



UNIVERSIDAD NACIONAL DE LA MATANZA

ESCUELA DE POSGRADO

MAESTRIA EN INFORMÁTICA

TESIS DE MAESTRÍA

Título de Tesis:

Para un proyecto software: Metodología Ágil DSDM vs. SCRUM

Autora: Lic. Hebe Anadón

Directora: Dra. Alicia Mon

Co-Directora: Dra. Bettina Donadello Anadón

Buenos Aires, 3 de setiembre de 2014

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

INDICE

| | |
|---|----|
| CAPITULO 0 – AGRADECIMIENTOS | 13 |
| CAPITULO I - INTRODUCCION | 15 |
| Resumen..... | 15 |
| Área de trabajo..... | 17 |
| Necesidad de una metodología para el Desarrollo de Software | 17 |
| ¿Qué es una Metodología? | 18 |
| Descripción del Problema..... | 18 |
| Importancia del problema | 18 |
| Objetivos, Motivación e Hipótesis..... | 19 |
| Objetivos..... | 19 |
| Motivación..... | 19 |
| Hipótesis de la Tesis | 20 |
| Dimensiones de análisis | 21 |
| Breve esbozo de la Solución | 21 |
| CAPITULO II - ESTADO DE LA CUESTION | 23 |
| Metodologías de Desarrollo de Software..... | 23 |
| Breve reseña histórica..... | 23 |
| Modelos de Desarrollo Agiles | 24 |
| Resumen de las Características de un Método Agil..... | 28 |
| Metodología Scrum..... | 30 |
| Extreme Programming (XP) o Programación Extrema (PE)..... | 30 |
| Valores originales de la programación extrema | 31 |
| Scrum..... | 33 |
| Metodología DSDM | 38 |
| Origen de DSDM..... | 38 |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | |
|--|----|
| Creación de DSDM | 39 |
| DSDM Consortium..... | 41 |
| Business Centred Development o Desarrollo Centrado en el Negocio | 42 |
| Delivery Framework o Entrega del Marco | 42 |
| Business Development Framework o Marco de Desarrollo Empresarial..... | 42 |
| e-Business Framework | 43 |
| DSDM Atern | 44 |
| DSDM Agile Project Management - AGILE | 45 |
| DSDM Agile Project Framework | 46 |
| Agile Business Analysts – AgilBA | 47 |
| DSDM Members | 49 |
| Proyecto para el Desarrollo de un Proyecto de Software | 49 |
| Imprescindibles para el desarrollo de un proyecto software | 53 |
| DSDM va más allá..... | 53 |
| Enfoque en el Proyecto o Enfoque en el Producto | 55 |
| Scrum de Scrums - DSDM Atern | 55 |
| Escalabilidad..... | 55 |
| Qué, Cuándo y Cuánto?..... | 59 |
| Scrum - Agile Project Management | 63 |
| El Software no es ingeniería tradicional ni es investigación | 63 |
| Siguen otras opiniones sobre DSDM vs Scrum | 66 |
| ¿Por qué preferir DSDM a Scrum?..... | 66 |
| DSDM Scrum Roadmap..... | 68 |
| Lean y PMBOK..... | 69 |
| Scrum & DSDM Atern: AND, not XOR..... | 71 |
| Compuertas de decisión y etapas de preparación | 73 |
| Decisión de la Jerarquía..... | 73 |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | |
|--|------------|
| Usando MoSCoW para fijar timescales..... | 73 |
| Un rol para Project Management..... | 74 |
| DSDM is Agile's Best Kept Secret?..... | 74 |
| Puntos de interés..... | 75 |
| Los beneficios..... | 75 |
| Defining Agile Quality: Scrum versus DSDM - June 14, 2012..... | 76 |
| Investigaciones sobre Metodologías Agiles | 77 |
| CAPITULO III - PLANTEAMIENTO DEL PROBLEMA..... | 79 |
| ¿Qué es Scrum? | 79 |
| Bases de Scrum..... | 79 |
| Requisitos para poder utilizar Scrum | 79 |
| Desarrollo Iterativo e Incremental..... | 80 |
| Roles de Scrum..... | 83 |
| Métricas para Scrum..... | 85 |
| ¿Qué es DSDM? | 86 |
| Filosofía de DSDM..... | 86 |
| Técnicas básicas | 87 |
| ¿Qué es DSDM Atern? | 94 |
| Características básicas de DSDM Atern..... | 94 |
| Principios de DSDM Atern..... | 95 |
| Atern Products o Productos de Atern | 96 |
| CAPITULO IV – PROPUESTA DE ANALISIS COMPARATIVO | 103 |
| Scrum - Aspectos Fundamentales..... | 103 |
| Personas - Roles de Scrum y skills necesarios | 103 |
| Planificación del producto o proyecto | 110 |
| Planificación de la iteración | 113 |
| Métricas para Scrum..... | 118 |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | |
|--|-----|
| Sprint Review o Revisión de la Iteración - Inspección y adaptación | 119 |
| ¿Por qué son buenas las demostaciones en Scrum? | 120 |
| Sprint Retrospective o Retrospectiva | 121 |
| Beneficios de Scrum..... | 122 |
| DSDM Framework - Características Destacadas..... | 126 |
| Requisitos flexibles | 126 |
| People & Team o Gente y Equipo | 127 |
| Facilited Workshops o Talleres Facilitados | 128 |
| Técnicas básicas | 129 |
| Enumeración de las técnicas básicas que se presentaron en el CAP. III..... | 129 |
| ¿Cuándo usar DSDM?..... | 130 |
| Suitability Checklist o Listado de Control de Aptitudes | 130 |
| Tailoring o Adaptación..... | 130 |
| Usando DSDM con otros Métodos..... | 131 |
| The DSDM Agile Project Framework for Scrum..... | 131 |
| Variables..... | 132 |
| Lifecycle | 133 |
| Roles | 134 |
| Artefactos..... | 135 |
| Techniques..... | 138 |
| DSDM Roadmap..... | 140 |
| Características esenciales de las metodologías DSDM y Scrum..... | 147 |
| Cuadro comparativo entre DSDM y Scrum..... | 148 |
| Consideraciones especiales sobre DSDM y Scrum | 150 |
| Dimensiones de análisis | 150 |
| Propuesta de Análisis Comparativo | 154 |
| Propuesta de Análisis Comparativo - Variables | 155 |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | |
|---|-----|
| Valores pasibles de ser considerados | 156 |
| CAPITULO V - DISEÑO CUALITATIVO | 159 |
| Dos temas a investigar | 159 |
| Respuestas a los cuestionarios del ANEXO II..... | 160 |
| Respuestas al Cuestionario I..... | 161 |
| Respuestas al Cuestionario II | 162 |
| Respuestas al Cuestionario III | 164 |
| Algunas conclusiones extractadas de las respuestas..... | 168 |
| CAPITULO VI - CONCLUSIONES Y FUTUROS TRABAJOS | 169 |
| Conclusiones | 169 |
| Futuros trabajos..... | 172 |
| CAPITULO VII - BIBLIOGRAFIA | 173 |
| ANEXO I - DSDM ATERN..... | 179 |
| ANEXO II – CUESTIONARIOS I, II y III..... | 189 |
| ANEXO III - EMPRESA DESARROLLADORA DE SOFTWARE (Confidencial) . | 199 |
| ANEXO IV - TABLA DE TESIS SOBRE METODOLOGIAS AGILES | 205 |

