



METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Ing. Moisés Álvarez Huamán

¿Qué es una metodología de desarrollo del software?

- ❑ Es un "estilo" de hacer las cosas con rigurosidad y basados en normas.
- ❑ ¿Cómo se hace?... ¿Con quién hay que entrevistarse? ¿Qué debo preguntar? ¿Cómo es el informe que debo escribir? ¿Quién lo va a leer? ¿Qué va a hacer con él? ¿Cuánto debo tardar? ¿Cuándo sé que he terminado?
- ❑ Algunas metodologías están escritas en innumerables paginas de lectura. Otras, se describen en apenas unas pocas páginas. Algunas metodologías intentan describirlo todo al detalle. Otras son más genéricas



¿Qué metodología debo utilizar?

- ☺ Hay una serie de metodologías que solemos llamar tradicionales propuestas casi todas ellas con anterioridad a los años 90 que pretendían ayudar a los profesionales indicando pautas para realizar y documentar cada una de las tareas del desarrollo del software.
- Así pues, los pasos que sugieren para llevar a cabo cada tarea, aunque bien intencionados, están cargados de burocracia, reiteraciones, ambigüedades... No suelen tener en cuenta cosas como la calidad, la satisfacción, la competitividad, los beneficios.
- Fueron metodologías creadas en los años 70-80 pensando en los negocios de los años 50.



¿Qué metodología debo utilizar?

- ▶ Las metodologías surgidas desde los 90 hasta aquí suelen tener otra mentalidad... una cierta agilidad. Siendo conscientes de lo cambiante y amplio que es el mundo del software, una metodología debe ser lo suficientemente precisa como para que todo el mundo la pueda seguir y sea de utilidad como pauta común, pero también debe ser lo suficientemente adaptable como para poder aplicarse en distintos proyectos, y lo suficientemente sencilla como para que no resulte muy gravosa su utilización, pero lo suficientemente completa y compleja como para que la utilización por parte del equipo sea provechosa... En una palabra: agilidad
-



¿Qué metodología utilizar?

- ▶ Pues depende:
 - ✓ Si formas parte de un equipo de desarrollo en un proyecto grande y **te toca decidir** qué metodología hay que utilizar significa que tienes un puesto de responsabilidad. Escoge una metodología moderna, bien definida, que dé respuesta a las necesidades del proyecto.
 - ✓ Si formas parte de un equipo de desarrollo en un proyecto grande y no ocupas un puesto de responsabilidad, no deberías decidir qué metodología utilizar: **alguien lo decidirá por ti**. Si nadie toma esa decisión... ¡Mucho ánimo!... el proyecto en el que estás involucrado está destinado al fracaso.
 - ✓ Si formas parte de un equipo pequeño en un proyecto pequeño, lo mejor es **consensuar** la metodología a utilizar. Incluso, combinar buenas ideas de más de una.
-





¿Y qué metodologías hay?

Entre las metodologías tradicionales podemos citar:

- Desarrollo de sistemas de Jackson System Development (JSD). De los años 80. propone una metodología lineal.
 - Ingeniería de la Información Methodology. De los 80 también.
 - Structured System Analysis and Design Method (SSADM). También de los 80. Muy popular en Europa, ya que tiene su origen el Reino Unido.
 - Nuestra querida metodología MÉTRICA, promovida por el Ministerio de las Administraciones Públicas. Es un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software
 - Algunas, como las dos primeras (Jackson, Ingeniería de la información), tienen un interés principalmente histórico. Otras, como SSADM o MÉTRICA, tienen cierta vigencia, en especial en lo que concierne a proyectos públicos.
-

Entre las metodologías modernas -unas más, otras menos- se puede destacar

- ❖ Rapid Application Development (Desarrollo rápido de aplicaciones - RAD).
- ❖ Scrum: Es un tipo ingeniería de software, es iterativo e incremental; es un método ágil de software.
- ❖ Extreme programming. (Programación extrema - XP) ideada por Kent Beck, es una metodología ligera, eficiente, con bajo riesgo, flexible, predecible y divertida para desarrollar software.
- ❖ Rational Unified Process. (Proceso Racional Unificado - RUP) es una metodología para la ingeniería de software es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental
- ❖ Agile Unified Process. (Proceso Ágil Unificado - AUP)
- ❖ Hay muchas más, pero quizá éstas sean las más populares en el omento de escribir estas líneas. Personalmente recomendaría un cierto conocimiento general de al menos las cuatro últimas (Scrum, XP, RUP y AUP)



