ENTORNOS DE DESARROLLO

IES Santiago Hernández
Curso 2017-2018
Ignacio Agudo Sancho

Metodologías Ágiles

- Introducción
- Manifiesto Ágil
- Principales metodologías ágiles:
  - Scrum
  - Kanban
  - eXtreme Programming

Metodologías Ágiles

- Introducción
- Manifiesto Ágil
- Principales metodologías ágiles:
  - Scrum
  - Kanban
  - eXtreme Programming

Metodologías Ágiles

- La finalidad será crear software de una manera más rápida que con las metodologías tradicionales.
- Trabajar con menos documentación.
- Funcionan como una combinación de las metodologías tradicionales.
- Surgen debido a la dificultad para acoplar las tradicionales con las nuevas tecnologías, nuevos lenguajes y con los programadores modernos.
- Intentan dar mas importancia al cliente y al equipo de desarrollo

Reflexión en clase:

- ¿Gana la liga el equipo con los mejores jugadores?
- Busca información:
  - ¿Cuales son los equipos con más presupuesto en la liga?
  - ¿Cuántas ligas ganó el F.C. Barcelona en las decadas de los 70 y 80?
  - ¿Cuántas ganó en los 90? ¿Cuál es la clave?
  - ¿Y en las décadas siguientes?
  - Busca otros equipos que hayan ganado la liga entre 1980 y la actualidad. ¿Cuál es su clave?

Enlaces:

- Presupuestos liga 17/18
  - http://lajugadafinanciera.com/presupuestos-la-liga-2017-2018/
- Clasificación actual
  - https://www.google.com/search?q=clasificacion+liga+santander&ie=utf-8&oe=utf-8&client=firefox-b#sie=lg;/g/11c6w1q_2s;2;/m/09gqx;st;fp;1
- Histórico de campeones de liga:
  - https://es.wikipedia.org/wiki/Primera_Divisi%C3%B3n_de_Espa%C3%B1a

Metodologías Ágiles

- Introducción
- Manifiesto Ágil
- Principales metodologías ágiles:
  - Scrum
  - Kanban
  - eXtreme Programming

Manifiesto Ágil

- Hay que comprender en qué consiste la metodología ágil.
- Es un documento en el cual se resume la filosofía de este enfoque de desarrollo.
- Aclara dónde se pretende llegar y cómo se pretenden conseguir los objetivos.
- Tiene 4 valores priorizados.
- Se basa en 12 principios.
- http://agilemanifesto.org/iso/es/manifesto.html

Manifiesto Ágil - Valores

- “Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:
  - **Individuos e interacciones** sobre procesos y herramientas
  - **Software funcionando** sobre documentación extensiva
  - **Colaboración con el cliente** sobre negociación contractual
  - **Respuesta ante el cambio** sobre seguir un plan
- Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.”

Manifiesto Ágil - Valores

- **Individuos e interacciones** sobre procesos y herramientas
  - Las personas consiguen los éxitos.
  - Si el equipo falla, el éxito pasa a incertidumbre.
  - El equipo no tiene que ser de las personas más brillantes del mercado, sino personas que simplemente sepan hacer bien su trabajo.
  - No imponer un entorno de trabajo antes de empezar.
  - Es mejor formar primero el equipo de trabajo y que ellos vayan creando su espacio de trabajo.
  - No necesitaremos apenas periodo de adaptación.

Manifiesto Ágil - Valores

- **Software funcionando** sobre documentación extensiva
  - No generar documentación a menos que sea sumamente necesaria en ese momento para tomar alguna decisión.
  - No dedicar tanto tiempo como en las metodologías tradicionales.
  - Documentación corta y breve.
  - Sin documentación, ¿cómo entra un nuevo miembro?
    - Código bien hecho.
    - Interacción con el equipo de trabajo.

Manifiesto Ágil - Valores

- **Colaboración con el cliente** sobre negociación contractual
  - Interacción constante entre el cliente y el equipo de desarrolladores.
  - Que el cliente vaya viendo cómo avanza el sistema y analice nuevas funcionalidades u objetivos.
  - No tiene que determinarse todo desde el principio.
  - El desarrollo nos puede llevar a muchas nuevas posibilidades.
  - El cliente queda satisfecho y hemos conseguido éxito.

