UNIVERSIDAD
DE GRANADA

Metodologías de desarrollo ágil

DSDM: Dynamic System Development Method

Un método iterativo y creciente

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

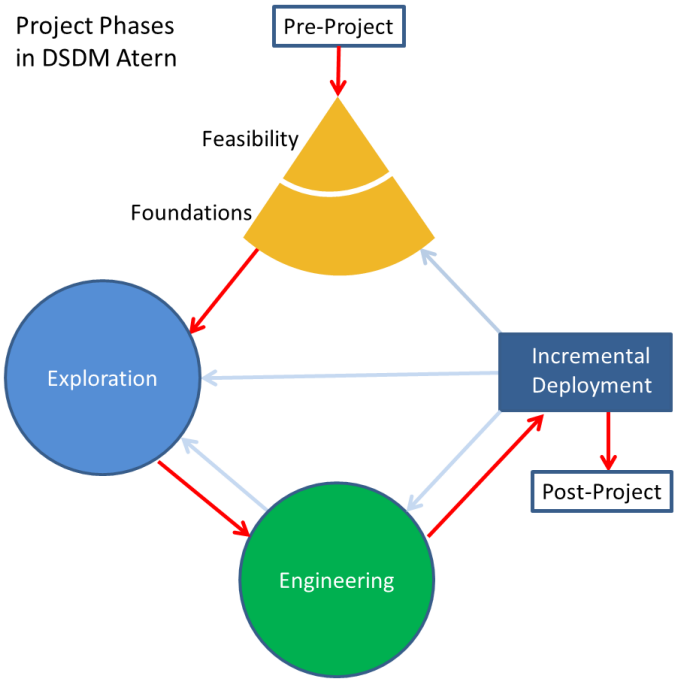
Curso 2019-2020

18 de diciembre de 2019

Índice

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. Esta metodología 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 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 opera principalmente en el ambiente corporativo donde demuestra consistentemente que es una métodología muy válida y permite completar exitosamente los procesos corporativos existentes. DSDM se ha ido actualizando a lo largo de los años, cuya última versión es Atern.

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. Roles

Entre los roles principales de la metodología DSDM, destacan:

- Patrocinador ejecutivo. Es alguien que actúa como el CEO de una empresa. Ofrece recursos al proyecto y establece relaciones con el resto de ejecutivos relacionados al proyecto. También asegura la financiación del proyecto y una eficaz toma de decisiones.
- Visionario. Lo lleva a cabo alguien responsable de la detección inicial de requisitos para comenzar el proyecto, el cual también actúa como product manager. Motiva al equipo y les ofrece recursos de apoyo. Coordina los objetivos del proyecto con los de negocio. También ayuda con el diseño y las sesiones de review.
- Usuario embajador. Es alguien con un gran conocimiento sobre el público objetivo y ofrece feedback al equipo de desarrollo. Se trata de un consultor externo. Comunica al equipo de desarrollo con el usuario. Trabaja en la detección de requisitos, diseño y sesiones de review. Ofrece visión de negocio para todas las decisiones que se toman. Ayuda al diseño y al testeo mostrando situaciones de negocio. Asegura que el producto final es el esperado.
- Project manager. Es responsable de que el proyecto se complete con éxito y el día a día del mismo. Realiza una planificación de alto nivel. Monitoriza el progreso, disponibilidad de recursos, configuración del proyecto, controla riesgos y problemas.
- Desarrolladores. Se encargan de la ingeniería y codificación del proyecto. No importa si el miembro del equipo es programador, analista o diseñador, todos se clasifican como desarrolladores.
- Testers. Su trabajo es probar y comprobar que el proyecto no contiene fallos y funciona correctamente. Define escenarios de prueba para el proyecto. Ofrece los datos de prueba obtenidos al líder del equipo.