INDICE DE FIGURAS Y TABLAS

| | | |
|-------|--|----|
| Fig.1 | Manifiesto por el Desarrollo Ágil de Software | 25 |
| | Fuente: <i>Manifiesto por el Desarrollo Ágil de Software</i> < http://agilemanifesto.org/iso/es > | |
| Fig.2 | Logotipo de DSDM Consortium y DSDM Members | 49 |
| | Fuente: Elaboración propia sobre <i>DSDM Members</i> . Why DSDM? (2002) < www.dsdm.org > | |
| Fig.3 | Esquema de los roles en un equipo que combina Scrum con DSDM Atern | 58 |
| | Fuente: <i>Scaling Agile Projects</i> < http://www.mcpa.biz/2012/03/scaling-agile-projects/ > (2012, marzo) | |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | |
|--------|--|-----|
| Fig.4 | Esquema de los roles en 3 equipos combinando Scrum con DSDM Atern Fuente: <i>Scaling Agile Projects</i> < http://www.mcpa.biz/2012/03/scaling-agile-projects/ > (2012) | 59 |
| Fig.5 | Diferentes métodos ágiles pueden utilizarse en diferentes niveles operacionales Fuente: <i>Scrum & DSDM Atern: AND, not XOR</i> < http://abithon.wordpress.com/2012/05/15/scrum-dsdm-atern-and-notxor/ >(2012) | 72 |
| Fig.6 | DSDM ¿es el secreto ágil mejor guardado? Fuente: <i>DSDM is Agile's Best Kept Secret?</i> < http://agilescout.com/dsdm-is-agiles-best-kept-secret/ > | 74 |
| Fig.7 | Proceso Scrum Fuente: <i>Qué es Scrum</i> < http://www.proyectosagiles.org/que-es-scrum > | 80 |
| Fig.8 | Timebox Fuente: <i>Timeboxing DSDM Tour</i> < www.dsdm.org > | 88 |
| Fig.9 | Reglas MoSCoW Fuente: <i>Moscow DSDM Tour</i> < www.dsdm.org > | 89 |
| Fig.10 | Sistema de Gestión de la Configuración del Software Fuente: <i>DSDM Tour</i> < www.dsdm.org > | 92 |
| Fig.11 | Fases del proceso de un proyecto software. Fuente: <i>DSDM Tour</i> < www.dsdm.org > | 93 |
| Fig.12 | Documentos creados Una única vez. Fuente: <i>Mapping documents to DSDM Atern</i> < http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/ > (2012) | 97 |
| Fig.13 | Documentos creados Por Períodos. Fuente: <i>Mapping documents to DSDM Atern</i> < http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/ > (2012) | 98 |
| Fig.14 | Documentos creados Para cada realización externa. Fuente: <i>Mapping documents to DSDM Atern</i> < http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/ > | 99 |
| Fig.15 | Documentos creados Para cada realización interna. Fuente: <i>Mapping documents to DSDM Atern</i> < http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/ > (2012) | 100 |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | |
|--------|--|-----|
| Fig.16 | Documentos creados Si es necesario. Fuente: <i>Mapping documents to DSDM Atern</i> < http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/ > (2012) | 101 |
| Fig.17 | Tarjeta Historia de Usuario Fuente: <i>Objetivos como historias de usuario</i> < http://www.proyectosagiles.org/introduccion-estimacion-planificacion-agil#historias-usuario > | 111 |
| Fig.18 | Product Backlog Fuente: Lista de objetivos / requisitos priorizada (Product Backlog) < http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog > | 111 |
| Fig.19 | Product Burndown Chart Fuente: <i>Gráficos de trabajo pendiente (Burndown charts)</i> < http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts > | 113 |
| Fig.20 | Sprint Burndown Chart Fuente: <i>Gráficos de trabajo pendiente (Burndown charts)</i> < http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts > | 113 |
| Fig.21 | Planificación de la Iteración. Fuente: <i>Planificación de la iteración (Sprint Planning)</i> < http://www.proyectosagiles.org/planificacion-iteracion-sprint-planning > | 113 |
| Fig.22 | Scrum Taskboard Fuente: <i>Ejemplo de uso del tablero o pizarra de tareas (Scrum Taskboard)</i> < http://www.proyectosagiles.org/ejemplo-tablero-pizarra-tareas-scrum-taskboard > | 115 |
| Fig.23 | Construcción del Tablero Fuente: <i>Ejemplo de uso del tablero o pizarra de tareas (Scrum Taskboard)</i> < http://www.proyectosagiles.org/ejemplo-tablero-pizarra-tareas-scrum-taskboard > | 115 |
| Fig.24 | Cuadro de mandos integral para Scrum Fuente: <i>Métricas ágiles y cuadro de mandos integral para Scrum</i> < http://www.proyectosagiles.org/metricas-agiles-cuadro-mandos-balanceado-scrum > | 118 |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | |
|---------|---|-----|
| Fig.25 | Beneficios de Scrum. Fuente: <i>Beneficios de Scrum</i> < http://www.proyectosagiles.org/beneficios-de-scrum#expectativas > | 122 |
| Fig. 26 | Relación Cliente Proveedor Fuente: <i>Beneficios de Scrum</i> < http://www.proyectosagiles.org/beneficios-de-scrum#expectativas > | 122 |
| Fig. 27 | Requisitos flexibles. Fuente: <i>DSDM Tour</i> < www.dsdm.org > | 126 |
| Fig. 28 | Parámetros balanceados. Fuente: <i>The DSDM Agile Project Framework for Scrum</i> (2012) | 132 |
| Fig. 29 | Ciclo de Vida. Fuente: <i>The DSDM Agile Project Framework for Scrum</i> (2012) | 133 |
| Fig. 30 | Tres grupos de roles Fuente: <i>The DSDM Agile Project Framework for Scrum</i> (2012) | 135 |
| Fig. 31 | Artefactos. Fuente: <i>The DSDM Agile Project Framework for Scrum</i> (2012) | 136 |
| Fig. 32 | Timeboxing Aplicado a un Sprint. Fuente: <i>The DSDM Agile Project Framework for Scrum</i> (2012) | 138 |
| Fig. 33 | Triángulo de Hierro Fuente: Elaboración propia | 144 |
| Fig. 34 | Variables de un proyecto Fuente: <i>What is DSDM Atern?</i> < http://www.mcpa.biz/2011/08/what-is-dsdm-atern > (2011, agosto) | 179 |
| Fig. 35 | Ciclo de vida del proyecto Fuente: <i>DSDM Atrn Project Structure</i> < http://www.mcpa.biz/2011/10/dsdm-atern-project-structure-overview/ > (2011, octubre) | 181 |
| Fig. 36 | Roles y Responsabilidades de Aatern. Fuente: <i>DSDM Atern Project structure</i> < http://www.mcpa.biz/2011/10/dsdm-atern-project-structure-overview/ > (2011, octubre) | 183 |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | |
|----------|---|-----|
| Tabla 1 | Firmantes del Manifiesto por el Desarrollo Ágil de Software. Fuente: <i>Manifiesto por el Desarrollo Ágil de Software</i> < http://agilemanifesto.org/iso/es/ > (2001) | 25 |
| Tabla 2 | Diferencias entre Metodologías Ágiles y Tradicionales.Fuente: Canós, J et al (2003) | 27 |
| Tabla 3 | Ideas y Tareas Clave en las Actividades de XP. Fuente: Elaboración propia sobre <i>Ingeniería de Software</i> Pressman (2006:85) | 32 |
| Tabla 4 | Empresas que emplearon, para sus desarrollos, la Metodología Scrum Fuente: <i>Historia de Scrum</i> , < http://www.proyectosagiles.org/historia-de-scrum > | 37 |
| Tabla 5 | Secciones del Business Development Framework Fuente: <i>Business Development Framework. DSDM Tour</i> < www.dsdm.org > | 43 |
| Tabla 6 | Secciones del e-Business Framework Fuente: <i>e-Business Framework. DSDM Tour</i> < www.dsdm.org > | 44 |
| Tabla 7 | Principios de DSDM Fuente: <i>DSDM Tour</i> < www.dsdm.org > | 87 |
| Tabla 8 | Roles en DSDM Fuente: <i>DSDM Tour</i> < www.dsdm.org > | 87 |
| Tabla 9 | Significado de las Reglas MoSCoW Fuente: <i>DSDM Tour</i> < www.dsdm.org > | 90 |
| Tabla 10 | Cómo se consiguen Beneficios de Scrum. Fuente: <i>Beneficios de Scrum</i> < http://www.proyectosagiles.org/beneficios-de-scrum#expectativas > | 125 |
| Tabla 11 | Tipos de Proyectos. Fuente: <i>DSDM Tour</i> < www.dsdm.org > | 131 |
| Tabla 12 | DSDM Scrum Roadmap. Fuente: <i>DSDM Scrum Roadmap</i> . < http://orangefortune.com/DSDMScrumRoadmap.html > (2010) | 144 |
| Tabla 13 | Cuadro comparativo. Fuente: Elaboración propia | 150 |
| Tabla 14 | Respuestas al Cuestionario I para Desarrolladores de Proyectos Software. Fuente: Elaboración propia | 162 |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | |
|----------|---|-----|
| Tabla 15 | Respuestas al Cuestionario II para Evaluadores de Calidad de Empresas Desarrolladoras de Proyectos Software. Fuente: Elaboración propia | 164 |
| Tabla 16 | Respuestas al Cuestionario III para Ingenieros en Informática con conocimiento teórico de las Metodologías DSDM y Scrum. Fuente: Elaboración propia | 167 |
| Tabla 17 | Fases de un proyecto. Fuente: <i>DSDM Atern Project Structure</i> < http://www.mcpa.biz/2011/10/dsdm-atern-project-structure-overview/ > (2011, octubre) | 181 |
| Tabla 18 | Principios de Atern. Fuente: <i>DSDM Atern Principals</i> < http://www.mcpa.biz/2011/10/dsdm-atern-principals-overview/ > (2011, octubre) | 182 |
| Tabla 19 | Roles del Proyecto. Fuente: <i>DSDM Atern Roles and responsibilities</i> < http://www.mcpa.biz/2011/10/dsdm-atern-roles-and-responsibilities-an-overview/ > (2011) | 183 |
| Tabla 20 | Roles del desarrollo de soluciones. Fuente: <i>DSDM Atern Roles and Responsibilities</i> < http://www.mcpa.biz/2011/10/dsdm-atern-roles-and-responsibilities-an-overview/ > (2011) | 184 |
| Tabla 21 | Otros roles - Fuente: <i>DSDM Atern Roles and responsibilities</i> < http://www.mcpa.biz/2011/10/dsdm-atern-roles-and-responsibilities-an-overview/ > (2011) | 184 |

CAPITULO 0 – AGRADECIMIENTOS

Mi agradecimiento especial a Alicia Mon, por la claridad y calidad de sus clases, por orientarme y alentarme, con generosa paciencia y amabilidad, por su calidez humana.

A Bettina Donadello Anadón, por darme pautas formales y de contenido para la presentación de este trabajo y por impulsarme para terminarla, recordándome siempre que “la mejor tesis es la acabada”. Por ser mi amada hija y “mi mejor compañera”.

A Domingo Donadello, mi compañero, por su comprensión.

A Sebastián, mi querido hijo, que me dio “la felicidad”: Alejandro y Andrés, y otros dos nietos maravillosos, Julieta y Valentino.

Gracias a todos los docentes de la Maestría en Informática de la UNLaM, por los cursos que nos impartieron, y a los compañeros de estudio, con los que compartimos momentos deliciosos.

A Julio Bertúa y a Mabel Zanga, por conferirme confianza y respeto profesional.

Agradezco fundamentalmente a las Autoridades de la UNLaM, principalmente al Ing. Osvaldo Spóssito, al Ing. Gabriel Blanco, al Ing. Santiago Igarza y demás autoridades del Departamento de Ingeniería, por permitir mi desempeño como docente investigadora. También a todos mis compañeros de labor, por su solidaridad y colaboración.

Gracias a aquellos que me ayudan en el cuidado de mis nietos. No voy a nombrarlos porque son muchos, pero están muy presentes conmigo todos, en cada uno de mis días, por su dedicación profesional y por el afecto que sé que sienten por ellos.

A mis familiares, particularmente a Graciela y a Carla. A mis amigos.

Gracias.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

CAPITULO I - INTRODUCCION

Resumen

Esta tesis pretende colaborar en la toma de decisión sobre qué metodología ágil aplicar ante un proyecto de software para una determinada empresa; seleccionando entre las metodologías de desarrollo ágiles y, de estas, a DSDM y Scrum, para lo cual, en el CAP.I, se plantean dos hipótesis, dimensiones de análisis. Así como también, se esboza una solución: la construcción de un marco comparativo entre DSDM y Scrum.