Manifiesto Ágil - Valores

- **Respuesta ante el cambio** sobre seguir un plan
  - Evitar la planeación extensa y crear código que permita la expansión.
  - Si el cliente quiere incrementar objetivos, especificaciones o requerimientos, lo puede hacer.
  - El sistema debe ser flexible a cambios.
  - El cliente quedará totalmente satisfecho porque no ha tenido que conformarse con lo primero que se le ocurrió, sino que ha podido ir actualizando con las ideas que han ido surgiendo a medida que avanzaba el proyeto.

Manifiesto Ágil - Principios

- 1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- 2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- 3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- 4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- 5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- 6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

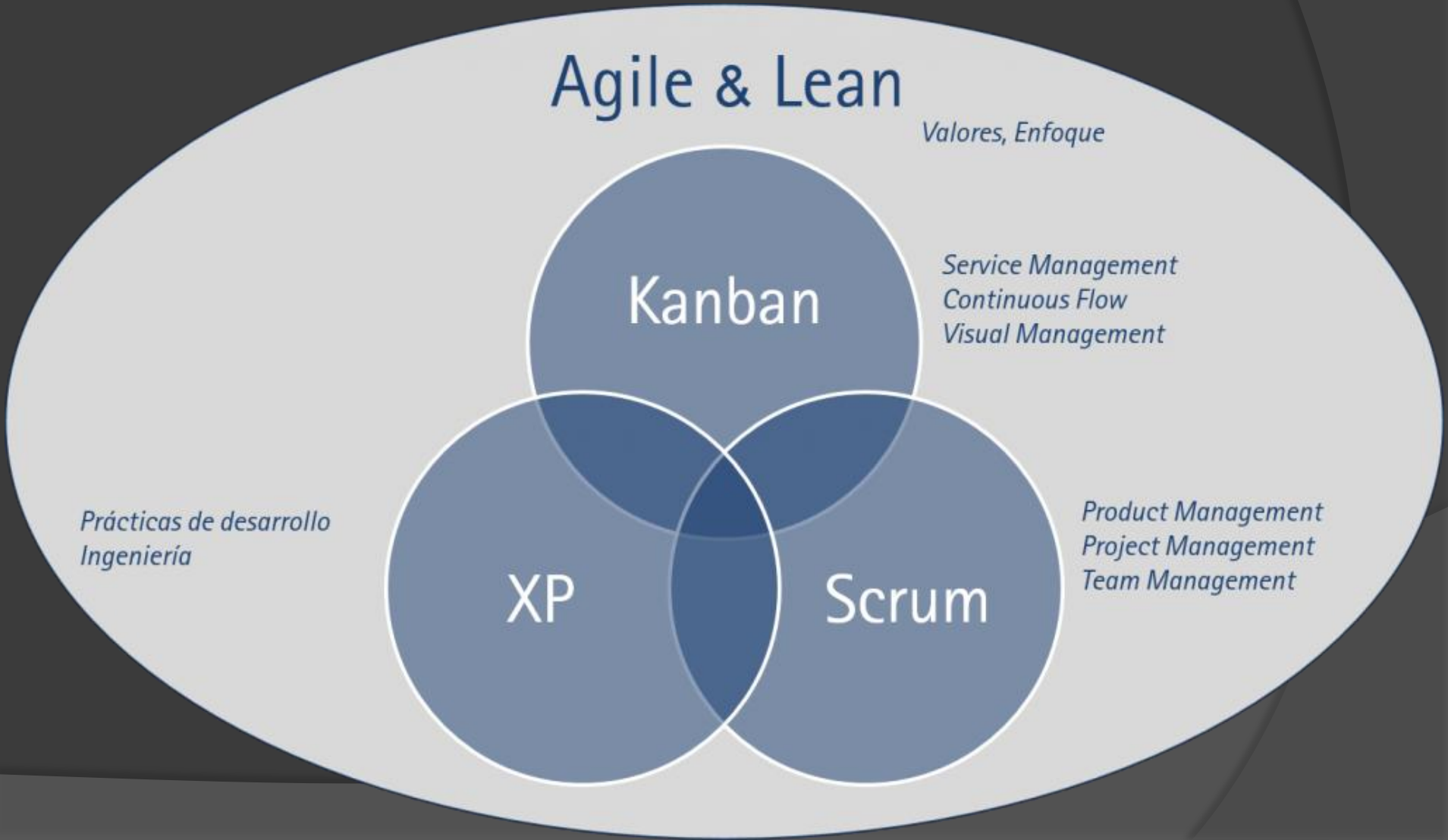
Manifiesto Ágil - Principios

- 7. El software funcionando es la medida principal de progreso.
- 8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- 9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- 10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- 11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- 12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Metodologías Ágiles

- Introducción
- Manifiesto Ágil
- Principales metodologías ágiles:
  - Scrum
  - Kanban
  - eXtreme Programming