3. 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. Es vital que durante esta fase el equipo satisfaga la mayoría de roles mencionados en su sección.
- Ciclo de vida del proyecto. Está formado por las siguientes fases:
    - Viabilidad. Normalmente una fase corta en la que se realiza un estudio para asegurar cual es el valor del proyecto, coste y beneficio 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. Durante esta fase se especifican los requisitos funcionales y no funcionales. Además los roles principales son Project Manager y CEO.
    - Exploración. Los líderes de proyecto y el equipo de desarrollo trabajan junto con el usuario embajador para asegurar un correcto desarrollo, detallando los requisitos previamente establecidos.
    - Ingeniería. Fase de desarrollo iterativo en la que la solución se diseña para que pueda desplegarse para su lanzamiento. El consultor externo se involucra para asegurar que las necesidades de los usuarios estén cubiertas.
    - Despliegue. Se presenta el sistema a los usuarios y se incorpora el feedback durante varias iteraciones.
- Post proyecto. Evalúa los beneficios acumulados.

Las fases de exploración y la fase de ingeniería normalmente se mezclan, dado que el método es flexible, permite a los equipos que lo adapten trabajar dependiendo de la situación en la que se encuentren.

4. Principios

El método DSDM se usa para procesos de desarrollo ágil. Para mostrar como DSDM se relaciones es fundamental comprender los principio de DSDM y como estos se completan con las filosofías ágiles.

Los siguientes nueve principios son fundamentales para poder entender cual es la filosofía de desarrollo detrás de DSDM. En las descripciones de DSDM, si bien se explicita que no es necesario seguir al pie de la letra todas las fases del desarrollo, pues es un método adaptable a cada situación, si que explicitan que no seguir alguno de estos principios si que grava fuertemente el proceso de desarrollo.

1. La participación activa del usuario. Es imperativo este punto. Como bien sabemos uno de los principales aspectos de las metodologías ágiles es ser user-focused. El primer principio se considera el más importante, porque la participación de los usuarios a lo largo de todo el proyecto reduce efectivamente los errores en términos de percepción de los usuarios, y por lo tanto, reduce los costes de error.

   En lugar de trabajar con un gran número de usuarios, las directrices de DSDM recomiendan trabajar con un pequeño y selecto grupo de usuarios continuamente, evitando así descubrir defectos solamente al final del proyecto en las reuniones con el usuario.

2. Los equipos deben tener poder para tomar decisiones. Esto ayuda a proceder lo más rápidamente posible eliminando los costes de trasmisión y la espera de las comunicaciones. Estos retrasos y fricciones debido a la comunicación obligatoria entre equipo y gerentes deben ser evitadas. Solicitar la autorización de recursos, incluso modestos, o cambios sencillos en los requisitos ralentizarán un proyecto de forma significativa.

   Del mismo modo los usuarios y otra gente involucrada en el proceso deben tener una autoridad, existente pero limitada para:

   - Crear requisitos.
   - Qué funcionalidad necesita estar en un incremento dado
   - Priorización de requisitos y características
   - Detalles de la solución técnica

3. Concentrarse en el lanzamiento y los límites de entrega. Las entregas frecuentes de los resultados aseguran que los errores se detecten rápidamente, son fáciles subsanados y es más fácil de detectar la fuente del error. Esto se aplica tanto al código del programa como a la así como a documentos como requisitos o modelos de datos.

4. Considerar como criterio para el software su validez en el mercado.Como el nombre del marco de trabajo de DSDM sugiere, su esfuerzo principal es entregar software que es lo suficientemente bueno para resolver la necesidad del negocio y preocuparse de cualquier mejora en una iteración posterior.

   DSDM no promueve la escritura ad-hoc sino que sugiere que se satisfagan primero las necesidades del negocio, y que se usen TimeBoxes (explicadas posteriormente) para refactorización y actividades relacionadas en una iteración posterior. Para ello es necesario que cosas como la refactorización, ingeniería de diseño y mejora de características se consideren parte fundamental del proyecto. También es es crucial identificar en las primeras etapas del proyecto las cuestiones clave, que requieren un diseño robusto.

5. El desarrollo iterativo e incremental. Para reducir la complejidad del proyecto es necesario descomponerlo en pequeños paquetes de características, y en cada iteración se añaden nuevas características hasta que todos los requisitos del negocio se cumplen.