En el CAP.II se enumeran los 12 principios de las Metodologías Ágiles. Se las compara con las metodologías tradicionales y se valoran, en términos de resultados, sus características esenciales, para introducir luego las Metodologías Scrum y DSDM. De las mismas, se relatan sus historias, mencionando sus ideólogos y/o creadores y sus antecedentes, los proyectos en los cuales fueron empleadas, los respectivos certificadores, y, en el caso de DSDM las versiones que fue generando DSDM Consortium, mencionando la creación de DSDM Atern.

El CAP.III describe de Scrum y de DSDM sus bases, roles, artefactos, técnicas, métricas, beneficios, restricciones, riesgos. Agregando al final las características sustanciales, los principios y productos de DSDM Atern.

En el CAP.IV se presenta la Propuesta de Análisis Comparativo, para lo cual se destacan primero los aspectos fundamentales de Scrum, en cuanto a roles y skills necesarios y potenciación del equipo, la planificación del proyecto, la planificación de la iteración, las métricas y el uso del cuadro de mandos integral, para terminar resaltando los beneficios de Scrum. Luego, haciendo lo propio con DSDM, se mencionan: la posición que asume en cuanto a los requisitos, los roles primordiales, el beneficio de los workshops que se implementan, las técnicas básicas y los criterios de aptitud/riesgo para determinar la utilización de DSDM.

Se expone un resumen de las características centrales de las metodologías DSDM y Scrum y se conforma un cuadro comparativo entre ambas que pone de manifiesto las diferencias entre las mismas.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Desde allí se realizan consideraciones especiales sobre DSDM y Scrum, teniendo en cuenta las dos hipótesis presentadas en el CAP.I, las que se estudian a partir de dimensiones específicas, contribuyendo a profundizar en el proceso de análisis. De allí en más surge la Propuesta de Análisis Comparativo que colabore a determinar el empleo de una u otra metodología o ambas, ante un determinado proyecto software.

El CAP.V presenta un diseño de análisis, considerando dos temas a investigar, referidos a la mayor fortaleza de DSDM respecto de Scrum y la mayor agilidad de Scrum en relación a DSDM. Agregándose las respuestas procesadas de los cuestionarios que se presentan en el ANEXO II, referidos a los temas a investigar,.

El CAP.VI contiene las conclusiones y detalla dos factibles trabajos a futuro: el análisis e investigación de posibles valores a tomar por las cuatro variables que se determinan en la Propuesta de Análisis Comparativo y la ampliación del campo de estudio del CAP.V

Área de trabajo

El área de trabajo, en la cual se desenvuelve esta tesis, es el de las Metodologías Ágiles de Desarrollo de Software, con énfasis particular en DSDM (Dynamic System Development Method) y SCRUM.

“Debido a que el software, como cualquier capital, es conocimiento materializado, y dado que el conocimiento en un inicio es disperso, tácito, latente y en gran medida incompleto, el desarrollo de software es un proceso de aprendizaje social. El proceso es un diálogo en el cual el conocimiento que el software debe convertir se conjunta y se materializa en este último. El proceso proporciona interacción entre los usuarios y las herramientas en evolución, y entre los diseñadores y sus herramientas (tecnología). Es un proceso iterativo en el que la herramienta en evolución sirve como un medio para la comunicación, en el cual cada nueva etapa del diálogo logra tener más conocimiento útil de las personas implicadas” (Baetjer, Jr. H., 1998:85).

Necesidad de una metodología para el Desarrollo de Software

El proceso de construcción del software requiere de una metodología, pues necesita identificar las tareas que se han de realizar y aplicarlas de forma ordenada y efectiva. Máxime que es un equipo de personas, desarrolladores y/o usuarios, quienes deberán disponerse en forma simultánea y coordinada, en cada fase del desarrollo, a prestar sus esfuerzos, dirigidos hacia un objetivo en común, en los menores tiempo y costes posibles.

Además, una metodología debe definir con precisión los artefactos, roles y actividades, junto con las prácticas, técnicas recomendadas y guías de adaptación de la metodología al proyecto. La complejidad del proceso de creación de software es netamente dependiente de la naturaleza del proyecto mismo, por lo tanto se escogerá la metodología de acuerdo al nivel de aporte al proyecto, ya sea pequeño, mediano o de gran nivel.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

¿Qué es una Metodología?

Etimológicamente, metodología viene del griego: *methodos* = camino a seguir y *logos* = estudio, pues *methodos* proviene de *meta* = más allá y *hodos* = camino.

Este concepto hace referencia al plan de investigación que permite cumplir ciertos objetivos en el marco de una disciplina, ya sea científica, ya sea artística, pudiendo tratarse también de algún plan aplicable a otros ámbitos toda vez que se trate de llevar a cabo una observación rigurosa. Es decir que puede entenderse que metodología es el conjunto de procedimientos planificados y organizados para alcanzar un objetivo determinado, en un ámbito específico.

Una Metodología para el Desarrollo de Software es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del Proceso Software, es decir del “conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o remplazado (su muerte)” [1]: Laboratorio Nacional de Calidad del Software de INTECO. España. (2009, Marzo).

Descripción del Problema

Ante un proyecto software, el problema consiste en determinar, entre DSDM y SCRUM, cuál metodología elegir para su desarrollo.

Se analizarán los fundamentos, roles, técnicas y artefactos de cada metodología, DSDM y Scrum, para determinar sus concordancias, solapamientos y diferencias esenciales.

Importancia del problema

La decisión que se tome en la elección de la metodología es relevante, porque en ella radica la diferencia de tiempo de entrega, coste, ROI, eventual desfasaje respecto del budget (presupuesto), riesgos y calidad, usabilidad, etc., siendo ROI (Return On

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

investment o Retorno sobre la Inversión) una razón financiera que compara el beneficio o la utilidad obtenida en relación a la inversión realizada.

La importancia del problema radica en la necesidad de determinar, ante un posible proyecto de desarrollo de software y en base a las diferencias que pudieran evidenciarse entre ambas metodologías, la conveniencia del uso de una u otra, si es que resulta perentoria una elección concluyente y terminante, en función del tiempo de entrega, costes y envergadura y características del proyecto y del cliente.

Objetivos, Motivación e Hipótesis

Objetivos

Se plantean dos objetivos:

- Objetivo General:
Describir y analizar las características esenciales de las metodologías ágiles DSDM y Scrum: fundamentos, actividades, roles, herramientas, técnicas.

- Objetivo Específico:
Comparar ambas metodologías, DSDM y Scrum, y profundizar, en cada una, cuáles son los beneficios y las restricciones de la aplicación de las mismas en proyectos software.

Motivación

Se eligió trabajar sobre metodologías ágiles por la particular manera de conformación de los equipos de desarrollo, por sus características:

- Competencia, en el mejor y más sano de los sentidos: por la capacidad, por el talento
- Enfoque común, aportando distintas habilidades en forma mancomunada y sin rivalidad
- Colaboración, entre ellos y con el cliente

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Habilidad para la toma de decisiones, autonomía: porque se reconoce su capacidad y esfuerzo
- Capacidad de resolución de problemas confusos, porque asimilan la experiencia, propia y ajena, y la comparten
- Confianza y respeto mutuo, esencial para este tipo de trabajo fundamentalmente creativo
- Organización propia, porque asumen absolutamente la responsabilidad sobre el tiempo y el coste

Además, porque son necesarios menos roles con más compromiso en las metodologías ágiles que en las tradicionales.

Se particularizó la elección de las metodologías DSDM, porque se adelanta a la creación de las metodologías ágiles, como precursora, y SCRUM porque se supone muy utilizada en Argentina, de hecho se ha radicado en Buenos Aires una organización que propicia y difunde esa metodología. Su página en internet es: <http://www.scrumargentina.com.ar>. Y porque justamente un equipo SCRUM desempeña menos roles que uno de DSDM, lo que implicaría, a priori, menos costes.

Hipótesis de la Tesis

Las hipótesis de esta investigación se basan en que las metodologías ágiles aportan beneficios a los proyectos de desarrollo software porque proveen una dinámica de trabajo más provechosa.

A continuación se enuncian las hipótesis:

1. DSDM es un enfoque que mantiene rigor metodológico y un control exhaustivo de un proyecto de software.
2. SCRUM obtiene resultados tempranos y a bajo coste, cumpliendo estándares de calidad.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Dimensiones de análisis

Las dimensiones que contribuyan a profundizar en el proceso de análisis para luego extraer conclusiones provisionales son:

- Registro exhaustivo.
- Control del proyecto.
- Calidad.
- Definición de roles.
- Coste.
- Gestión de riesgos.

Breve esbozo de la Solución

Construir un marco comparativo entre DSDM y Scrum, que brinde conocimiento completo sobre las características de ambas, para decidir, si conviene aplicar una u otra metodología o las dos, para el desarrollo de un proyecto de software determinado, para una organización específica.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

CAPITULO II - ESTADO DE LA CUESTION

En el presente capítulo se desarrolla el estado de la cuestión en relación a la historia de las metodologías ágiles, se las compara con los métodos tradicionales para desarrollo de software, para entonces mencionar, de las metodologías Scrum y DSDM, cómo surgen, algunas características que le son propias: su bibliografía, cómo certificar para poder utilizar, acabadamente y responsablemente, una u otra metodología, las empresas que las aplicaron para el desarrollo de sus proyectos o productos. A continuación, se hace referencia a las diferencias sustanciales entre Proyecto y Producto, marcando la forma en que DSDM encara un proyecto. Luego se agregan comentarios y opiniones, de distintos consultores internacionales, sobre la importancia de elegir una u otra metodología para un proyecto de desarrollo de software. Y finalmente, se hace referencia a investigaciones realizadas sobre metodologías ágiles para tesis de maestría y de doctorado.