Metodologías Ágiles



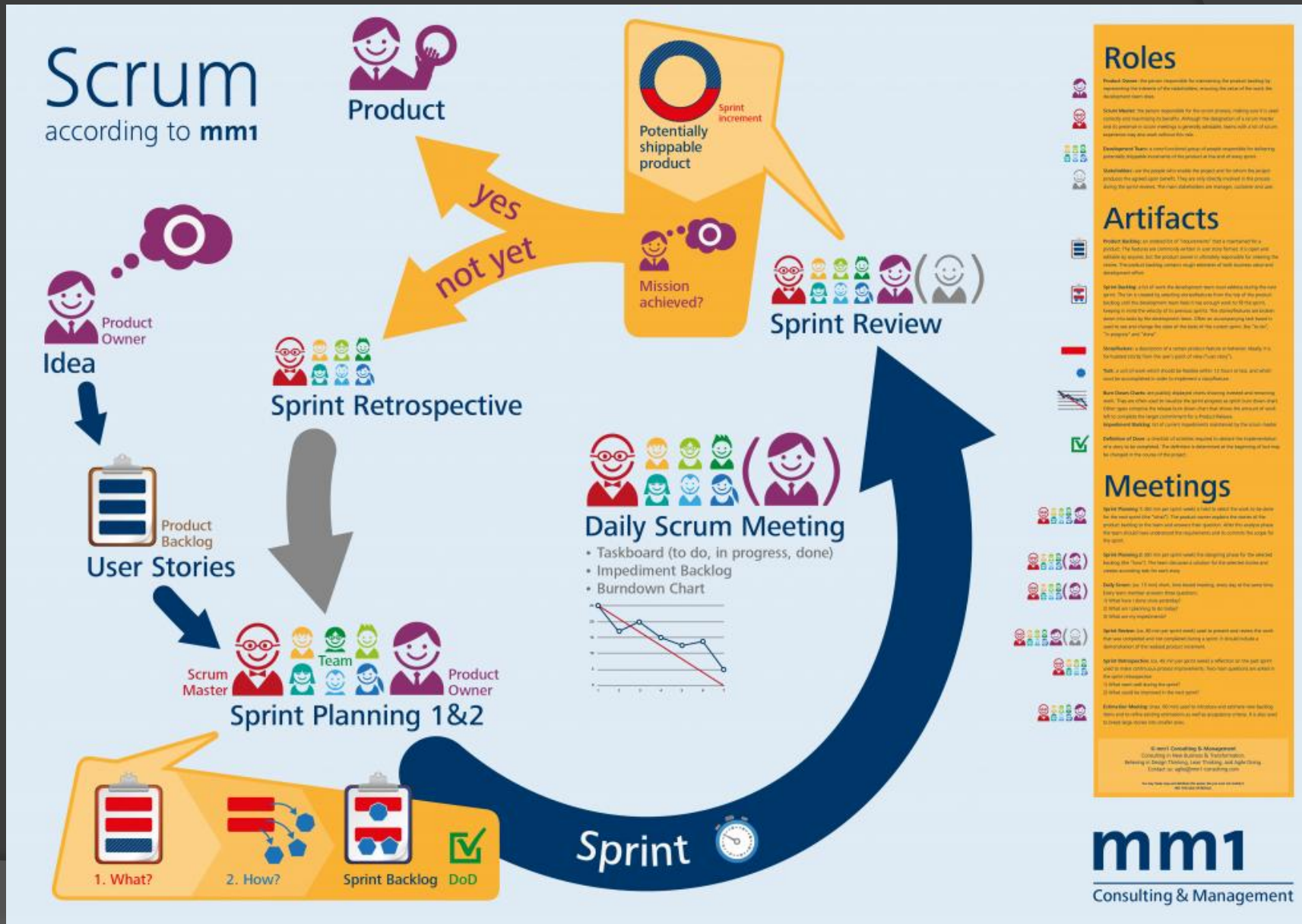
Metodologías Ágiles

- Los creadores de las metodologías crearon el manifiesto ágil.
- Pero cada metodología tiene
  - su propia personalidad y
  - características propias, que la hacen diferente de las demás.

Metodologías Ágiles

- Introducción
- Manifiesto Ágil
- Principales metodologías ágiles:
  - Scrum
  - Kanban
  - eXtreme Programming

Scrum


Scrum
according to mm1
Product
Potentially shippable product
Sprint increment
yes
not yet
Mission achieved?
Product Owner
Idea
Sprint Retrospective
Sprint Review
Product Backlog
User Stories
Daily Scrum Meeting
• Taskboard (to do, in progress, done)
• Impediment Backlog
• Burndown Chart
Scrum Master
Team
Product Owner
Sprint Planning 1&2
1. What?
2. How?
Sprint Backlog
DoD
Sprint
Roles
Artifacts
Meetings
mm1
Consulting & Management

Scrum

- Metodología amigable.
- Fomenta el trabajo en equipo.
- Trata de conseguir los objetivos de una forma rápida.

Scrum - Características

- Desarrollo incremental: el desarrollo se irá incrementando poco a poco, sin importar el orden.
- Calidad de las personas: la calidad dependerá de las personas, la auto organización y el conocimiento de los equipos de trabajo.
- Adiós al Secuencial y Cascada: existe el solapamiento. No importa en qué proceso estemos, que si otro necesita ser trabajado, se vuelve a él.
- La comunicación es fundamental: toda la información que se maneje será comunicada sin problema.

Scrum - Equipo

- Product Owner:
  - Líder del equipo. Los ojos del cliente.
  - Encargado del proyecto y de comprobar que va tal y como se ha acordado, cumpliendo las expectativas.
- Scrum Master:
  - Líder de las reuniones.
  - Ayudará en los problemas que hayan surgido.
  - Será un “facilitador” que tratará de minimizar obstáculos, pero no omitirlos.
