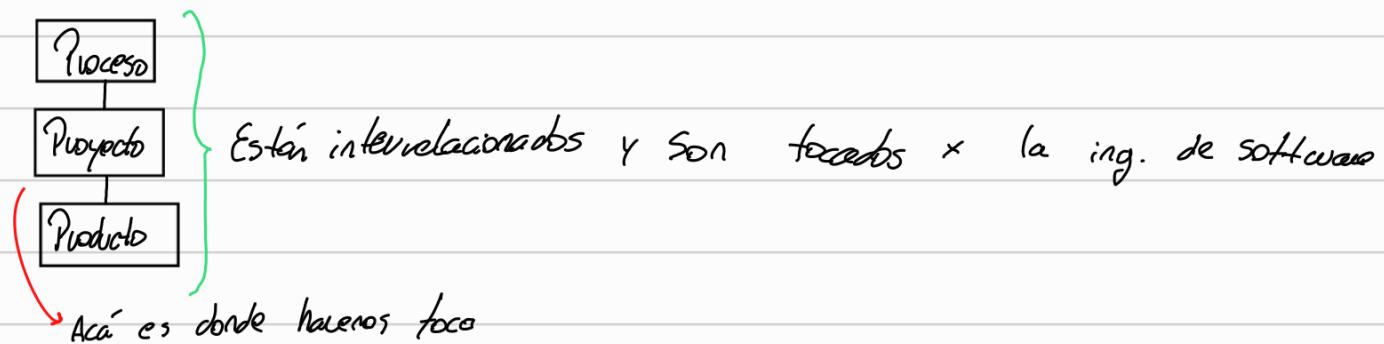


User Stories



¿Qué es un proyecto? → gabinete de usuarios y pasos, ejecución de un proceso con objetivos en un plazo. Para obtener un producto o servicio. Resultados únicos.

Hay una relación de control
entre las actividades y hace
que se cumplan

Procesos empíricos → Hay en día tan solo 2 tipos de procesos: definidos (ASI, DSI, como PDD y RUP) y empíricos (tengo q' saber q' cosas debería hacer y desarrollar software en base a la experiencia, buscando repetitividad y q' eso nos diga en q' estado del proyecto estamos y así evitamos compromisos).

Los pasos se van para llevar a cabo el proyecto, hasta el punto de q' se quiera agregar a las personas como una vía "P", el problema de la comunicación hace fallar los proyectos y no las limitaciones tecnológicas.

Quienes definen los procesos son distintos a los q' llevan a cabo los proyectos.

Para los procesos empíricos surge los cuadros ágil y lean, hoy están bastante unidos y mezclados por lo q' llamamos "cuadro lean-agil". NO son metodologías, son filosofías.

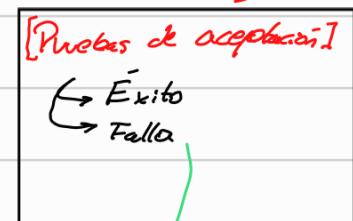
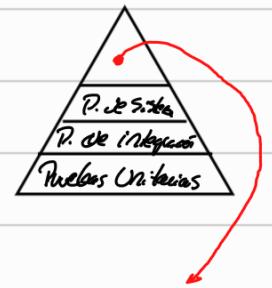
El empírico se basa en 3 pilares: inspección, adaptación y transparencia, necesitanos general experiencia por lo que hacemos procesos cortos. Quienes trabajan en empirismo tienen q' trabajar en ciclos de vida iterativos si o si para así adoptar la inspección y la adaptación, mientras más corto el ciclo mejor pq' recibe la retroalimentación más rápido y más fácil serán conseguir el envío. La transparencia viene por el manejo de la información.

Adaptación

Requerimientos → A diferencia de los procesos definidos donde teníamos los CV y trago fino, en los empíricos tienen las User Stories.

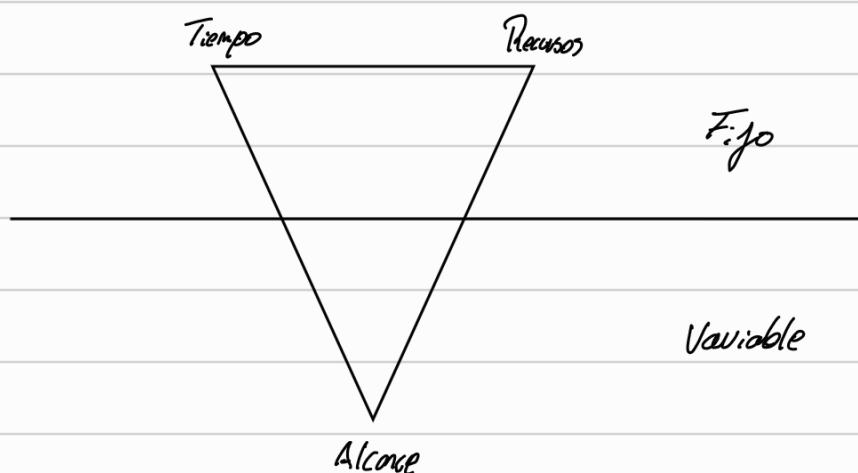
- Token para una conversación con el usuario.
- 3 C's: tangible (card), conversación y continuación
- El template lo define el equipo