Metodologías de Desarrollo de Software

Breve reseña histórica

Las metodologías más utilizadas a nivel mundial en orden cronológico son:

“Década de los 70s

- Programación Estructurada Jackson desde 1975

Década de los 80s

- Structured Systems Analysis and Design Methodology (SSADM) desde 1980
- Structured Analysis and Design Technique (SADT) desde 1980
- Ingeniería de la Información (IE/IEM) desde 1981

Década de los 90s

- Rapid Application Development (RAD) desde 1991.
- Programación Orientada a Objetos (OOP) a lo largo de la década de los 90's
- Virtual Finite State Machine (VFSM) desde 1990s
- Dynamic Systems Development Method desarrollado en UK desde 1995.
- Rational Unified Process (RUP) desde 1999

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Año 2000 en adelante

- Extreme Programming (XP) desde 1999
- Enterprise Unified Process (EUP) extensiones RUP desde 2002
- Constructionist Design Methodology (CDM) desde 2004 por Kristinn R. Thórisson
- Agile Unified Process (AUP) desde 2005 por Scott Ambler”.

[]: *Metodologías para el desarrollo de software*

<wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_para_el_desarrollo_de_software> (2012, julio)

Modelos de Desarrollo Agiles

En febrero de 2001, convocados por Kent Beck, quien había publicado un par de años antes *Extreme Programming Explained*, libro en el que exponía una nueva metodología denominada *Extreme Programming*, se reunieron en Snowbird, Utah, un grupo de 17 expertos en la industria del software, que incluía creadores o impulsores de metodologías de software.

Las cuestiones eran justamente:

- Esbozar valores y principios que permitieran a los equipos desarrollar software rápidamente y respondiendo a cambios eventuales que pudieran surgir en el desarrollo del proyecto.
- Pretender ofrecer una alternativa a los procesos de desarrollo de software tradicional, rígido y dirigido por la documentación que se genera en cada una de las actividades del proceso.
- Estimular las estructuras y actitudes de los equipos para la comunicación, entre los miembros del equipo, entre los técnicos y la gente de negocios, entre los ingenieros de software y sus gerentes.

Los integrantes de la reunión resumieron los principios sobre los que se basan los métodos alternativos en cuatro postulados, lo que ha quedado denominado como *Manifiesto Ágil*.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM



Fig.1: Manifiesto por el Desarrollo Ágil de Software.

Fuente: *Manifiesto por el Desarrollo Ágil de Software*. <<http://agilemanifesto.org/iso/es>>

“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”. [(2001): *Manifiesto por el Desarrollo Ágil de Software*

<<http://agilemanifesto.org/iso/es/>>

| | | |
|-------------------|----------------|------------------|
| Kent Beck | James Grenning | Robert C. Martin |
| Mike Beedle | Jim Highsmith | Steve Mellor |
| Arie van Bennekum | Andrew Hunt | Ken Schwaber |
| Alistair Cockburn | Ron Jeffries | Jeff Sutherland |
| Ward Cunningham | Jon Kern | Dave Thomas |
| Martin Fowler | Brian Marick | |

Tabla 1: Firmantes del Manifiesto por el Desarrollo Ágil de Software

Fuente: *Manifiesto por el Desarrollo Ágil de Software*. <<http://agilemanifesto.org/iso/es/>>

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Tras los cuatro valores descritos, los firmantes redactaron los siguientes 12, como los principios que de ellos se derivan:

- I. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- III. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- IV. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- V. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- VII. El software que funciona es la medida principal de progreso.
- VIII. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- IX. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- X. La simplicidad es esencial.
- XI. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- XII. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

El siguiente cuadro simplifica las diferencias esenciales entre Metodologías Ágiles y las Tradicionales:

| Metodologías Ágiles | Metodologías Tradicionales |
|--|--|
| Basadas en heurísticas provenientes de prácticas de producción de código | Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo |
| Especialmente preparados para cambios durante el proyecto | Cierta resistencia a los cambios |
| Impuestas internamente (por el equipo) | Impuestas externamente |
| Proceso menos controlado, con pocos principios | Proceso mucho más controlado, con numerosas políticas/normas |
| No existe contrato tradicional o al menos es bastante flexible | Existe un contrato prefijado |
| El cliente es parte del equipo de desarrollo | El cliente interactúa con el equipo de desarrollo mediante reuniones |
| Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio | Grupos grandes y posiblemente distribuidos |
| Pocos artefactos | Más artefactos |
| Pocos roles | Más roles |
| Menos énfasis en la arquitectura del software | La arquitectura del software es esencial y se expresa mediante modelos |

Tabla 2: Diferencias entre Metodologías Ágiles y Tradicionales.

Fuente: Canós, J et al, (2003)

La que sigue es una definición útil sobre metodologías ágiles:

“Agilidad se ha convertido actualmente en la palabra de moda en cuanto se describe un moderno proceso de software. Cualquiera es ágil. Un equipo ágil es un equipo rápido que responde de manera apropiada a los cambios. Estos son, en gran parte, la materia del desarrollo de software. Cambios en el software que se va a construir, cambios entre los miembros del equipo, cambios debidos a las nuevas tecnologías, cambios de todo tipo que pueden incidir en el producto que se construye o en el proyecto que crea el producto. En cualquier actividad de software se debe incluir un soporte para los cambios, esto es algo que adoptamos porque es el alma y el corazón del software. Un

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

equipo ágil reconoce que el software lo desarrollan individuos que trabajan en equipo y que las aptitudes de esta gente, y su capacidad para colaborar, son esenciales para el éxito del proyecto” (Jacobson, 2002)

Resumen de las Características de un Método Ágil

- Se basa en el Desarrollo iterativo e incremental
- Pone foco en la gente y no en el proceso.

Los métodos ágiles son orientados a la gente y no orientados al proceso. “La meta de los métodos ingenieriles es definir un proceso que funcionará bien con cualquiera que lo use. Los métodos ágiles afirman que ningún proceso podrá nunca maquillar las habilidades del equipo de desarrollo, de modo que el papel del proceso es apoyar al equipo de desarrollo en su trabajo. Explícitamente puntualizan el trabajar a favor de la naturaleza humana en lugar de en su contra y enfatizan que el desarrollo de software debe ser una actividad agradable” (Fowler, 2002)

- Es Cooperativo
- Es Sencillo
- Es Adaptable

Beneficios de la adaptabilidad para el cliente en este sentido:

- Un desarrollo de software mucho más sensible.
- Un sistema usable, aunque mínimo, puede entrar en producción enseguida. La posibilidad de cambiar sus capacidades de acuerdo a los cambios en el negocio,
- Aprender cómo se usa el sistema en realidad.

Los métodos ágiles son adaptables en lugar de predictivos.

“Los métodos ingenieriles tienden a intentar planear una parte grande del proceso del software en gran detalle para un plazo largo de tiempo, esto funciona bien hasta que las cosas cambian. Así que su naturaleza es resistirse al cambio. Para los métodos ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio, incluso al punto de cambiarse ellos mismos”. (Fowler, 2002)

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Se pueden mencionar, como contrapartida, algunos perjuicios de un método predictivo:

El cliente no puede aprovechar resultados del proyecto hasta se hayan realizado todas las actividades planeadas, con la eventualidad de haberse producido entretanto un cambio del contexto del proyecto. Finalmente el proyecto, tras varios meses, está listo para ser entregado, pero con el agravante de que pueden surgir, en ese momento, riesgos y tareas no previstas, que retrasen la entrega, obliguen al equipo a un sobreesfuerzo, al rehacer partes del producto provocando un incremento de su coste (para el proveedor y/o para el cliente), deteriorándose en consecuencia la calidad del producto final.

Beneficios de los métodos empíricos:

- Admiten que siempre hay cambios en el contexto del proyecto, por su indeterminación y complejidad.
- Se funda en inspección y adaptación regular de acuerdo a resultados obtenidos y en función del contexto y sus posibles variaciones.
- El cliente va entendiendo el producto a medida que se va desarrollando, lo que le implica un esfuerzo menor al tomar decisiones.
- En cada nueva iteración los requisitos a trabajar dependen del conocimiento que el cliente va teniendo del proyecto, de la velocidad de desarrollo, de las demandas del mercado, de los movimientos de los competidores, etc. Surgen requisitos y es necesario adaptarse a ellos.

Priorización de requisitos por valor y coste:

En cada iteración, en el último día, luego de la demostración, el cliente reprioriza los requisitos, lo que configura un fundamental soporte al control empírico, según su valor en ese momento y su coste estimado de desarrollo. De esta manera actualiza la lista de requisitos priorizada (Product Backlog). Y puede llegar a un punto en que no valga la pena desarrollar los requisitos restantes, de acuerdo al poco retorno de inversión (ROI) que tengan.

- Se realizan Revisiones regulares del proceso, se hacen con cada iteración.

Se trata de responder, en una corta reunión, cuatro preguntas escogidas por N. Kerth (2001):

- “¿Qué hicimos bien?”
- “¿Qué hemos aprendido?”
- ¿Qué podemos hacer mejor?

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- “¿Qué es lo que nos confunde?”
- Las decisiones técnicas la toman los desarrolladores.
- Están más orientados al código.

“La agilidad es dinámica, con contenido específico, ajustable al cambio de manera dinámica y orientada al crecimiento”. (Steven Goldman et al, 1996)

“La agilidad se puede aplicar en cualquier proceso de software. Sin embargo, para lograrlo es esencial que el proceso sea diseñado en una forma que permita al equipo del proyecto adaptar y coordinar las tareas, conducir la planeación en una forma que entienda la fluidez de un enfoque de desarrollo ágil, eliminar todo pero no los productos de trabajo esenciales y mantenerlos controlados, y enfatizar una estrategia de entrega incremental que proporcione software en funcionamiento al cliente tan rápido como sea factible para el tipo de producto y el ambiente operativo”.(Pressman, 2006:80)

Metodología Scrum

Se describe brevemente a continuación, la metodología en que Scrum basa sus prácticas de código:

Extreme Programming (XP) o Programación Extrema (PE)

Tiene como objetivo reducir el riesgo en el ciclo de vida del software mediante grupos de desarrollos pequeños.

