09/03 Estimaciones Agiles y Gestión de producto

Tuesday, September 03, 2024

11:35 AM

Dogmatico -- defender una idea hasta la muerte.

Pragmatico -- tiene la capacidad de adaptarse al contexto o situacion. Es la clave del éxito en casi todo.

No ser dogmatico.Ser pragmático es la clave del éxito. (poder cambiar el pensamiento y dudar).

Tenemos siempre una presion para poder informar algo cuando no tenemos una informacion certera.

Scrum recomienda estimar con poker plannig

|  |  |
| --- | --- |
| **Estimar** | Cuando uno habla de estimaciones, estamos hablando de una situacion en la cual hay un universo de insertidumbre por que falta imformacion. |
|  | No son certesas, no son compromisos |
|  | Tienen un nivel de probabilididad, por ende puede fallar. |
|  | Debemos asumir que las estimaciones son promesas |
|  |  |

# Estimaciones Agiles

Si las estimaciones se utilizan como compromisos son muy peligrosas y 
perjudiciales para cualquier organización. 
Lo más beneficioso en las estimaciones es el "proceso de hacerlas". 
La estimación podría servir como una gran respuesta temprana sobre si el 
trabajo planificado es factible o no. 
La estimación puede servir como una gran protección para el equipo. 

* Las estimaciones son una protección frente a no tener nada.

**Collaboration – Clave de la estimación en Agil**

* Estima en colaboración con el equipo. Estima el que hace el trabajo. No hay un guru estimador que dice cuanto el equipo va a tardar.
* Clave es colaboración de la gente que va a hacer el trabajo

**Estimaciones Relativas**

* Somos mas eficiente haciendo comparaciones. Es mas facil que hacerlo en terminos absoluto.
* No tenemos presicion osea hay ambiguedad (ej para dos edificios casi iguales no vamos a tener certeza pero vamos a asumir una postura y la decision no tiene que ser con mucho esfuerzo por que si no es mas caro el remedio que la enfermedad).

**Story Points**

* Es la medida que usamos para estimar. Medida para hacer una estimacion respecto del tamano de la User Story =

**Tamano**

* Es la medida de la cantidad necesaria para producir una feuture/story.
* Problema que se confunde tamano vs esfuerzo
* Es que vas a hacer el tamnao
* Esfuerzo es confundir el tamano con lo que yo quiero construir

Hay que tener cuidado por que asumimos compromisos basados en estimaciones.

**Tamano vs Esfuerzo**

Las estimaciones basadas en tiempo son más propensas a errores debido a varias razones.

* Habilidades
* Conocimiento
* Asunciones
* Experiencia §
* Familiaridad con los dominios de aplicación/negocio

**Forma de trabajar las estimaciones**

* Serie de fiboncci -- es una serie exponencial. La complejidad del software crece en modo exponencial tambien

**Tres dimenciones que componen el Story point**

* Complejidad
* Esfuerzo
* Duda

**Velocidad/Velocity**

* Mide cantidad de puntos de historia que el product ouwner me acepto . Suma trabajo terminado y aceptado medidos en puntos de historia. Por que la mejor metrica de progreso es el software funcionando.

**Poker estimation**

* Convina la idea de la serie de fibonacci para asociar los numeros

conejos 
1 
mes 
20 mes 
mes 
40 me 
50 mes 

* El cero es parte de la serie de fibonacci.
* Canonica
* Spyke

No es lo mismo tareas que user storys las tareas tambien se estiman pero en horas.

La user story la puedo partir. Pero las particiones deben tener un valor de negocio para ser una US

# Gestión de Productos

Validacion que yo cree un producto y que el mercado lo consuma.

|  |  |
| --- | --- |
| MVP – Producto minimo viable | Apunta a la version minima para validar si el mercado lo quiere o no lo quiere. Apunta rapido y barato. Descripcion de un producto que no tengo para ver si invertir. |
| MMP—Minimal comercializable  Product  M es marketianizable | Se usa cuando ya tengo un producto que quiere salir a produccion. |
| MMF | Una sola caracteristicas del producto |
| MVF | Una sola caracteristica del producto para validar |

.V. --> viabilidad

.M. --> marketianizable

**Preparar un MVP**

* **Encontrar el nicho**
* **Reoad Map –** Plan de alto nivel
* **Investigar la competencia.**
* **Provender el MPV**
* **Testear las supociciones**
* **Asegurarse que el MPV resuelve el problema correcto.**

**Caracteristicas de un MPV**

* La idea de un mvp es avanzar verticualmente

Diwho 
ñabdidad 
Funcionalidad 
Cómo Si hacer un MVP 

**Priorizacion –**

O 
2-4 
2-4 
Urgente 
1 . Hacer 
Incluir en el MV P 
3. Delegar 
Considerar integraciones de 
terceras partes. 
No urgente 
2. Planear 
Desarrollar para un lanzamiento beta 
4. Eliminar 
Remover del Roadmap del producto 

Estan pensados en valor de negocio.