

Ejemplo de Backlog y resumen de Scrum

Que es SCRUM?

Es un grupo de técnicas y recomendaciones que proponen una dinámica muy unida de grupo para funcionar optimamente. Lleva tiempo que todos los miembros del equipo comprendan su funcionamiento completamente y mas todavía que vean los beneficios. De todos modos, scrum mismo indica que lo mejor es utilizar solo las herramientas que el equipo crea conveniente ya que cada configuración del equipo y cada proyecto posee sus propias limitaciones y capacidades. Scrum posee un ciclo de repetición necesario para poder existir como método de trabajo. Se puede decir que SCRUM es una manera de organizar un proyecto.

Que son los Sprints?

SCRUM propone dividir el desarrollo en pequeñas etapas a superar. Por un lado esto hace que el desarrollo no se vea como un gigante a vencer, sino como pequeños pasos a dar hacia un destino al que nos acercamos de a poco. No desmotiva, y hace factible y visible la evolución de un producto. Estos períodos de 1, 2 o **aveces 3 semanas**, se llaman **SPRINTS** (corrida de poder).

Cada sprint se planifica en equipo al principio del mismo, especificando que historias del “Backlog de Producto” (BP) general se harán avanzar en este período. El resultado de este trabajo es el “Backlog del Sprint” (BS) que será utilizado por el equipo durante estas semanas.

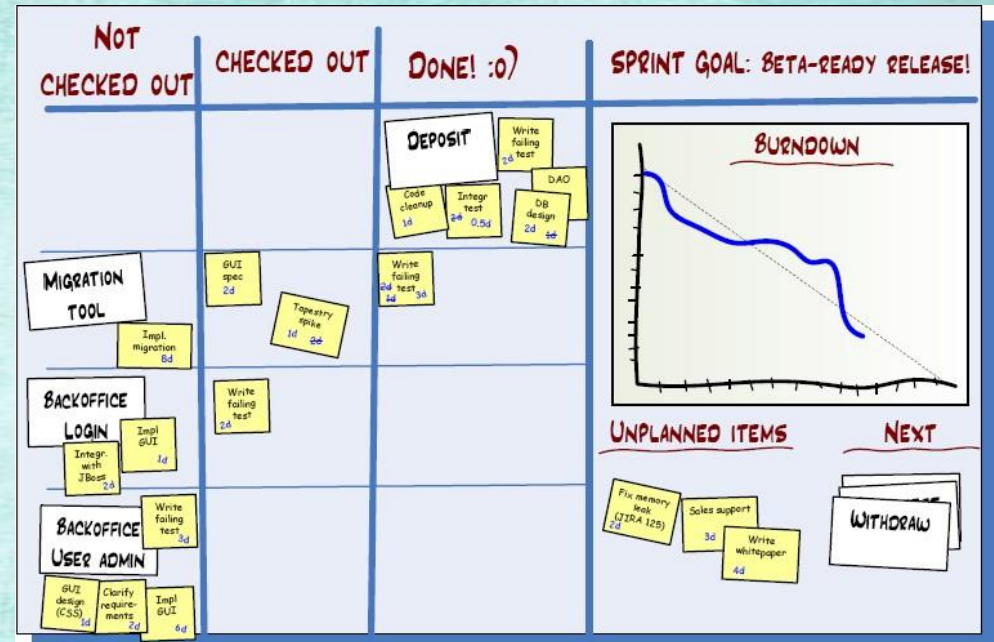


Tablero de tareas

El tablero de tareas es una herramienta visual que permite al equipo estar mas sincronizado e informado. Al ser completamente visual permite enterarse rápidamente de lo que hace el de al lado.

Al indicar que es un tablero, se está hablando de forma literal. Se trata de una pared o un pizarrón donde se pegará un papel “post it” por cada tarea que figure en el BS. El tablero debe dividirse en 5 columnas, que albergarán las tareas. Los nombres de las columnas son: “HISTORIA”, “SIN HACER”, “EN DESARROLLO”, “TERMINADAS A VERIFICAR”, Y “CONFIRMADAS” tal como se ve en la imagen.

1. Al principio del sprint, los post it que representan las tareas comienzan todas en “SIN HACER”
2. Cada persona toma una tarea de esa columna y la pega en la columna “EN DESARROLLO” para indicar que dicha tarea ya esta tomada.
3. Al terminar con esa tarea, deberá mover el post it a la columna de “...A VERIFICAR”. Y vuelve al punto 2 para seguir con el desarrollo.
4. QA irá tomando las tareas entrantes en la columna “...A VERIFICAR” y la busca en el backlog para chequear si se cumple la condición de verificación. De ser correcto, debe reposicionar los post it en la columna “CONFIRMADAS”.