Fue creado por Kent Beck (1980), quien publicó su trabajo fundamental sobre la materia 19 años más tarde. Los libros subsiguientes de Jeffries et al. (2001), sobre los detalles técnicos de la XP, y el trabajo adicional de Beck y Martin Fowler sobre la planeación de la XP expusieron los detalles del método.

La XP utiliza un enfoque OO.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

XP define cuatro variables para proyectos de software: coste, tiempo, calidad y ámbito, de las cuales sólo tres pueden ser establecidas por jefes de proyecto y clientes, y el valor de la cuarta será establecido por los programadores en función de las otras tres.

Valores originales de la programación extrema

- **Simplicidad:** hacer la cosa más simple que pueda funcionar, en relación al proceso y la codificación. Base de la programación extrema.
- **Comunicación:** se programa por parejas, requiere que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Mientras uno redacta Pruebas de Unidad, el otro piensa acerca de la clase que satisfará a dicha prueba, por ejemplo. Los estudios demuestran que, tras aprender PeopleSkills, dos programadores son más que doblemente productivos que uno sólo para una tarea determinada. Por eso los programadores se comunican constantemente, y la comunicación con el cliente es fluida porque él mismo forma parte del equipo de desarrollo. Se aclara que PeopleSkills es la capacidad de entender y de comunicarse con otros, según lo descrito en el libro *How To Win Friends And Influence People* por Dale Carnegie (1998, octubre). Uno de los trabajos más importantes e influyentes sobre habilidades de comunicación efectivas

Los desarrolladores realizan un ejercicio, PairDraw, que es una simulación de la programación en parejas, diseñado para dar a los programadores y administradores una idea de cómo es la programación en parejas, antes de que realmente lo intenten. Esta simulación fue inventada por Joshua Kerievsky (1996), fundador de Industrial Logic cuya página es <http://www.industriallogic.com>.

Se puede leer más sobre PairDraw en www.industriallogic.com/?s=pairdraw.

- **Retroalimentación:** concreta y frecuente del cliente, del equipo y de los usuarios finales da una mayor oportunidad de dirigir el esfuerzo.
- **Refactorización:** los desarrolladores reconstruyen su código cuando sea necesario.
- **La planificación es incremental**
- **Propiedad colectiva:** las parejas de desarrolladores trabajan en todas las áreas del sistema.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Ritmo sostenible: no se consideran aceptables grandes cantidades de horas extras, ya que a menudo, reduce la calidad del código y la productividad a medio plazo.
- Cliente presente: un representante de los usuarios finales del sistema a tiempo completo tiene que estar a disposición del equipo de XP.

XP abarca un conjunto de reglas y actividades que ocurren en el contexto de cuatro actividades del marco del proceso, en cada una de las cuales se asocian ideas y tareas clave:

| Actividades | Ideas y Tareas Clave |
|---------------------|---|
| Planeación | <ul style="list-style-type: none">- Historias del usuario: describen las características y funcionalidades requeridas y el cliente las escribe en una carta índice y les da un valor o prioridad- Valores: los miembros del equipo evalúan las historias para asignarles un costo- Criterios de las pruebas de iteración- Plan de iteración |
| Diseño | <ul style="list-style-type: none">- Diseño simple, es preferible a una presentación más compleja- Cartas CRC (Clase, Responsabilidad y Colaboración) identifican y organizan las clases orientadas al objeto relevantes para el incremento del software actual. Son una metodología para el diseño de software orientado por objetos creada por Kent Beck y Ward Cunningham. <p>Soluciones pico, prototipos que se implementan y evalúan, con el fin de reducir riesgos</p> |
| Codificación | <ul style="list-style-type: none">- Programación en pareja, concepto clave.- Refactorización, proceso de cambiar un sistema de manera que no altere el comportamiento externo del código y que mejore la estructura interna. Es una forma de “limpiar el código”.- Integración continua, entre el trabajo de unos desarrolladores y otros, evita problemas de compatibilidad e interfaz. |
| Pruebas | <ul style="list-style-type: none">- Pruebas de unidad, deben implementarse en un marco que permita automatizarlas. Estas pruebas consisten en verificar que el resultado de un método, una clase específica, corresponda con un resultado esperado. Se aconseja escribir primero las pruebas antes de escribir propiamente el código que va a verificarse.- Integración continua- Pruebas de aceptación, o pruebas del cliente, las especifica el cliente. Se derivan de las historias del usuario. |

Tabla 3: Ideas y Tareas Clave en las Actividades de XP.

Fuente: Elaboración propia sobre Pressman (2006:85)

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Scrum

“El concepto de Scrum tiene su origen en un estudio de 1986 sobre los nuevos procesos de desarrollo utilizados en productos exitosos en Japón y los Estados Unidos (cámaras de fotos de Canon, fotocopiadoras de Xerox, automóviles de Honda, ordenadores de HP y otros). Los equipos que desarrollaron estos productos partían de requisitos muy generales, así como novedosos, y debían salir al mercado en mucho menos tiempo del que se tardó en lanzar productos anteriores. Estos equipos seguían patrones de ejecución de proyecto muy similares. En este estudio se comparaba la forma de trabajo de estos equipos altamente productivos y multidisciplinarios con la colaboración entre los jugadores de Rugby y su formación de Scrum (Melé en español)”.([: *Historia de Scrum*. <<http://www.proyectosagiles.org/historia-de-scrum>>).

Pero recién en 1993 se realizó el primer Scrum. Ken Schwaber, creador de Scrum junto con Sutherland, dice: “que procesos definidos y repetibles sólo funcionan para atacar problemas definidos y repetibles con gente definida y repetible en ambientes definidos y repetibles”. Pero, por supuesto, no todos los problemas son definidos y repetibles.

Ken Schwaber, nacido en 1945, es desarrollador de software, gerente de producto y consultor de la industria. Ken trabajó con Jeff Sutherland para formular las primeras versiones de la Metodología Scrum proceso de desarrollo y presentar Scrum como un proceso formal en OOPSLA '95 (OOPSLA: Programación Orientada a Objetos, Sistemas, Lenguajes y Aplicaciones, evento anual ACM (Association for Computing Machinery - Conferencia de Investigación).

Schwaber y Sutherland (2001) son de los firmantes iniciales del Manifiesto Ágil y coautores de la definitiva *Scrum Guide*, puesta a disposición, en forma gratuita, por Scrum.org.

Hoy Schwaber continúa en Scrum.org, que proporciona recursos de Scrum, capacitación, evaluaciones y certificaciones para los principales roles:

- Scrum Masters o Facilitador, directo colaborador del Product Owner o Cliente y del Equipo. No tiene la autoridad de gestión y no puede comprometerse a trabajar en

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

beneficio del equipo. Mantiene los procesos y trabaja de forma similar al director de proyecto.

- Scrum Developers o Scrum Desarrolladores, son parte del Scrum Team.
- Scrum Product Owner o Scrum Dueño del Producto, representa a los stakeholders (interesados externos o internos). Es un interesado clave de un proyecto. Parte de su responsabilidad es tener una visión de lo que desea construir y transmitirla al equipo de Scrum. El propietario del producto es normalmente un usuario líder del sistema o alguien de marketing, gestión de productos, o cualquier persona con un conocimiento sólido de los usuarios, el mercado, la competencia y las tendencias futuras de en el dominio o tipo de sistema a desarrollar.
- y las organizaciones que utilizan Scrum.

Schwaber es uno de los fundadores de la Alianza Ágil, el responsable de la fundación de la Alianza Scrum y la creación de los programas de Maestría certificados Scrum y sus derivados. Ken Schwaber dejó la Alianza Scrum en el otoño de 2009, después de un grave accidente de bicicleta. Luego fundó Scrum.org (2009), donde lideró el desarrollo de nuevos cursos, evaluaciones, y las asociaciones para mejorar la calidad y la eficacia de Scrum. Recientemente ha publicado y actualizado Scrum con Jeff Sutherland, y escribió *Software de Treinta Días* (2012, 5 de abril), también con Sutherland.

Scrum divide un proyecto en iteraciones, que ellos llaman carreras cortas (Sprint), de 30 días.

Antes de que comience una carrera se define la funcionalidad requerida para esa carrera y se deja al equipo para que la entregue. El punto es estabilizar los requisitos durante la carrera.

Sin embargo la gerencia no se desentiende durante la carrera corta. Todos los días el equipo sostiene una junta corta, de quince minutos, llamada Scrum, donde el equipo discute lo que hará al día siguiente. En particular señalan a la gerencia los impedimentos que surgen e impiden progresar, y que la gerencia debe resolver, si no pueden llegar a resolverlos solos. También informan lo que se ha hecho para que la gerencia tenga una actualización diaria de por dónde va el proyecto.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

La literatura de Scrum se enfoca principalmente en la planeación iterativa y el seguimiento del proceso. Es muy cercana a las otras metodologías ágiles en muchos aspectos y debe funcionar bien con las prácticas de código de la metodología ágil XP (Programación Extrema).

Después de mucho tiempo sin un libro que describiera su metodología, finalmente Ken Schwaber y Mike Beedle escribieron el primer libro de scrum: *Agil Software Development with Scrum* (2001, 21 de octubre)

Ken Schwaber también se manifiesta en controlChaos.com, que probablemente es la mejor apreciación global sobre Scrum, donde publicita el libro que escribiera junto a Jeff Sutherland: *Software in 30 days* (2012, abril).

Jeff Sutherland siempre ha tenido un sitio web activo sobre temas de tecnologías de objetos e incluye una sección sobre Scrum. Hay también una buena apreciación global de las prácticas de Scrum en el libro *PLoPD 4* (1999, diciembre), escrito por Brian Foote, Neil Harrison y Hans Rohnert.

