



# UNIVERSIDAD DE GRANADA

METODOLOGÍAS DE DESARROLLO ÁGIL

## **DSDM: Dynamic System Development Method**

*Un método iterativo y creciente*

*Pedro Bonilla Nadal  
David Infante Casas  
Laura García Gómez  
Antonio Martín Ruiz  
Juan Ocaña Valenzuela*

Curso 2019-2020

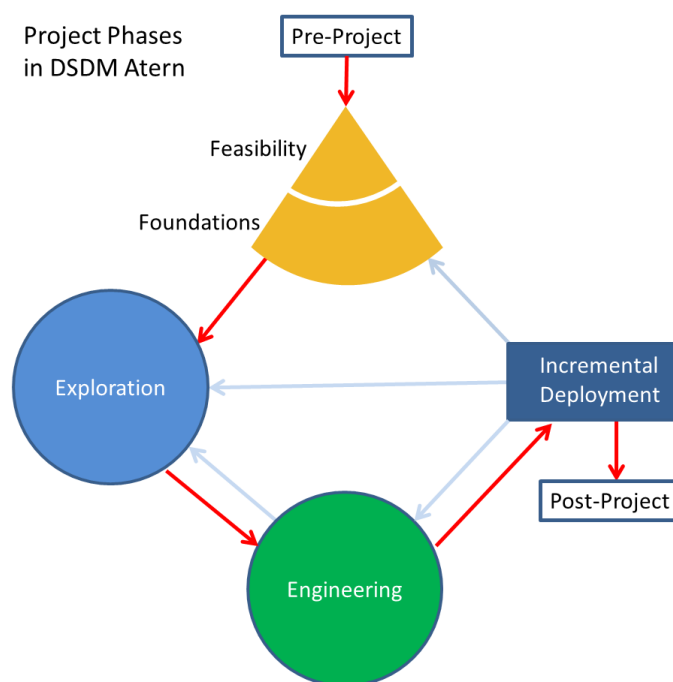
18 de noviembre de 2019

# Índice

<b>1. Introduccion</b>	<b>2</b>
<b>2. Fases del desarrollo</b>	<b>3</b>
<b>3. Principios</b>	<b>3</b>
<b>4. Roles</b>	<b>3</b>
<b>5. TimeBoxing</b>	<b>3</b>
<b>6. Conclusiones</b>	<b>4</b>

# 1. Introducción

El método de desarrollo de sistemas dinámicos (Dynamic Systems Development Method, DSDM) es un framework para el desarrollo de software de forma ágil creado en 1995 por lo que actualmente se conoce como el Consorcio DSDM. Determina fases, subfases y principios que permiten a equipos de desarrollo trabajar de forma eficiente. Fue una metodología muy popular durante los años 90 y comparte cualidades con otros métodos de desarrollo ágil como SCRUM o Programación Extrema. Es por esto que no es una metodología muy explotada a día de hoy, pues muchos equipos prefieren el uso de SCRUM. DSDM se divide en tres fases: preproyecto, ciclo de vida del proyecto y postproyecto. A su vez, el ciclo de vida se subdivide en cuatro fases, viabilidad, exploración, ingeniería y despliegue.



DSDM es el método ágil más antiguo, lanzado en 1995 y es el único método ágil para centrarse en la gestión de proyectos ágiles. Arie van Bennekum representó a DSDM en el lanzamiento de Agile Alliance y su Agile Manifesto en 2001. DSDM ha operado principalmente en el ambiente corporativo donde demuestra consistentemente su habilidad para trabajar exitosamente dentro y complementar los procesos corporativos existentes. Practicando el desarrollo evolutivo, la última versión de DSDM (Atern) incorpora esas mejoras.

Algunas características de DSDM incluyen al usuario como componente activo durante el desarrollo, equipos bien preparados y lanzamiento de prototipos frecuente. También se usa la técnica «Timeboxing», de la cual hablaremos más adelante, en la que se establecen intervalos en los que se liberarán los prototipos. Estos intervalos son muy similares a los sprints de SCRUM.

## 2. Fases del desarrollo

Para explicar las fases de desarrollo, de entre las diversas variedades de DSDM, explicaremos en particular las del DSDM-Atern. Atern difiere de los enfoques ágiles más comunes en que abarca todo el ciclo de vida del proyecto y no sólo el desarrollo de software (donde prevalece Scrum). Incorpora disciplinas de gestión de proyectos y proporciona mecanismos para asegurar que los beneficios del proyecto sean claros, que la solución propuesta sea factible y que existan bases sólidas antes de que se inicie el trabajo detallado. .

