o si debemos perseverar en el camino actual. Sin embargo, el producto está en constante cambio o mejora, mediante un proceso de optimización que surge del aprendizaje validado, pivotear implica un cambio de estrategia.

Siempre debemos tener una idea clara de a dónde estamos yendo. Todas las startup tienen un norte claro, un destino, es la visión de la startup. La visión del startup no debe cambiar, a pesar de que su producto si lo haga.

La función más vital es aprender, debemos aprender qué elementos de nuestra estrategia nos permiten realizar la visión que perseguimos. Debemos aprender qué es lo que los clientes quieren, no lo que ellos dicen que quieren o piensan que quieren.

**La fase de construir el MVP**

La idea es ingresar al mercado lo más rápido posible con un MVP. Hay varios tipos de MVP, dependiendo de su complejidad y van desde las pruebas de humo (smoke test) extremadamente simples (el producto no existe, es un video, un anuncio) hasta prototipos tempranos (producto con características faltantes)

La idea es mantener la simplicidad, y eso se logra maximizando el trabajo no hecho.

No es fácil decidir qué tan complejo se debe hacer un MVP, no existe una fórmula:

* Requiere de juicio para cada caso en particular.
* Si existe duda, simplifique la idea.
* Evitar la construcción y la promesa excesivas.
* Cualquier trabajo adicional más allá de lo que se necesita para comenzar el ciclo es desperdicio.

Un MVP no solo habla del diseño de un producto y las preguntas técnicas, sino que sirve para probar hipótesis comerciales fundamentales.

Un MVP tiene dos posibles resultados, o es desechado o se continúa invirtiendo en el mismo.

**Dilema de la audacia cero**

Es más fácil recaudar dinero cuando se tiene cero ingresos, cero clientes y cero tracción. No se tiene nada que perder, por lo que se tiene mayor imaginación. Mientras menos tenemos que perder, estamos en mejor condición de crear un buen MVP.

Sin embargo, también hay algunos problemas, como la duda de que, si ganamos poco al principio, podamos llegar a ganar mucho al final. Es posible aplazar el lanzamiento de cualquier versión del producto hasta que esté seguro del éxito. Si se pospone experimentar con el MVP (lanzarlo) pueden surgir algunos resultados indeseados, como el aumento de la cantidad de trabajo desperdiciado, se pierde de feedback importante y se incrementa el riesgo de construir un producto que nadie utilice.

Es muy importante experimentar con el MVP, primeramente, en silencio, con los primeros usuarios en el mercado. Es importante probar todos los elementos del MVP, comenzando por los más riesgosos.

**Supuestos de “Saltos de Fe”**

Los elementos más riesgosos del plan o concepto de una startup se denominan supuestos de salto de fe. Pero confiamos en que es un problema que nosotros podemos resolver, y debemos estar dispuestos a asumir el riesgo. Las asunciones de valor y de crecimiento son las más importantes.

Para disminuir el riesgo y la incertidumbre debemos experimentar, y el primer paso es dividir la visión de la empresa en sus partes componentes. Si el producto realmente es de valor debe satisfacer las hipótesis de valor y de crecimiento

* Hipótesis de valor: Prueba si el producto es el correcto, es decir, si realmente está entregando valor a los clientes. Hay una métrica que se puede usar, que es la tasa de retención de usuarios.
* Hipótesis de crecimiento: Prueba cómo nuevos clientes descubrirán el producto. Como métricas de prueba se puede usar la tasa de referencia o la tasa NPS
  + Tasa NPS (Net Promoter Score): Es una fórmula que solo tiene en cuenta los extremos, aquellos que defienden y promueven el producto, y los detractores (los que no quieren que el producto tenga éxito).

Una vez que corroboramos la hipótesis de valor y de crecimiento, el primer paso es entrar a la fase de construcción lo más rápido posible con un MVP

**Preparar un MVP**

1. El primer paso y el más difícil es encontrar un nicho.
2. Crear un roadmap realista. Es una definición a alto nivel de las características del producto que tendremos listas en cada trimestre.
3. Investigar la competencia, qué cosas hace bien la competencia y qué cosas fallan o no tiene.
4. Pre-vender el MVP
5. Testear las suposiciones
6. Asegurarse que el MVP resuelve el problema correcto.
7. Focalizarse en las funcionalidades principales