Frase Verbal	Estimación
Como [rol] quiero / puedo [acción / que] para [valor de Negocio] Hemos q' hacer q' el usuario acepto como completado la US	[Criterios de aceptación] P. de Sist. P. de integración Rebasas Unid. fin.



Si tienes muchas (20+) podes
seguir: CAV q' haya q' dudar
la user Story

Req. ágiles → los caract. la entrega de valor de Negocio, trabajan con procesos empíricos (q' estos los caract. la entrega tiene todo el soft. funcionando). En la gestión ágil de los requerimientos uno no pade con la descripción total del producto a construir, este no esta 100% definido. \oplus



① Tenemos una visión del producto, en base a este definimos el product backlog d
Cuál está siempre cambiando y nunca contiene al 100% los requerimientos.

→ Lean → Relieve a atrasar las decisiones hasta (en la medida de lo posible) tener toda la información.
Eliminación desperdicios y obtener las decisiones → JIT

User Stories (US)
Card
Conversation! la más importante
Confirmation

En cuanto a la granularidad:

User stories < Epicas < Tareas
US más grande
contiene epicas y US

Criterio de list: checklist definido por el equipo para validar si una US puede ser elegida. Caract. mínimas del INVEST model: independiente, negociable, valorable, estimable, small, testeable.

Elegible en cualquier momento (le das libertad al PO)
La parte negociable es el "cómo".
Esto se ve en el "pala".
Redigible en una reunión
No la puedes estimar x demasiado incertidumbre
Puedes de aceptación (confirmación)

SPIKE: US con investigación x lo q' no puedes estimarla. Viene de 2 tipos: funcionales y técnicas. Va en el backlog de igual manera, se le asigna un tiempo q' le voy a dedicar en x sprint (no es lo mismo q' una estimación) a partir de esto se la cancela o se la convierte en una US. Puede pasar q' una US se convierta en un Spike.

Estimación → Se considera todo lo necesario para presentarlo al cliente (caso de prueba, revisión de paos).

Do done: **No lo dio**

Estimación ágil: siendo una estimación una predicción de un valor, antes de preguntarnos el cuándo y el cuánto debemos preguntarnos el qué. Son **velocías**, los datos humanos sonos buenos comparado, no sonos buenos haciendo absolutas, implica elegir un elemento **concreto** fuerte

al cual comparten, en este caso, una US. Hacer ~~todo en la categoría~~, no en la precisión (que día, a qué hora, etc.). Estima ^{el equipo} ~~quién hace el trabajo~~.

El punto de estimación se llama *Story Point*, indica el tamaño de la US en 3 dimensiones: complejidad, esfuerzo y duda (incertidumbre).

↓
nivel de detalle menor, en base a la comparación y vedere el error.

↓
Ciclo experimental, para lo q' es este punto → en validación para los user points
Se suele calcular con la serie de Fibonacci → 0, 1, 2, 3, 5, 8, 13
(Hasta 13, si lo pasa ya es una época)

↓
aca va otro ↑ pero en los cartas de poker planning este valor desaparece no se repite

Poker planning → usa la serie Fibonacci

Wide band delphi: Elijo una US, alguien del equipo la presenta, todo el equipo daje una carta con un valor (con el valor de fibonaci) individualmente y oculto.

Canónica: tiene valor 1 (lo hacen los equipos q' emplean pq' no tiene matices para arriba, ni abajo, ni arriba a 1) idealmente la canónica no se cambia en todo el proyecto.

referido a las 3 dimensiones del US

¿Cómo se mide el esfuerzo? en horas ideales, siendo estos depurados de desperdicio pero en qual. son 6hs sobre la jornada de 9hs.