  - Debe conocer en profundidad los lenguajes del proyecto para poder ayudar bien.
  - Puede picar código o no (cambio de sombrero)

Scrum - Equipo

- Scrum Team:
  - El núcleo de la metodología Scrum. El equipo de desarrollo (Los que pican código).
  - Encargado del desarrollo del software y de cumplir las metas u objetivos propuestos por el Product Owner.
- Cliente:
  - También forma parte del equipo.
  - Puede influir en el proceso.
  - Puede proponer nuevas ideas o hacer comentarios o sugerencias.
  - Puede coincidir con el Product Owner

Scrum - Funcionamiento

- Product Backlog:
  - Lista de los productos a desarrollar.
  - Debe ser elaborada por el Product Owner.
  - Debe estar ordenado por prioridades. De mayor a menor, de tal forma que las más prioritarias sean las que se hacen en primer lugar.
  - Debe darnos respuesta a la pregunta “¿Qué hay que hacer?”

Scrum - Funcionamiento

- Sprint Backlog:
  - Se seleccionan algunos puntos del Product Backlog, que serán los que se desarrollen.
  - Se debe marcar el tiempo en el que se va a realizar el Sprint. Lo aconsejado es hacer un Sprint cada dos o tres semanas.
- Sprint Planning Meeting:
  - Se realiza antes de iniciar un Sprint.
  - Se definen y valoran las tareas que se van a realizar en el siguiente Sprint.
  - En cada Sprint se realizarán varias tareas o procesos.
  - Asisten Scrum Master, Product Owner y Scrum Team

Scrum - Funcionamiento

- Daily o Stand-up Meeting:
  - Reuniones diarias durante el Sprint en las que cada miembro del equipo responde a las siguientes preguntas:
    - ¿Qué hice ayer?
    - ¿Qué voy a hacer hoy?
    - ¿Qué ayuda necesito?
  - El Scrum Master determinará la solución de los problemas y complicaciones que surjan.
  - Son reuniones breves, nada de sentarse a charlar y debatir.