   Este principio requiere aceptar el hecho de que que cualquier sistema de software está sujeto a cambios, uno de los principios universales de las metodologías ágiles. Este principio puede introducirse fácilmente incluso al principio de un proyecto, ya que las especificaciones y otros resultados pueden ser producido de manera iterativa también.

6. Todos los cambios durante el desarrollo deben ser reversibles. Para asegurar la capacidad de respuesta al cambio se requiere que las configuraciones del sistema cambien durante el desarrollo del proyecto debido a cambios en las prioridades de los requisitos.

7. Los requisitos se definen a alto nivel. Para evitar limitar el grado de libertad con el que se pueden modificar los requisitos durante el proceso de desarrollo. Estos requisitos se deben acordar durante el estudio de mercado del problema y después ser congelados.

8. Las pruebas están integradas a lo largo de todo el ciclo de vida. Muchos métodos de desarrollo, normalmente de carácter tradicional, requieren pruebas demasiado tarde como para que puedan tener un impacto poco costoso. DSDM requiere pruebas al principio del proceso de desarrollo.

9. Enfoque colaborativo y cooperativo. Hay que evitar la separación y fomentar la colaboración entre el personal técnico y el personal empresarial de un proyecto porque la cooperación es fundamental para llevar a cabo un proyecto de DSDM. Sin una atmósfera de confianza y honestidad será difícil obtener requisitos y un feedback conveniente.

5. Valores

El Manifiesto Ágil desarrolla 4 principios fundamentales. En esta sección vamos a ver como DSDM se complementa con la filosofía ágil.

Recordatorio del Manifiesto Ágil

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.

Analicemos uno a uno estos valores.

5.1. Individuos

Muchas metodologías de organización modernas expresan la importancia de considerar como individuos a los participantes del equipo. Los principios de DSDM 1 y 2 enfatizan este concepto, dando a entender que tanto los integrantes del equipo son personas con capacidad de tomar decisiones de modo correcto y obligando una implicación de los usuarios activa, puesto que su opinión es importante.

De modo menos obvia, en los principios 3 y 5 también está escondido este concepto, al necesitar un ecosistema laboral sano para poder ser llevados a cabo.

5.2. Software Funcionando

El principio 4 recuerda que el éxito del software depende de su existo comercial, o al menos de su éxito de cara al usuario. Los principios 3 y 5 también hacen énfasis en este concepto al querer software funcionando de manera continua. Por último el principio 7 es fundamental para la comprensión de que es el Software Funcionando.

5.3. Colaboración

La colaboración es más significativa dado el principio 9 porque DSDM reacciona al cambio y cuando éste se produce, independientemente de que forme parte de un proyecto o de la vida personal, se requiere una buena cantidad de comunicación para resolver la implicación del cambio. Poder hablar sin barreras ayuda mucho. La necesidad de interpretar las declaraciones de otras personas y el error de malentendidos se reduce en gran medida. Desafortunadamente, la colaboración no es fácil, ya que los humanos tienden a tomar decisiones egoístas, especialmente si se espera una recompensa inmediata. La creación de un entorno en el que la colaboración funciona de la forma más fluida posible es una tarea de gestión y liderazgo.

5.4. Respuesta al cambio

Aceptar el cambio y responder a él tiene sus raíces en el principio 6 y parcialmente en el principio 7. La gestión de las demandas cambiantes del negocio se percibe comúnmente como

la diferencia más prominente de los métodos ágiles y tradicionales. DSDM maneja el cambio de varias maneras, principalmente a través de la priorización.

6. TimeBoxing

La gestión de proyectos tradicional utiliza *miletones* para acordar un asset a realizar para un proyecto determinado en un determinado momento. 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 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 cambian las 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 la TimeBox de la misma fase si ha terminado.
3. Las TimeBoxes paralelas son posibles, incluso las TimeBoxes complementarias son permitidas.
4. Diferentes fases de DSDM pueden ser realizadas en una TimeBox al mismo tiempo.
5. Las Timeboxes pueden ser anidadas.

Referencias

[1] Dynamic Systems Development Method (DSDM)
`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

[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

[4] Agile software developmentmethods.
http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf

[5] Introduction to DSDM Atern
https://www.methodsandtools.com/archive/dsdmatern.php