- Pre-proyecto: Inicio del proyecto, acordando los Términos de Referencia para el trabajo.
- Viabilidad: Normalmente una fase corta para asegurar cual es el valor del proyecto y resaltar su modelo de negocio.
- Fundaciones: Fase clave para asegurar que el proyecto se entienda y defina lo suficientemente bien como para que el alcance se pueda basar en un alto nivel y los componentes y estándares tecnológicos se acuerden, antes de que comience la actividad de desarrollo.
- Exploración: Fase de desarrollo iterativo durante la cual los equipos amplían los requisitos de alto nivel para demostrar la funcionalidad.
- Ingeniería: Fase de desarrollo iterativo en la que la solución se diseña para que pueda desplegarse para su lanzamiento.
- Despliegue: Para cada incremento (conjunto de cuadros de tiempo) del proyecto se pone a disposición la solución.
- Post proyecto: Evalúa los beneficios acumulados.

Las fases de exploracion y la fase de ingenieria normalmente se mezclan, dado que el método es flexible, permitiendo a los equipos que lo adapten para ser más adecuado a su situación. Como podemos ver estas fasen comprenden dentro de ellas

## 3. Principios

## 4. Roles

## 5. TimeBoxing

La gestión de proyectos tradicional utiliza *miletones* para acordar un asset a realizar para un proyecto determinado en un punto en el tiempo. Si bien los *miletones* funcionan lo suficientemente bien, TimeBoxing es una técnica mucho más poderosa para lograr el mismo

resultado. Una TimeBox es un intervalo, normalmente no más largo de 6 semanas, donde un conjunto dado de tareas debe ser logrado. El motivo de la duración relativamente baja de las TimeBox es el hecho de que los seres humanos dan estimaciones mucho más precisas en el futuro cercano, con un pequeño conjunto de tareas. En las estimaciones de un futuro lejano, en el que intervienen grandes conjuntos de tareas, se acaban produciendo grandes errores.

Una TimeBox puede contener varias tareas, y al final de esta se ha de buscar entregar un producto. Los *miletones* también sufren de tener un entregable fijo, mientras que los TimeBoxes están sujetas a cambios, ya que se definen las tareas, no necesariamente el entregable, que puede cambiar si cambios de prioridades durante la iteración del TimeBox, lo que permite una respuesta rápida a necesidades del negocio. En resumen: DSDM deja de lado la funcionalidad a favor de la entregar a tiempo.

Podemos observar como características en el uso de timeboxing en combinación con las fases del proyecto DSDM:

1. Los Timeboxes pueden ser de diferente longitud.
2. Varias de las mismas fases de DSDM pueden ejecutarse inmediatamente después de una la TimeBox de la misma fase ha terminado.
3. Las TimeBoxes paralelas son posibles, incluso las TimeBoxes complementarias son permitido.
4. Diferentes fases de DSDM pueden ser realizadas en una TimeBox al mismo tiempo.
5. Los Timeboxes pueden ser anidados.

## 6. Conclusiones

---

TODO: hola, podemos añadir de la página <https://www.methodsandtools.com/archive/dsdmatern.php> cuales son los principios, los roles y los productos.

---

## Referencias

- [1] Dynamic Systems Development Method (DSDM)  
[http://www.students.science.uu.nl/~slegt001/me/final\\_5767202\\_Slegten.pdf](http://www.students.science.uu.nl/~slegt001/me/final_5767202_Slegten.pdf)

- [2] Dynamic System Development Method  
[https://files.ifi.uzh.ch/rerg/arvo/courses/seminar\\_ws03/14\\_Voigt\\_DSMD\\_Ausarbeitung.pdf](https://files.ifi.uzh.ch/rerg/arvo/courses/seminar_ws03/14_Voigt_DSMD_Ausarbeitung.pdf)
- [3] New Directions on Agile Methods:A Comparative Analysis  
[http://secure.com.sg/courses/ICT353/Session\\_Collateral/TOP\\_03\\_ART\\_06\\_ARTICLE\\_ABRAHAMSSON\\_New\\_Directions\\_Agile\\_Methods.pdf](http://secure.com.sg/courses/ICT353/Session_Collateral/TOP_03_ART_06_ARTICLE_ABRAHAMSSON_New_Directions_Agile_Methods.pdf)
- [4] Agile software development methods.  
<http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>
- [5] Introduction to DSDM Atern  
<https://www.methodsandtools.com/archive/dsdmatern.php>