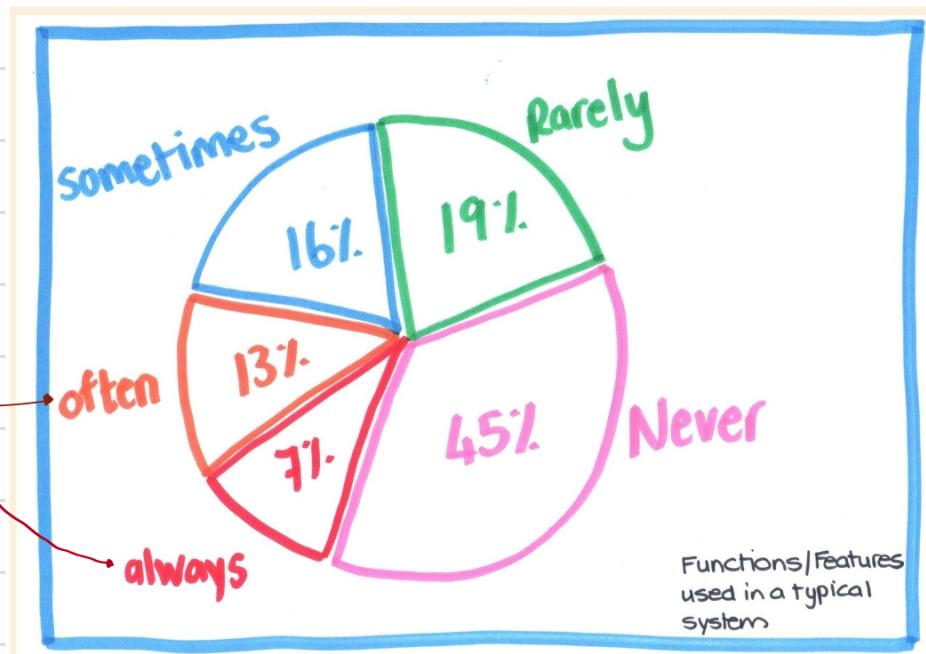
Gestión de productos → ppt nº 5

Ante la complejidad de hacer software aparece la necesidad de tener una unidad p/ gestionar el desarrollo, aparte el proyecto (considerar los 3P's). Se confunde un poco la diferencia entre proyecto y producto (entend q' los ciclos de vida son distintos, siembra q' el ciclo de vida del proyecto es más corto q' el del producto).

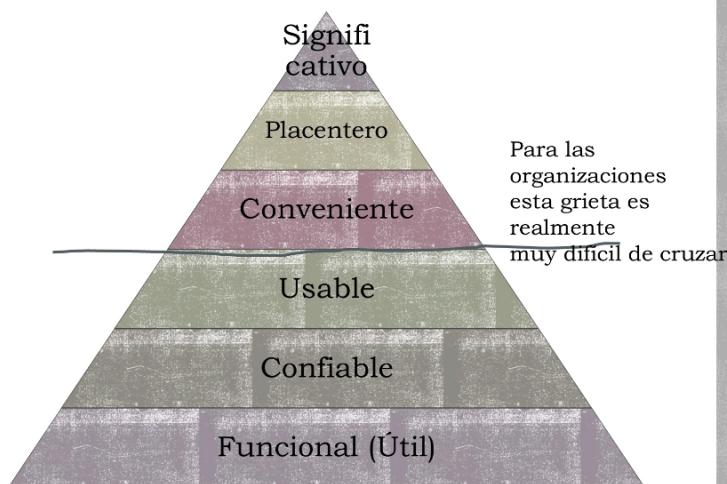
¿Para qué creamos productos? más allá del dinero, puede ser apartar a la comunidad (cambiar el mundo), satisfacer al cliente (si hubiera uno "atado" al producto) o aumentar la cantidad de clientes usándolo.

Al hacer un producto decidirnos cómo dividirlo para entregarlo, hace ahí el MVP con el fin de validar el interés del mercado en su producto. No entra

el concepto de un cliente q' nos pide el producto. (corresponde (entiendo) con este 20% o, incluso, el 7%.



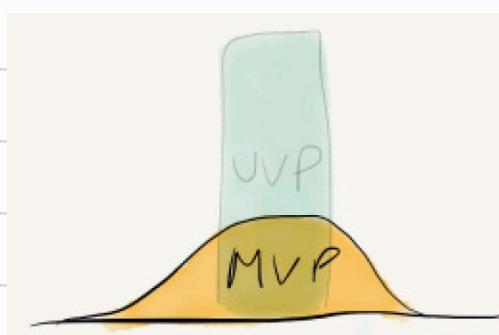
Focalizado en experiencias (gente, actividades, contexto)



Aparece el hecho de cambiar la visión a la hora de desarrollar un producto Significativo.