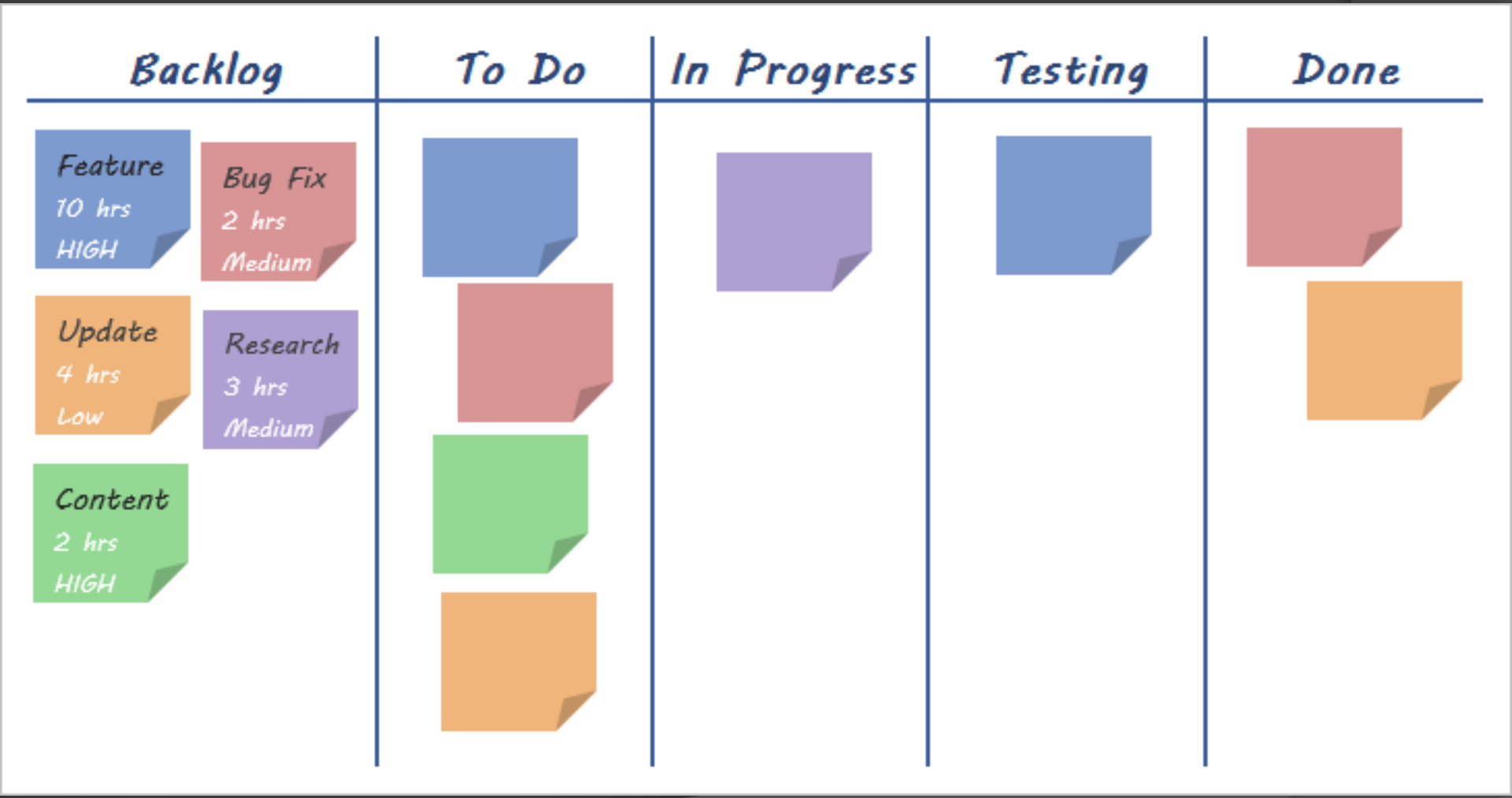
Scrum - Funcionamiento

- Sprint Review:
  - Revisión de lo que ha sido el Sprint terminado.
  - Aquí ya debería haber algo para mostrar al cliente, para que pueda ver y analizar cierto avance.
  - Aquí el cliente ve y valida lo realizado
- Sprint Retrospective:
  - Permite al equipo analizar los objetivos cumplidos, si ha habido errores, verlos y tratar de no cometerlos de nuevo en el futuro.
  - Se analiza si hay que implementar mejoras.
  - El Scrum Team y Scrum Master valoran el Sprint

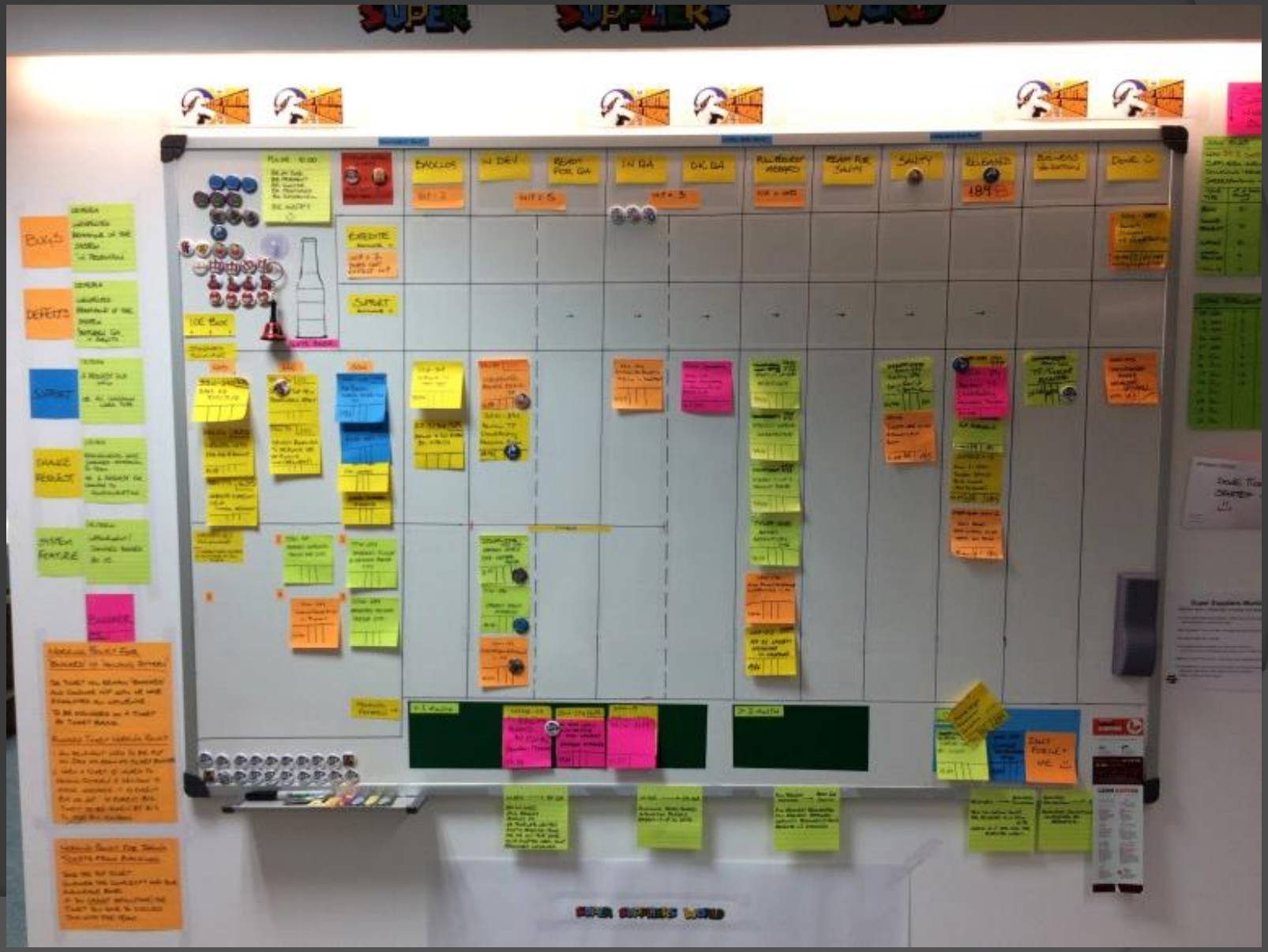
Metodologías Ágiles

- Introducción
- Manifiesto Ágil
- Principales metodologías ágiles:
  - Scrum
  - Kanban
  - eXtreme Programming