Scrum tiene un Grupo de Discusión en Yahoo:

<http://groups.yahoo.com/neo/groups/scrumdevelopment/info>

Group Email Addresses: scrumdevelopment@yahoogroups.com

Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software o en una aproximación de gestión de proyectos, Scrum de Scrums, como se verá más abajo.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

planificada. Por lo tanto, Scrum acepta que el problema no puede ser completamente entendido o definido y se centra en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

Scrum se inició en 1993 para equipo de software de Easel Corporation, donde Jeff Sutherland era VP of Object Technology.

IDX, una de las más grandes compañías de software médico de US, fue el campo de prueba para implementaciones de Scrum de equipo múltiple. Casi 600 desarrolladores trabajaron en docenas de productos, tales como: Flowcast, Groupcast, Carecast e ImageCast. Esta empresa fue adquirida por General Electric, que la integró como GE Healthcare.

En el año 2000, Scrum fue introducido para PatientKeeper, compañía de plataforma para móviles inalámbricos para servicios médicos.

El primer Scrum para internet se realizó para Individual Inc., y el primer Scrum, para desarrollo a gran escala, fue para IDX, en 1996. El enfoque en IDX era convertir a toda la organización para el desarrollo en un conjunto interrelacionado de Scrums. Cada parte de la organización fue equipo con base, incluyendo el equipo de gestores, el que incluía dos vice presidentes, un arquitecto senior y varios directores. El Front-line de Scrums se reunía diariamente. Un Scrum de Scrums, que incluía los líderes de equipo de cada Scrum en una product line, se reunía semanalmente. El Scrum management se reunía mensualmente.

En Argentina se puede recibir instrucción y certificación para desarrollo con Metodología Scrum, por ejemplo, en:

- FRBA- UTN, en modalidad online a través del Campus Virtual FRBA.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Hexacta, empresa argentina especializada en gestión de proyectos ágiles con el aval de CESSI (Cámara de Empresas de Software y Servicios Informáticos de la República Argentina).
- INTI Instituto Nacional de Tecnología Industrial

La Instrucción y Certificación Internacional como Scrum Master es otorgada por Yellowhouse a través de Scrum.org, siendo Yellowhouse una boutique consultora de la mejor gestión práctica, con sede en Australia.

“Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos”.([]: *Qué es Scrum* <<http://www.proyectosagiles.org/que-es-scrum>>).

| Sectores | Empresas |
|-------------------------------|--|
| Media y Telcos | BBC, BellSouth, British Telecom, DoubleYou, Motorola, Nokia, Palm, Qualcomm, Schibsted, Sony/Ericsson, Telefonica I+D, TeleAtlas, Verizon |
| Software, Hardware | Adobe, Autentia, Biko2, Central Desktop, Citrix, Gailén, IBM, Intel, Microfocus, Microsoft, Novell, OpenView Labs, Plain Concepts, Primavera, Proyectalis, Softhouse, Valtech, VersionOne. |
| Internet | Amazon, Google, mySpace, Yahoo |
| ERP | SAP |
| Banca e Inversión | Bank of America, Barclays Global Investors, Key Bank, Merrill Lynch |
| Sanidad y Salud | Patientkeeper, Philips Medical |
| Defensa y Aeroespacial | Boeing, General Dynamics, Lockheed Martin |
| Juegos | Blizzard, High Moon Studios, Crytek, Ubisoft, Electronic Arts |
| Otros | 3M, Bose, GE, UOC, Ferrari |

Tabla 4: Empresas que emplearon, para sus desarrollos, la Metodología Scrum
Fuente: *Historia de Scrum*. <<http://www.proyectosagiles.org/historia-de-scrum>>

Metodología DSDM

DSDM, Dynamic System Development Method, es una combinación eficiente del conocimiento de las personas y técnicas para realizar proyectos rápidamente.

El equipo de desarrollo y usuarios trabajan juntos.

Su principal objetivo es evitar producir sistemas que:

- No cumplan los requerimientos
- No funcionen correctamente
- Caigan en desuso.

Origen de DSDM

A principios de los 90 surgió el concepto de RAD (Rapid Application Development). El mismo fue pensado para ser diferente de los métodos clásicos, secuenciales o en "cascada", para el desarrollo de aplicaciones. Es una metodología de desarrollo de software, que se basa en el desarrollo iterativo y la construcción de prototipos. Y cuyos principios básicos son, entre otros:

- Reducción de riesgos inherentes al proyecto partiéndolo en segmentos más pequeños y proporcionando más facilidad de cambio durante el proceso de desarrollo.
- Uso de iteración por prototipos (en cualquier etapa de desarrollo), participación de los usuarios como imprescindible y uso de herramientas de desarrollo computarizadas (constructores de Interfaz Gráfica de Usuario (GUI), herramientas Computer Aided Software Engineering (CASE), Sistemas de Gestión de Bases de Datos (DBMS)), lenguajes de programación de cuarta generación, generadores de código, y técnicas orientadas a objetos.
- Hacer especial hincapié en el cumplimiento de la necesidad comercial, mientras que la ingeniería tecnológica o la excelencia es de menor importancia.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Que el control de proyecto implica el desarrollo de prioridades y la definición de los plazos de entrega. Si la finalización del proyecto comienza a aplazarse, se hace hincapié en la reducción de requisitos para el ajuste, y no se aumenta la fecha límite.
- Inclusión de Joint Application Development (JAD), que consiste en un taller donde los trabajadores del conocimiento y los especialistas en Tecnologías de Información se reúnen, incluso durante varios días, para definir y revisar los requerimientos de negocio para el sistema o proyecto.

JAD es un proceso usado en el área del ciclo de vida de prototipado del Método de Desarrollo de Sistemas Dinámicos (DSDM) para reunir requerimientos en el desarrollo de nuevos sistemas de información para una compañía. Chuck Morris y Tony Crawford, ambos de IBM, desarrollaron JAD a finales de 1970 y comenzaron a enseñar el método a través de talleres en 1980. El proceso JAD es una técnica que involucra al cliente o usuario final en el diseño y desarrollo de una aplicación, también incluye enfoques para la mejora en la participación de los usuarios, agilizar el desarrollo y mejorar la calidad de las especificaciones.

- Producción de la documentación necesaria para facilitar el futuro desarrollo y mantenimiento.

Creación de DSDM

En marzo de ese año, 1990, por decisión de 36 miembros, se crean tres tipos de grupos para supervisar varias actividades:

- El Grupo de Trabajo Técnico, creador del DSDM Framework, que estableció los grupos de tareas para definir completamente los contenidos detallados de Framework.
- Los Grupos de Trabajo de Políticas y Procedimientos, para idear el proceso de toma de decisión y producir un libro de reglas del consorcio. Este grupo fue reemplazado por el Grupo de Trabajo de Acreditación y Entrenamiento, dedicado a supervisar y controlar todas actividades de entrenamiento en DSDM
- El Grupo de Trabajo Promocional, para manejar un plan de marketing.

El workgroup técnico, motor detrás del desarrollo del framework, creó, para alcanzar resultados rápidos, los Grupos de Tarea para la Versión 1 de DSDM.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- El Grupo de Tareas de Herramientas y Técnicas de Desarrollo era responsable de definir el default estructurado, las técnicas O O y pautas JAD- Joint Application Development (Desarrollo Conjunto de Aplicaciones).
- El Grupo de Tareas de Herramientas y Técnicas de Gestión, responsable de definir pautas en control de cambio, gestión de configuración, gestión de riesgo y estimar grupo.
- El Grupo de Tareas de Personal, era responsable de Gestión de Proyecto y Estructuras de Equipo.
- El Grupo de Tareas de Calidad, era responsable de dirección de ediciones de Quality Management, Quality Assurance y Testing.
- El Grupo de Tareas de Consecución de Software era responsable de la dirección de ediciones contractuales.

En enero de 1995 se completó la Versión 1, con el acuerdo de todos los miembros del Consortium, publicándose en febrero de 1995. Junto a la publicación del framework, el Consortium dio lugar a un esquema de entrenamiento, con media docena de las organizaciones de entrenamiento acreditadas y el procedimiento de examinación, para que los practicantes de DSDM obtengan la certificación.

Junto a la publicación del framework, el Consortium dio lugar a un esquema de entrenamiento, con media docena de las organizaciones de entrenamiento acreditadas y el procedimiento de examinación, para que los practicantes de DSDM obtengan la certificación. La British Computer Society determinó y aprobó los procedimientos de acreditación y de examinación y continúa editando los certificados para los practicantes que vienen para la examinación a través del Consortium UK.

Para verificar en la práctica el framework, un Early Adopters Programme fue lanzado en paralelo con la Versión 2. Como los proyectos RAD se completan pronto, el feedback del Early Adopter Projects junto con el trabajo de Grupos de Tarea futuros condujo a la Versión 2 siendo publicada en diciembre de 1995, tres semanas antes de lo programado.

En enero de 1997, un workshop ayudó a decidir qué cambios podrían ser necesarios para el método. Se había incrementado el uso de DSDM en proyectos de cambio de procesos de negocio, mientras que el foco anterior había estado puesto en el desarrollo

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

de aplicaciones puramente. Este era el filtro para ser un mayor cambio que pueda ser reflejado en el framework y los Grupos de Tareas fueron fijados para considerar sus implicancias. La Versión 3 fue publicada en octubre de 1997.

El Consortium DSDM ayudó a producir y desarrollar un enfoque de hecho para construir ambos sistemas rápidamente y bien. En UK, DSDM es el más comúnmente usado enfoque para RAD desde que este tiene éxitos demostrados repetidos en organizaciones de todos los portes en ambos sectores el público y el privado. Esto está ganando prominencia internacional creciente.

La fuerza de DSDM es que estaba siendo construida sobre las bases de técnicas probadas y testeadas usadas por organizaciones miembro.

DSDM Consortium

La licencia para el uso del DSDM Framework es supervisada por el consorcio DSDM, que desarrolla y promueve continuamente el DSDM Framework. Solamente los Full Members del DSDM Consortium están autorizados para usar el DSDM Framework en proyectos, ya sea interna y/o externamente.