Focalizado en tareas (productos y características)

MVP → Aclaración, un MVP no necesariamente es código, puede ser un prototipo, un powerpoint, un video, etc.



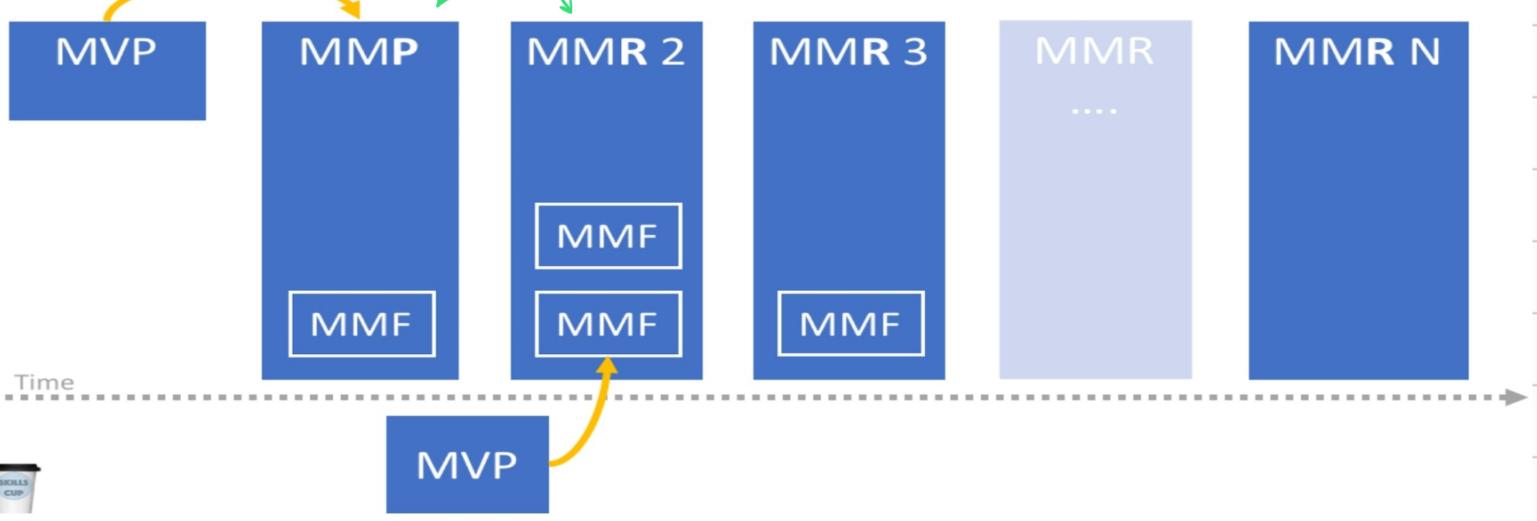
El MVP aparece para probar una hipótesis:

- * ¿Tendrá clientes?
- * ¿Hay mercado?
- * ¿Ya lo hace alguien?

No confundir con MVF (Feature), para que un MVP sea un MVP. Pueden ir apareciendo varias versiones de MVP y luego aparecer los MUF (Minimal marketable feature) en lo referido a que sea verdible. También MVF (se sacan features para verlas validando, la V= variable). (Puede q' me hayan faltado items, en la pot están errados).

Se avanza una especie de dinosaurio (ver ppt) de la forma

Puede pasar q' estos 2 sea lo mismo xq cada iteración del producto es una versión.



Con d MVP debemos obtener información para futuras iteraciones, como avanzar o debe avanzar el producto. Para constituirlo aparecen los conceptos de **valor** y **desperdicio** (todo lo q' no genera valor).

"La calidad del producto final está condicionada x la calidad del proceso q' uso para desarrollarlo".

Mejoramos a través de descartar desperdicios, Lean habla de 7 desperdicios (luego aparece un 8vo).

Unidad 1 → Componentes de un proyecto de SW → ojo! lo puedes preguntar

Volvemos a diferenciar de procesos de físicos (1) y empíricos (2):

- (1) Los define alguien q' no lleva a cabo el proceso, q' busca repetirse, p/ SW estos
- (2) Lo define el equipo de desarrollo, busca q' a través de la exp. puedes organizar el proyecto.

No es lo mismo q' un proceso, es el conjunto de estados pa' el q' pasea el proyecto desde q' se define iniciales mientas q' el proceso es el conjunto de tareas q' tiene q' hacer para cumplir el objetivo. El ciclo de vida te recomienda hacer las tareas en cierto orden, hay muchos ciclos de vida pero se pueden agrupar en 3 tipos: secuencial, iterativo y recursivo.