Reuniones

Las reuniones de Scrum deben ser cortas. Para ello se recomienda hacerlas en una habitación sin distracciones y sin sillas. Todo es mucho mas rápido de este modo. Si en algún momento los miembros se van de tema y foco (off topic) cualquiera de los presentes puede hacer “sonar la campana” indicándolo, y todos deben volver al tema en cuestión. Las siguientes son las diferentes reuniones propuestas por SCRUM. Cada empresa puede adaptarlas a sus hábitos o limitaciones.

Planificación del sprint

Todo el equipo se reúne durante medio día para analizar el estado del BP y definir las tareas a agregar al nuevo BS. Aquí se suelen efectuar correcciones a las estimaciones y se agregan y quitan tareas o incluso historias.

Reunión de Sprint diaria

Todas las mañanas, se hace una reunión de nomas de 15 minutos para charlar lo que cada miembro esta realizando, hacerse preguntas entre los miembros del equipo sobre las tareas y sobre todo las necesidades.

Reciclado del sprint

Luego, todas las tareas del BS deberían estar concluidas. De no estarlo, deben ser recicladas para el proximo Sprint con alta prioridad.

Análisis del ultimo sprint

Al terminar el sprint, se debe realizar una reunión para analizar el desempeño del equipo, analizar lo que salió mal para intentar corregirlo, y lo que salió bien para potenciar el próximo sprint.

Tecnología recomendada

Google Docs / Google Drive

En la dirección web.drive.google.com se puede encontrar una webapp tan completa como útil. Es como un disco rígido virtual con aplicaciones muy útiles: Spreadsheet, Document, Presentation, Form, y Drawing.

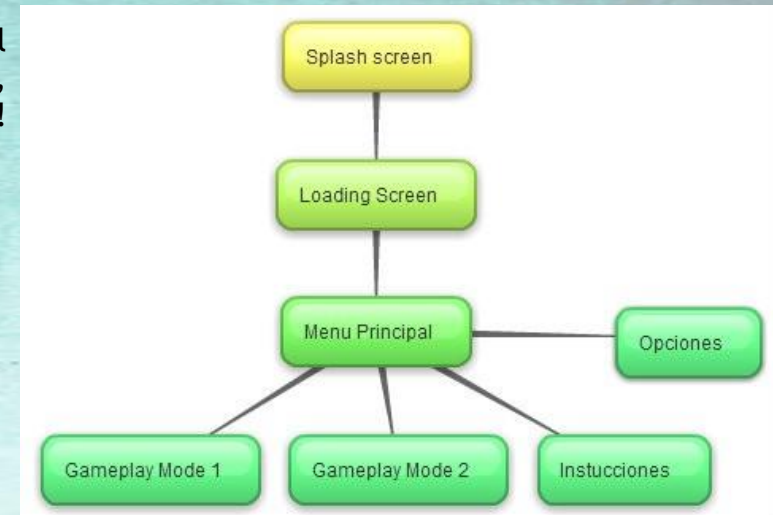
Uno puede crear un spreadsheet (una hoja de cálculo como el excel) online, sin tener que bajarse un software, es gratuito, y lo puede compartir el archivo con la gente del mismo equipo. Incluso se puede especificar si la persona con la que se comparte tiene permisos para editar o solo para ver la información.

Lo mas interesante es que si 2 o mas personas se conectan a la vez, y editan el mismo archivo de forma online al mismo tiempo, van a verse trabajando en tiempo real. Incluso los archivos incluyen un chat en un costado para poder potenciar el trabajo en equipo todavía mas rápido.

Lo mismo con los “Document” (que es el equivalente al Word de Microsoft Office) donde varios pueden escribir un archivo de texto a la vez.

Bubbl.us

Esta pagina web permite hacer gráficos de nodos de forma simple, rápida y el resultado es altamente estético. Es recomendable para todo tipo de gráficos, de estado, de inteligencia artificial, estructura de menues, y otros! RECOMENDABLE.



Microsoft Visio / Gantt Project

Si se quiere llevar al extremo de la organización el proyecto, y para poder calcular tiempos, y camino crítico (el camino minimo desde el comienzo del juego hasta tener un jugable, sin tomar en cuenta el desarrollo en paralelo) se recomiendan estos softwares.

El lider del proyecto puede listar las personas y de que equipo son, las tareas, las dependencias entre ellas, cuanto van a llevar, en que fecha empiezan, y se va formando un grafico gantt que permite administrar el avance del proyecto. De todos modos es un extremo de la organización que no todos aprecian.



Que es un Backlog?