Todos los miembros del Consorcio pueden colaborar en la evolución del Framework El Consorcio Internacional, está centrado en el Reino Unido, pero hay otros consorcios en Benelux (antigua región comprendida por los países que formaron el Reino de los Países Bajos, Bélgica, los Países Bajos y Luxemburgo), Francia, Norteamérica, Suecia y Dinamarca.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Business Centred Development o Desarrollo Centrado en el Negocio

El DSDM Framework se ha comprometido con esta clase de desarrollo desde 1994. Aseguran los miembros del Consortium que el desarrollo centrado en el negocio satisface mejor las necesidades de los miembros en la entrega de soluciones de negocio de alta calidad a tiempo y dentro del budget.

Delivery Framework o Entrega del Marco

El marco principal y el enfoque de e-Business se entregan online a todos los full members, quienes además reciben un CD para uso en intranets y notebooks. El DSDM Framework y los productos asociados son propiedad del DSDM Consortium. Una persona, al unirse al Consorcio como full member, está licenciado para usar el DSDM Framework.

Business Development Framework o Marco de Desarrollo Empresarial

Es entregado en forma de manual online, navegable de una sección a otra, proporciona rutas claras de los próximos pasos del proyecto. Dada la experiencia de los miembros que contribuyeron al framework, los mismos agregan en este manual sus mejores consejos.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| Business Development Framework Sections o Secciones del Marco de Desarrollo de Negocio | |
|---|--|
| Introducción | Visión general del Marco |
| Cuándo usar DSDM | Cuestiones a considerar antes de elegir el uso de DSDM |
| Ciclo de Vida | Va a través de cada fase definiendo cada una en detalle y proporciona vínculos claros para aplicar técnicas y productos de principio en cada fase. |
| Gente | Con amplia cobertura de los roles dentro del proyecto, explica cómo asegurarse de que el factor humano se gestiona eficazmente. |
| Productos | Productos entregados en cada fase del proyecto así como la orientación sobre la adecuada elección de las técnicas de modelado y desarrollo. |
| Gestión Herramientas y Técnicas | Detalles de diversas técnicas como Timeboxing, Moscow, planificación, gestión de riesgos, medición, estimación, gestión de la calidad |
| Desarrollo – Técnicas | Secciones detalladas sobre Workshops Facilitated, prototipeado, técnicas de modelado, pruebas, gestión de la configuración y entornos de apoyo. |
| Adaptación | Guía sobre el uso de DSDM en pequeños proyectos, grandes proyectos y proyectos híbridos. |
| Otros | Los temas relacionados con la introducción de DSDM en cualquier organización |

Tabla 5: Secciones del Business Development Framework

Fuente: *DSDM Tour*. <www.dsdm.org>

Donde Timeboxing es el uso de Timebox:

consiste en fijar el tiempo máximo para conseguir unos objetivos, tomar una decisión o realizar unas tareas, y hacer lo mejor que se pueda en ese intervalo. Y MoSCoW son reglas para la priorización de requisitos, se verán el CAP. III lo mismo que los Workshops Facilitated.

e-Business Framework

Es un marco específico para proyectos empresariales innovadores, que utilizan tecnologías basadas en la web, suplementos e-DSDM y complementos DSDM.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| e-Business Framework Sections o Secciones del Marco del e-Business | |
|---|--|
| Cuándo | Todas las preguntas habituales de la Suitability/Risk List o Lista de Adecuación/Riesgo siguen siendo válidas sin embargo, la participación del usuario necesita una atención especial como lo hace la alineación de la seguridad y la garantía con la estrategia de e-business de la organización. |
| Ciclo de vida | Un proyecto de e-DSDM sigue el mismo proceso que un proyecto de DSDM con la excepción de una fase Visión. La fase Visión precede a un conjunto de proyectos de e-DSDM. Su objetivo es establecer la estrategia de e-business para un programa de e-business. |
| Gente | Los roles adicionales incluyen Gestor de Co-Branding/Partnership (donde Co-Branding se define como Asociación de dos marcas con el fin de potenciar el valor y la rentabilidad de las mismas y Partnerchip significa Alianza), Content Manager, especialista en usabilidad, Diseñador Web y Webmaster. |
| Productos | Los nuevos productos incluyen una Política de Seguridad, Estilo y Guía de Normas, Prototipo de la Arquitectura Técnica y Evaluación de Riesgo Técnico. |
| Desarrollo | e-DSDM tiene una sección de calidad e implementación, que se ocupa de cuestiones tales como las pruebas y aseguramiento de la calidad. La prueba de la experiencia final del usuario y la validación de los contenidos son tratados en detalle. |

Tabla 6: Secciones del e-Business Framework

Fuente: *DSDM Tour*. <www.dsdm.org>

DSDM Atern

En el año 2007, en el Café Royal, DSDM lanza una nueva versión que llamó DSDM Atern. El nombre Atern es un acortamiento del charrán ártico, un pájaro que pueden viajar grandes distancias (“de polo a polo”), y personifica muchas facetas del método que son formas naturales de ejemplo de trabajo: establecimiento de prioridades y colaboración.

La versión anterior de DSDM que sigue siendo ampliamente utilizada y válida es DSDM 4.2, versión ligeramente ampliada de DSDM versión 4.

Atern aprovecha el conocimiento, la experiencia y la creatividad de los usuarios finales. Utiliza un ciclo de vida iterativo para desarrollar la solución más adecuada para satisfacer los objetivos del proyecto. Define los roles claramente. Al fraccionar el

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

proyecto en períodos de tiempo cortos enfocados, timeboxes con plazos inamovibles y resultados acordados, cada uno con resultados claramente especificados, el control se ejerce en todo momento por el director de proyecto y por los propios miembros del equipo.

DSDM Agile Project Management - AGILE

Agile Project Management, AGILE, se publica en 2011, es un enfoque para la gestión de proyectos, que ayuda a trabajar de manera efectiva para lograr los objetivos de negocio. Ofrece la posibilidad de entregar los proyectos ágiles en organizaciones que requieren normas, rigor y visibilidad en la Gestión de Proyectos, permitiendo un ritmo y el cambio rápido de Agile. Agile Project Management se basa en los aspectos fundamentales ya comprobados en Atern DSDM, abordando el concepto de un proyecto ágil y proporcionando una guía detallada sobre cómo gestionar y entregar los proyectos ágiles. Agile Project Management es el resultado de la colaboración entre APMG-International y el consorcio DSDM.

Beneficios de AGILE :

Pharro Richard (2011), CEO, APMG-International, expresa en el prefacio del Agile Project Management:

- “Agile Project Management provee un enfoque que ofrece agilidad pero conserva los conceptos de proyecto, entrega de proyecto y gestión de proyecto”
- “Se obtienen los beneficios de un enfoque ágil, sin introducir riesgos innecesarios”.
- “La organización puede adoptar un enfoque comprobado, sin necesidad de desarrollar e integrar procesos específicos de la gestión ágil de una compañía”.
- “Agile Project Management proporciona una guía que permite una gestión y una entrega exitosas aunque la organización sea nueva en la gestión de proyectos pero quiera entregar proyectos ágiles de forma controlada”.
- “Combina conocimientos de la gente con las técnicas”.
- “AGILE fija tiempo, costo y calidad en las fases iniciales de un proyecto, lo que la diferencia de cualquier enfoque tradicional”.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- “Trabaja junto a PRINCE2, el más formal enfoque de gestión de proyectos, y complementa los procesos de calidad, como ISO9001 y CMMI”. ([1] (2011): *Agile Project Management* <<http://www.apmg-international.com/>>)

Es oportuno comentar que PRINCE2 proviene del acrónimo en inglés PRojects IN Controlled Environments, es decir, convertir proyectos, que manejan una carga importante de variabilidad e incertidumbre, en entornos controlados.

PRINCE2 es una Metodología de Gestión de Proyectos que cubre, a través de lo conocido como Temáticas, Calidad, Cambio, Estructura de Roles del Proyecto (Organización), Planes (Cuánto, Cómo, Cuándo), Riesgo y Progreso del Proyecto, justificado por un Business Case (o estudio de viabilidad) a revisarse durante el ciclo de vida del proyecto y justificar en todo momento el proyecto como consecución de los beneficios esperados.

La metodología PRINCE2 se puede aplicar en proyectos de toda índole, como Desarrollo de Software o Construcción.

DSDM Agile Project Framework

DSDM crea el Agile Project Framework, una abstracción de DSDM Atern diseñado para complementar otros enfoques ágiles, en particular Scrum.

En el año 2012, DSDM Consortium publicó, en www.dsdm.org, *The DSDM Agile Project Framework for Scrum*, producido por Andrew Craddock, Director Técnico de DSDM Consortium, en coautoría con otros miembros del Technical Team en el DSDM Consortium, para DSDM Consortium, bajo la revisión importantes miembros de la comunidad Scrum.

Se señala que *The DSDM Agile Project Framework for Scrum* consta de una Introducción (a DSDM Agile Project Framework, a DSDM y a Scrum), una descripción

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

de las características y una enumeración de las técnicas del DSDM Agile Project Framework, un breve Resumen y una referencia a los autores.

En principio DSDM Consortium publicó sólo un capítulo, *Timeboxing*, extracto del DSDM Agile Project Framework, exclusivamente para fines promocionales y en su versión completa, se publicó a finales de 2013 en su sitio web www.dsdm.org.

Agile Business Analysts – AgilBA

Actualmente, DSDM está desarrollando la orientación basada en roles. Se está creando un manual para los analistas de negocio ágiles, basado en DSDM Atern, pero extendiendo esto con otras prácticas e información para apoyar al analista de negocios en cualquier proyecto Agile. Se previó el lanzamiento del manual completo de AgilBA para principios del 2014.

En principio, DSDM Consortium publicó en www.dsdm.org un extracto de AgilBA, *El Ciclo de Vida de los Requisitos en un Proyecto Ágil*, en cuya introducción aclara: “El enfoque Agile (DSDM) para los requisitos es aclarar el objetivo del proyecto y el Business Case y luego definir los requisitos, sólo en un alto nivel, al principio del proyecto. La definición de los requisitos detallados se dejó deliberadamente hasta justo antes de que sea necesario: justo antes de que el aspecto particular de la solución es desarrollada y construida. Esto evita el desperdicio y reproceso. También asegura que el producto pueda evolucionar para reflejar las necesidades de la empresa en el momento de la construcción de la solución. Permite el aprendizaje durante el proyecto y cambiar para ser aceptado, en lugar de ser tratado como un problema.