Kanban


Backlog
To Do
In Progress
Testing
Done
Feature
10 hrs
HIGH
Bug Fix
2 hrs
Medium
Update
4 hrs
Low
Research
3 hrs
Medium
Content
2 hrs
HIGH

Kanban


SUPER
SUPPLIERS
WORLD
BACKLOG
IN DEV
READY FOR QA
IN QA
DONE QA
READY FOR SAFETY
SAFETY
BLOCKED
DONE
EXECUTE
SUPPORT
BUGS
DEFECTS
CHANGE REQUEST
SYSTEM FAILURE
SUPER SUPPLIERS WORLD

Kanban

- Inventado por Toyota para mantener el nivel de mejoras.
- Consiste en ir etiquetando con tarjetas cada uno de los procesos que se deben llevar a cabo.
- Kanban = tarjetas visuales.
- Fácil de usar e implementar.
- El equipo se unirá y trabajará a la par en el desarrollo.

Kanban - Principios

- Garantía de Calidad:
  - Promueve la calidad antes que la velocidad.
  - Un producto bien hecho a la primera es más rápido que uno mal hecho al que se tienen que dedicar más horas para arreglarlo.
  - Todo debe salir bien desde el principio, con poco o nulo margen de error.
- Desperdicios:
  - Solo se debe hacer lo necesario y requerido.
  - Evita todo lo extra, superficial o innecesario.
  - Ofrece mayor calidad, y mejores tiempos y costos.

Kanban - Principios

- Mejora continua:
  - Permite ir mejorando en los procesos a medida que se utiliza.
  - Puede ser utilizado no solo en desarrollo software sino en cualquier tipo de procesos.
- Es flexible:
  - No necesitamos seguir una línea de trabajo
  - Podemos adelantarnos a un proceso que queramos hacer o que sea más prioritario.
  - Esto hace que sea una metodología dinámica y que permita resolver problemas que surjan de improviso.

Kanban - Estrategia

- Se necesita un tablero donde poder poner unos post-it o elementos similares. Se puede utilizar también herramientas online como Trello.
- Una vez que tengamos el tablero, deberemos tener en cuenta:
  - El flujo de trabajo.
  - Las fases del ciclo de producción.
  - Stop Starting, start finishing.
  - Tener un control.

Kanban - Estrategia

- Definir el flujo de trabajo:
  - El tablero deberá ser dividido dependiendo de las tareas, las fases o proyectos que tengamos.
  - El tablero deberá estar a la vista de todo el mundo.
  - Se trata de que en las columnas definamos los estados por los que pueden pasar las tareas, como por ejemplo:
    - Por hacer | Haciendo | Probando | Hecho.
      - Se debe especificar la definición de “Hecho”
  - Las tareas estarán en los post-it que iremos añadiendo al tablón.
  - Cada miembro deberá asignarse tareas del tablero.
  - Es necesario estar pendiente de los procesos e ir actualizándolo para que todo el equipo sepa en qué estado se encuentra cada tarea en cada momento.

Kanban - Estrategia

- Fases del Ciclo de Producción:
  - Es necesario que las tareas estén bien divididas para que se pueda agilizar y no se queden estancadas con demasiada duración.
  - Habrá que poner la estimación en horas de cada tarea en las tarjetas.
  - Si se ha agotado el tiempo y la tarea no ha sido finalizada, se deberá determinar por qué no se ha terminado, o se avanzará a otra fase.