El Backlog es un “entregable” propuesto por SCRUM que trata de un documento de tipo hoja de calculo (ej. excel) que permite organizar y predecir las tareas que el equipo de desarrollo debe efectuar en un plazo corto de tiempo. Ejemplo de un Backlog de un plataformero:

		Tiempo total				
	Historia	Estado	Estimado (días)	Fecha inicio Estimada	Dependencia	% finalizado
A	Movimiento del personaje	Iniciado	13	xx/xx/xxxx	C,B	20%
B	Nivel de prueba	Terminado	0	xx/xx/xxxx	-	100%
C	Personaje Estetica	Iniciado	20	xx/xx/xxxx	-	80%
D	Enemigo de prueba	Sin comenzar	15	xx/xx/xxxx	B	0%
E	Estructura y menues	Sin comenzar	5	xx/xx/xxxx	-	0%
F	Interfaz de usuario	Terminado	0	xx/xx/xxxx	-	100%
G	Sistema de puntaje	Sin comenzar	6	xx/xx/xxxx	B	0%
H	Condiciones de victoria y fracaso	Sin comenzar	5	xx/xx/xxxx	G	0%

Tarea ID	Historia ID	Tarea	Estimacion (días)	Depto Responsable	Dependencia (ID)	Prioridad	Estado (%)	Condición de aprobación	Aprobado
1	A	Definir movimientos del personaje	1	Game Design	11	100	100%	El texto debe indicar cuales son las mecánicas que el jugador posee como opciones a realizar.	OK
2	A	Tomar Input del teclado	1	Programación	1	50	100%	Identificar las teclas siendo presionadas en el teclado, hasta 3 a la vez como mínimo.	OK
3	A	Ejecutar animaciones	4	Programación	2,6	50	20%	Ejecutar animaciones según lo que presione el teclado. Cancelar acciones cuando nuevas acciones contrarresten la anterior.	-
4	A	Respetar mapa de durezas	2	Programación	5	50	20%	...	-
5	A	Mapa de durezas Placeholder	1	Arte	-	100	0%	...	-
6	A	Animaciones Placeholder	2	Arte	-	100	0%	...	-
7	B	Diseño funcional del nivel (diseño de los retos)	2	Game Design	-	100	100%	...	OK
8	B	Fondo	4	Arte	7	80	100%	...	NO OK
9	B	Objetos	2	Arte	8	50	100%	...	NO OK
10	B	Mapa de durezas	1	Game Design	7	80	100%	...	OK
11	C	Backstory de personaje principal	1	Game Design	-	80	100%	...	NO OK
12	C	Conceptos de personaje principal	2	Arte	11	50	50%	...	-
13	C	Assets de personaje principal	3	Arte	12	50	50%	...	-
14	C	Listado de animaciones	1	Game Design	11	40	0%	...	-
15	C	Animaciones finales	4	Arte	14,1	90	0%	...	-
16	C	Implementar animaciones finales	1	Programación	15	90	0%	...	-
17	D	Backstory de personaje enemigo	1	Game Design	-	50	0%	...	-
18	D	Assets de personaje enemigo	2	Arte	17	40	0%	...	-
19	D	Animaciones finales	4	Arte	18	40	0%	...	-
20	D	Implementar animaciones finales	1	Programación	19	90	0%	...	-
21	E	-
22	E	-
23	E	-

Para hacer un backlog en 5 pasos simplificado:

1. Historias: Escriben los grandes temas que faltan. Ejemplos: "player de sonido y música" o "movimiento del personaje" inicialmente solo lo prioritario para llegar la próxima entrega.
2. Establecen una prioridad de las cosas que definieron (ejemplo bobo. es mas importante el sistema de movimiento, que el sistema de sonido), a menos que sea un juego de música :P
3. Al terminar el equipo debe ir 1 por 1 con estos temas y detallar que se necesita para terminarlo.
 - Ejemplo: "movimiento del personaje" se divide en.
 - listado de movimientos (game designer),
 - animaciones en loop y entrando y saliendo de standing (artistas),
 - programación detectando input del teclado (programador).
 - Este listado de tareas tiene que ser lo suficientemente pequeño como para que sea lógico y no aterrador. No se debe poner tareas como "HACER TODOS LOS NIVELES" porque el que lo lea, se va a asustar, pero tampoco se debe poner cosas como "HACER EL COLOR ROJO DEL PRINCIPIO DEL NIVEL 1". Todo en su buena medida.
4. Se efectúa una reunión y se ponen las dependencias. Ejemplo: "el programador y el animador necesitan del game designer que defina los movimientos", y si da el tiempo y las ganas, se hace un pequeñísimo grafiquito de globitos (www.bubbl.us) para ver el orden de desarrollo. A veces ayuda.
5. Sabiendo cuanto tarda alguien en hacer cada tarea se pueden planificar tareas en paralelo en diferentes cosas sin que nadie esté esperando.