Iterativo e incremental

espiral

→ complejos, los de mayor éxito en el mercado y suelen mostrar crecimientos solo

cescada

en períodos de mucho riesgo.

La elección de ciclo de vida es indep. del tipo de proceso elegido, se recomiendan unos en base al elegido pero son eso, recomendaciones.

Perito de los ciclos de vida iterativo, no son todos iguales. Se dividen en 2 tipos: de alcance fijo y de alcance fijo.

PUD

SCRUM

Proyecto

Único de gestión q' administra recursos. Para ser llamado proyecto debe dar un resultado único, tiene una duración limitada (fecha de inicio y fin determinada), tiene un objetivo de elaboración gradual y tareas interrelacionadas. En esta última es donde veo el proceso.

Proyecto q' no se planifica está condenado a fracasar. Lo primero a definir es un nombre, luego, un Objetivo (q' necesidad voy a satisfacer) este tiene q' ser claro y alcanzable. Hecho esto definimos el alcance del proyecto (todo el trabajo q' hay q' hacer para cumplir el objetivo). El plan de proyecto debe tener explícito el objetivo y alcance, es el artefacto q' responde todas las preguntas q' vamos a tener.

➔ importante

El objetivo y alcance quedan sentados en la ERS.

DEL PROYECTO!

DEL PRODUCTO!

Plan de Proyecto

- Objetivo del proyecto
- Alcance " "
- Definición del proceso y del ciclo de vida
- Estimaciones

Claro

Alcanzable

- Tamaño del producto (veg iterativas, cu, alcance)
- Esfuerzo (llaves pasos lineales, asunto q' es 1 paso se trabaja + pasa a la vez)
- Tiempo/calendario
- Costo

→ Recursos Críticos (depende del proyecto, pueden ser licencias de algún SW, acceso de banda, etc.)

→ Riesgos: la probabilidad de q' ocurre algo q' afecte negativamente a mi proyecto. Si tiene probabilidad = 1 no es un riesgo, es un problema.

Probabilidad

Impacto

Exposición → Para comparar riesgos

Métricas

Seguimiento: controles, reportes, cada 1 cuarto, etc. todo lo q' necesites.

Software Configuration Management (SCM)

Ing. de soft. está compuesto x disciplinas técnicas (Req., Análisis, Diseño, programación, prueba y despliegue), disciplinas de gestión (planificación de proyectos, seguimiento y supervisión) y las de soporte (SCM, métricas y aseguramiento de calidad de proyectos).

Software ≠ Código, Sino q' Soft = información

Todo artefacto que sujeto de una tarea llevada a cabo en un proyecto es Software (por ejemplo, US, plan de proyecto, etc.).

Ante la facilidad q' tiene modificar código (pj. borrar archivos) aparece la necesidad de SCM.

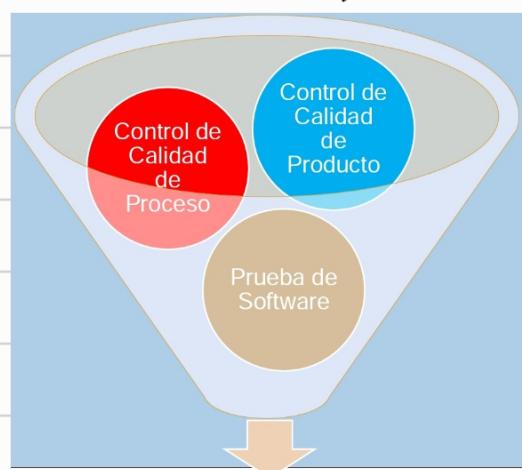
SCM → Es una disciplina transversal en el sentido en q' está en todo el proyecto.

Nos va asegurar integridad de los ítems del proyecto y así asegurar calidad del producto mientras se lo construye.

Definición de SCM:

Su propósito es establecer y mantener la integridad de los productos de software a lo largo de su ciclo de vida.

Disciplina q' ...



Aseguramiento de Calidad de Software

Trabajamos con ítems que varían con el tiempo, un ítem en un momento dado es una versión, es importante identificarla, identificar qué es lo que se quiere y asegurar su integridad.

Trazabilidad → vínculos de interacción entre ítems. Puede ser pre-ERS o Post-ERS.
Permite hacer un análisis de impacto.

Para rendir, tener en cuenta estos ítems!



Configuración de software → instante de un software en un momento determinado.

Línea base → ítem o conj. de ítems de config. en estado de aprobación que están en un estado estable. Se marca etiquetado. Hay 2 tipos: especificación, producto.

Auditaria → Solo la línea base, ya que, es estable.

