

1º parcial: leer artículo ppt.

clase 03/01

estimaciones ágiles.

dogmático = extremo → depende sin importar qué.
no tiene flexibilidad.

pragmático = se ajusta/adapta al contexto. No flexible.

Estimación

- tiene asociado a ella cierto grado de incertidumbre.
- no son certezas, ni promesas, ni compromisos.
- tienen asociado un grado/porcentaje de probabilidad.

* Es clave ser PRAGMÁTICO, es la clave del éxito.

estimación ágil

→ estima quien hace el trabajo. Estimaciones colaborativas de quienes hacen el trabajo.

story point = medida de complejidad, riesgo y esfuerzo.

unidad de medida que se utiliza para estimar User stories es: story points.
de una user story

→ estimaciones relativas: aceptación de que somos más eficientes haciendo comparaciones. Necesitan un marco referencial para comparar.

↓ absolutas → en las relativas uno hace un proceso de comparación.
se trabaja con un nivel de precisión ↓ de comparación.
que 1 medida que hay 0 info. o más difícil de lograr.
ej: E.AB: cuanto pesa esto.

!!
La apuntan a la certeza, no a la precisión.

Tamaño

→ medida de la cantidad de trabajo necesario para producir una story.

E.rel: qué pesa más

→ se elige una escala y luego no se cambia.

escala serie de Fibonacci

→ complejidad: dimensiones, distintas interpretaciones.

→ la escala debe ser 1)

→ el tamaño ≠ al esfuerzo para construir la story. → esfuerzo.

exponencial, ya que la complejidad del software crece de esta manera.

≠ al mundo → calendario

de donde sale el valor de punto final?

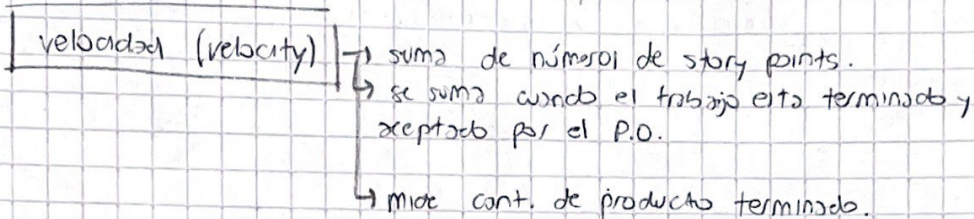
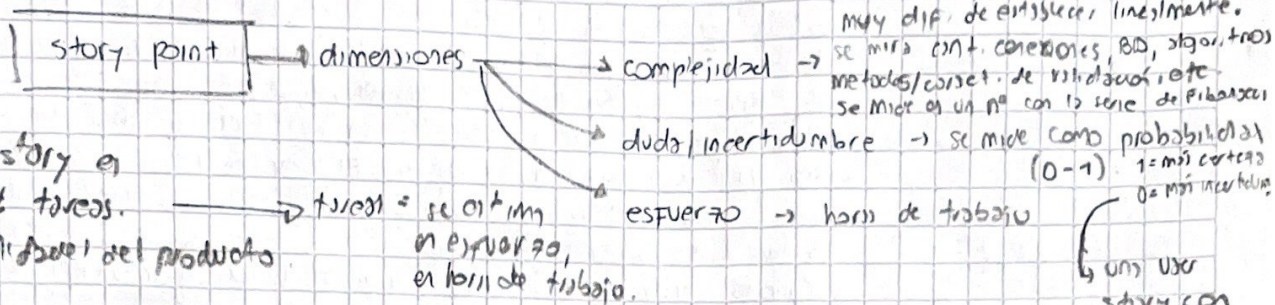
las tres dimensiones hacen el mismo grado de importancia? o por ej: la complejidad tiene mayor peso / genera más puntos que por ej el esfuerzo?

HOJA N° ICS (3)

FECHA



una user story es producto ≠ tareas.
son funcionalidades del producto.



una user story con muchos detalles breves
↓
SPIKE

¿cómo decodificar estimaciones?

partimos del product backlog

V = validador
¿verificar el producto?

MVP → producto mínimo viable.
↳ baja viabilidad

MMF → pieza más provechosa de funcionalidad

MVF → correct. mínima viable

MMF → características mínimas del ~~producto~~ release

MMF → características mínimas comercializable.

armar un MVP:

- 1) encontrar nicho de mercado → a quien le sirve/interesa el producto.
- 2) crear Roadmap → mapa de ruta / plan de alto nivel.
- 3) identificar la competencia → investigar.
- 4) pre-vender MVP → buscar inversores.
- 5) probar / testear algo / una parte del producto → testear suposiciones.
- 6) Asegurar que el MVP cumple/resuelve el problema correcto.
- 7) Focalizar en funcionalidades principales.

sobre user storys :

una US. de más de 8 puntos no la tomamos, como una sola, la partimos ya que no es consumible en una iteración y si no se termina en esa iteración no agg valor.

Al partirlo hay que ver si es Small o no (que sea consumible en una iteración) y que tenga valor para el negocio.

Algo muy chico que se termina en una iteración pero que no agg. valor de negocio No es una US. El valor de negocio es el límite para cuando queremos partir una US, si la parte no tiene/no genera valor de negocio no la partimos.

Cuando partimos, US se agg complejidad por las interfaces de comunicación con las demás partes.