Kanban - Estrategia

- Stop starting, start finishing:
  - No se empieza una tarea hasta que no se ha terminado otra.
  - La idea es tener un alto porcentaje de tareas terminadas, y no muchas tareas empezadas pero sin completar.
  - Parece una idea muy obvia pero la realidad es que es algo muy importante que muchos equipos no respetan aún cuando es algo fundamental.

Kanban - Estrategia

- Tener un control:
  - Se debe controlar el flujo de trabajo.
  - La idea es que los trabajadores tengan un flujo de trabajo constante y no se queden parados cuando hayan finalizado una tarea.
  - Permite que se pueda trabajar con varios proyectos a la vez. Cuando parte del equipo termine sus tareas, puede ponerse con el otro proyecto mientras termina el resto del equipo.
  - Se trata de no provocar interrupciones a cada momento
  - Si necesitas control, puedes ir almacenando las tarjetas

Metodologías Ágiles

- Introducción
- Manifiesto Ágil
- Principales metodologías ágiles:
  - Scrum
  - Kanban
  - eXtreme Programming

eXtreme Programming

- Es la metodología ágil más destacada.
- Gran capacidad de adaptación a cualquier tipo de imprevisto que surja.
- No se trata de mantener los requisitos desde el principio sino que éstos vayan cambiando o evolucionando a medida que avance el proyecto.
- Es mejor adaptarse en el proceso que iniciar con todo definido.
- Es una combinación de las demás metodologías, solo que se utilizan como se necesita en el momento.

eXtreme Programming - Valores

- Los principios básicos o valores de la metodología de programación extrema son:
  - Comunicación.
  - Simplicidad.
  - Retroalimentación.
  - Valentía.
  - Respeto.

eXtreme Programming - Valores

- Comunicación:
  - Constante con el cliente:
    - Actualizaciones, nuevas ideas, soluciones a problemas o problemas en sí.
  - El propio código fuente, con variables amigables y comentarios.
  - Documentación de lo necesario y/o extenso.
  - Comunicación entre programadores constante por el trabajo en parejas.
  - Todo debe ser comunicado.

eXtreme Programming - Valores

- Simplicidad:
  - Se intenta simplificar todo al máximo (diseño, líneas de código).
  - Se comenta el código para evitar hacer documentación extra (solamente la necesaria).
  - Nombres amigables en variables, métodos y clases, para que sea comprensible por otros desarrolladores y para que cuando una persona nueva se incorpore al proyecto, tenga una rápida adaptación.

eXtreme Programming - Valores

- Retroalimentación:
  - Programación por periodos cortos de tiempo.
  - El cliente está involucrado y ayuda con la retroalimentación.
  - Las pruebas unitarias también ayudan a ver la calidad de nuestro código de manera rápida y eficaz.
  - En proyectos grandes sin pruebas unitarias, cambios importantes pueden dar lugar a una cantidad de errores que hace que volver a empezar de cero sea una opción a considerar.

eXtreme Programming - Valores

- Valentía:
  - Dar solución a los problemas a los que nos enfrentemos como programadores.
  - Eliminar código que nos ha llevado tiempo hacer y que ofrece una funcionalidad que el cliente ya no quiere.

eXtreme Programming - Valores

- Respeto:
  - Muy importante para la buena comunicación entre los miembros del equipo.
  - No denigrar a nadie, no ofender.
  - Una autoestima alta garantizará un trabajo más eficiente.
  - Se debe respetar todo, incluso el código fuente, las modificaciones, los fallos obtenidos, los problemas o las soluciones a los problemas.

XP - Características

- Las características son las siguientes:
  - Tipo de desarrollo Iterativo e Incremental.
  - Pruebas unitarias.
  - Trabajo en equipo
  - Alguien del equipo trabaja con el cliente.
  - Corrección de errores.
  - Reestructuración del código.
  - El código es de todos.
  - Código simple es la clave.

XP - Equipo

- Programador: código y pruebas unitarias.
- Tester: hace las pruebas funcionales, se las comunica al cliente, y el resultado al equipo.
- Tracker: tareas de seguimiento, comparar estimaciones y resultados.
- Entrenador: responsable del proyecto, guía.
- Gestor: el líder más alto, vínculo entre cliente y programadores.
- [Extra] Consultor: externo al proyecto, para ayudar en la solución de problemas.