DIFERENCIAS ENTRE METODOLOGÍAS ÁGILES Y TRADICIONALES O PREDICTIVAS

METODOLOGÍA ágil

- ▶ Acuerdo Iterativo,
adaptativo a los
cambios

METODOLOGÍA TRADICIONAL

- ▶ Búsqueda de un
acuerdo inmutable
previamente
negociado



ágil

tradicional

Agilismo

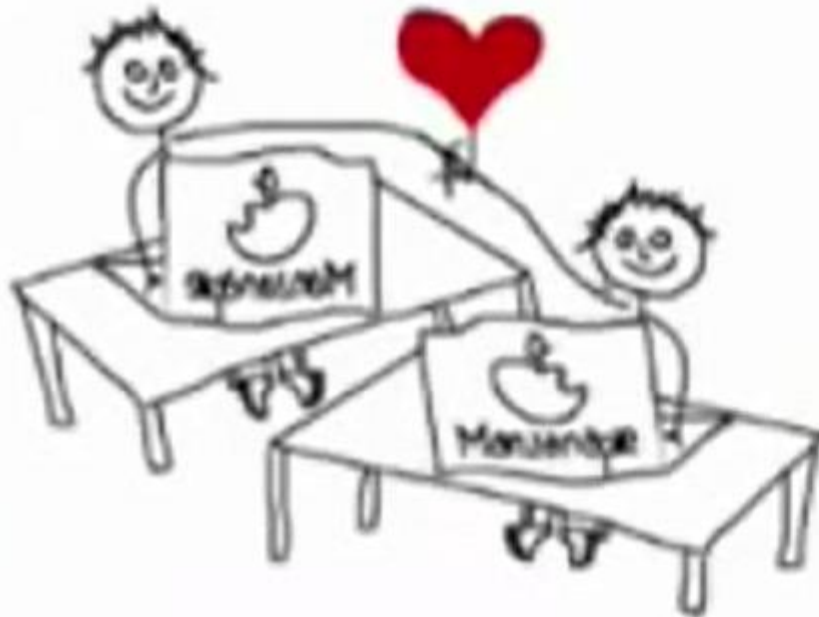


Vs

Predictibilidad



Individuos e iteraciones



Prioridad:

- Calidad profesional del equipo
- Entrega temprana y continua



Procesos y herramientas

Argumento: deben servir de ayuda pero no pueden ser el objetivo

Software Funcionando



Prioridad:

- Satisfacción del cliente
- Aportar valor al negocio



Documentación extensiva

Argumento: debe servir de complemento pero no significar un impedimento

Colaboración con el cliente



Prioridad:

- Participación del cliente
- Comunicación directa y continua



Negociación contractual

Argumento: solo el cliente conoce lo otorga verdadero valor al negocio

prioridad

- ▶ Aceptar los cambios de requerimientos.
- ▶ Ventaja competitiva para el negocio.

Seguir un plan

- ▶ Argumento: el cliente no está realmente seguro hasta que no pruebe el software.




Metodologías más usadas

- ❑ Scrum 58%
- ❑ eXtreme Programming
- ❑ RUP (Proceso unificado racional)
- ❑ TDD (Desarrollo guiado por pruebas)



NO son metodologías de desarrollo del software

- ✓ La "Programación estructurada" o la "Programación Orientada a Objetos" son *paradigmas* o modelos de *programación*. Indican pautas de comportamiento en los sistemas de programación... no tienen nada que ver con el ciclo de vida del software ni la manera en la que debe realizarse cada tarea para un proyecto concreto... así pues... **NO SON METODOLOGÍAS.**
-
- 

NO son metodologías de desarrollo del software

- ▶ Los términos "Ciclo de vida en espiral", "Incremental", en "Cascada", con "prototipo", etc... Indican esquemas generales de organización en las tareas del ciclo de vida, unas con respecto a otras y con respecto a otros aspectos como el tiempo, los requisitos o el riesgo. Actualmente se denominan "PATRONES" del ciclo de vida del software, aunque antaño fueron denominados simplemente distintos "Ciclos de vida". Indican ideas estructurales sencillas en el proceso de desarrollo, y no la manera en la que debe realizarse cada tarea del ciclo para un proyecto concreto... así pues... **NO SON METODOLOGÍAS**

NO son metodologías de desarrollo del software

- ▶ El lenguaje UML (Unified Modeling Language) es un gran logro de la ingeniería. Aún con sus carencias, es algo muy importante: un lenguaje común para que todos los profesionales del desarrollo de sistemas -de software o no- expresen sus ideas... pero el UML no le indica a nadie la manera de realizar las cada tarea en un proyecto concreto: tan solo es una herramienta para expresar ideas... así pues... **NO ES UNA METODOLOGÍA**. Sin embargo, algunas metodologías de las que hemos comentado, como RUP o METRICA hacen referencia a UML como herramienta para expresar ideas
-



Tarea

- ▶ Organizarse en grupos para exponer las diferentes metodologías; asimismo aplicarlos para el caso presentado.
 - 1. Metodología: Ingeniería de la Información 28/08/17
 - 2. Structured System Analysis and Design Method (SSADM). (Metodología Estructurada) 11/09/17
 - 3. Scrum: 25/09/17
 - 4. Extreme programming. (Programación extrema - XP) 02/10/17
 - 5. RUP (Proceso Unificado Racional) 16/10/17
 - 6. Crystal Clear 06/11/17
 - 7. Metodología TDD (Análisis de sistemas basado en Test) 20/11/17
 - 8. DSDM FDD 20/11/17
 - 9. Kanban 20/11/17
-