Sin embargo, el riesgo inherente a este enfoque gradual para la definición de los requisitos es que un conjunto inconsistente o internamente en conflicto de los requisitos podría evolucionar. Aquí es donde las habilidades del AgilBA son esenciales. El AgilBA debe facilitar la evolución de los requisitos de objetivos de alto nivel hasta el último detalle de bajo nivel en el momento adecuado, mientras se mantiene la

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

consistencia y el enfoque, la integridad y la priorización de los requisitos establecidos en su conjunto.

La tradicional disciplina de Análisis de Negocios de Ingeniería de Requisitos sigue siendo adecuada y se puede aplicar de una manera ágil. Ingeniería de Requisitos reconoce que cada requisito tiene un ciclo de vida propio, a partir de su elicitación inicial, a través de su análisis y validación para su eventual incorporación en la solución, o su rechazo o reducción del alcance”.

“...Exploramos cómo el AgileBA asegurará que los requisitos constituyen un todo consistente y coherente. Vemos cómo se capturan y registran adecuadamente los requisitos ágiles, a medida que evolucionan y se expanden en detalle. También consideramos el uso de prácticas ágiles clave en este proceso”. ([] (2014, febrero): *Agile Business Analysts, AgileBA*, <www.dsdm.org>)

Se recuerda que la elicitación de requerimientos abarca por un lado el contexto del sistema y otro el origen de los requerimientos, el que puede venir de los stakeholders, de la documentación de aplicaciones anteriores o de sistemas legacy (sistema legacy o sistema heredado, es un sistema informático, equipo informáticos o aplicaciones, que aunque resulte anticuado continúa siendo utilizado por el usuario, generalmente una organización o empresa, y no se quiere o no se puede reemplazar o actualizar de forma sencilla).

La ingeniería de requerimientos debe obtener las metas y los objetivos de los requerimientos desde sus orígenes. No identificar adecuadamente los requerimientos, puede impactar negativamente en el proyecto.

En el capítulo del AgilBA, antes mencionado, se agrega:

“Elicitación es la identificación, extracción y captura del requerimiento. En un proyecto Agile esto es típicamente alcanzado por:

- Conversaciones face-to-face
- Observación
- Workshops facilitados
- Demostraciones de elementos trabajados de la solución

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Escenarios
- Modelar y Prototipar” ([1] (2014, febrero): *Agile Business Analysts, AgileBA*, <www.dsdm.org>)

DSDM Members



| | |
|-------------------------------|-------------------------------------|
| AIB Bank | KnowledgePool |
| Alto Limited | Lloyds Bank |
| Aspective | Merrill Lynch |
| Bradford & Bingley | NatWest Bank |
| British Airways | OFSTED |
| BT | Orange |
| CMG (UK) | Parity Group |
| Computas AS (Norway) | Pricewaterhouse Coopers |
| Department of Health | Saucon Technologies (U.S.A.) |
| EDS | Storebrand (Norway) |
| Halifax Plc | Syntegra |
| Hewlett Packard | TNT Express Worldwide |
| ICL | Vodafone |
| John Lewis | VSO |

Fig. 2: Logotipo de DSDM Consortium y DSDM Members
Fuente: Elaboración propia sobre *Why DSDM?* <www.dsdm.org> (2002)

Proyecto para el Desarrollo de un Proyecto de Software

El Proceso Unificado de Desarrollo de Software (1999) de Ivar Jacobson, Grady Booch, James Rumbaugh, resume las cuatro "P" en el desarrollo de software, Personas, Proyecto, Producto y Proceso, en otras palabras, de la siguiente manera:

- Personas:

Son los principales autores de un proyecto de software, estos son arquitectos, desarrolladores, ingenieros de prueba, el personal de gestión que les da soporte y también los usuarios, clientes y otros interesados.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Proyecto:
Es el conjunto de actividades interrelacionadas y coordinadas para alcanzar algún objetivo específico dentro de los límites que impone un presupuesto. El resultado de un proyecto es una versión de un producto.
- Producto:
Son los artefactos que se crean durante el período de vida del proyecto, estos son los modelos, el código fuente, los ejecutables y la documentación.
- Proceso:
Es el conjunto de actividades necesarias para transformar los requisitos de usuario en un producto. Es una plantilla para crear productos.

El Ing. Arce Riobóo José Carlos, Mg.en Ingeniería Estructural DEPMI de la UNAM, Certificate Program en Business Administration de la Universidad de California, Berkley, expresa que, en la mayoría de las organizaciones, la forma de conseguir recursos y/o presupuestos suplementarios es mediante “proyectos”. Luego de algunos años ese dinero es empleado en trabajos habituales, pero todos los años se sigue pidiendo ese presupuesto, como si el trabajo fuera un proyecto cuando ya no lo es.

Riobóo, en *La Diferencia entre un Proyecto y un Proceso*, que se publica en <http://www.keisen.com/article62.html> de Keisen Consultores, aclara: “Por otro lado, hay muchos gerentes de proyecto que ven todo como proyecto: implementar un nuevo sistema de administración de proyectos, es un proyecto; desarrollar un nuevo producto, es un proyecto. Pero hay trabajos que están consumiendo recursos que se deben de dedicar a proyectos reales que se utilizan en el otro tipo de trabajo: procesos de negocio. Es importante saber distinguir cuál es cuál”.

Hay dos formas en que puede llevarse a cabo el trabajo:
Por procesos de trabajo o por proyectos.

Agrega Riobóo: “Todo lo nuevo o lo mejorado que ocurre es un proyecto. Todas las operaciones rutinarias requieren de procesos de trabajo.

Un proyecto es un esfuerzo temporal que produce un producto, un entregable o un servicio (en la amplia definición de los mismos) únicos”.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Un proyecto es desarrollado por un lapso de tiempo y para realizarlo se necesita un equipo que construye y ejecuta un plan. Al terminar el proyecto, logrado el producto, puede ocurrir que el equipo se desintegre y sus miembros vuelvan a sus lugares de origen o que el equipo siga trabajando y el proyecto empiece de tiempo en tiempo, por lo que deja de ser un proyecto para constituirse en un proceso. En cambio, un proceso es un esfuerzo que puede llegar a ser permanente en el sentido de que produce un entregable o un producto repetitivo (aunque sea pasible de cambios, los que se manejarán por un proceso o parte del mismo). Toda vez que se realiza el proceso, se obtiene aproximadamente el mismo producto o output.

Tanto el proyecto como el proceso son un conjunto interrelacionado de actividades que transforman insumos en productos o entregables (agregándoles valor). La diferencia radica en que para el proyecto, en ese conjunto de actividades interrelacionadas y dependientes entre ellas, conviven actividades distintas, diferentes, únicas, que tal vez nunca se hayan realizado antes en la organización y que incluso pueden ser absolutamente innovadoras, es decir que no se relacionan a lo que ya se hizo antes. Mientras que los procesos u operaciones son un conjunto de actividades, que también se interrelacionan y son dependientes pero que ya se hicieron antes, de forma similar o parecida, al menos una vez en la organización.

Añade Arce Riobóo: “Ambos, los procesos y los proyectos requieren de administrarse, pero solo sabiendo la diferencia entre ellos podremos elegir las técnicas y herramientas administrativas necesarias para llevarlos a cabo.

Los proyectos requieren de administración de proyectos, pero los procesos requieren de administración de procesos”.

Luego expresa: “Utilizar administración de proyectos para administrar la operación de toda una organización es altamente ineficiente”. Por ejemplo: si cada vez que una empresa de entregas va a transportar un paquete, reuniera a un equipo de proyecto para realizar un plan y ejecutar y monitorear las actividades del envío, desperdiciaría esfuerzos, porque ya se conocen los requerimientos del cliente y las tareas que se necesitan realizar para hacerlo, aun cuando se modifiquen en cada oportunidad el destino, el peso del paquete, etc. Y continúa: “Se tienen un costo y un calendario para el

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

proceso y es parte del presupuesto anual (de operaciones) y del ejercicio de planeación anual (que también es un proceso, no se empieza de cero y aunque el plan sea distinto, la manera de hacer el plan es también un proceso). Se tienen personas que realizan los trabajos de ese proceso. No se necesita administración de proyectos”.

Esto es claro pero”...La confusión surge cuando hay procesos que realizan los mismos outputs o productos o entregables en versiones que “se hacen a la medida”. Aunque el proceso utiliza la misma serie de conjunto de trabajos y pasos cada vez, con las mismas personas, el producto o entregable puede variar”. Por ejemplo una revista es un macro proceso que se realiza a la medida. En cada edición la revista mantiene el mismo formato pero distinto contenido, sin embargo producir una revista no es un proyecto, es un proceso: “La revista es un esfuerzo no temporal, es decir es permanente, recurrente repetitivo por lo que debe de ser predecible y confiable cada vez que se realice. Reproduce un output o entregable repetitivo (la revista) cada vez que el proceso se lleva a cabo (una vez al mes). Hay trabajos asignados definidos para las personas que trabajan en la editorial y en el proceso de producción. Es un proceso de negocio customizado o hecho a la medida y se debe de administrar con las técnicas de administración de procesos, mejora continua, ISO, etc. , con las métricas definidas, un presupuesto de proceso definido, con descripción de trabajos definidos, diagramas de flujos, políticas de proceso definidas, etc., etc..”

Comentado esto es necesario saber qué debe tenerse en cuenta en un proyecto para el desarrollo de un producto software. Se sabe que es importante la escalabilidad del software, es decir, su habilidad para reaccionar a cambios y poder adaptarse a estos, manejar el crecimiento, estar dispuesto para hacerse más grande pero sin perder calidad. Por otro lado, es necesario, para cualquier proyecto de negocio, poder responder tres preguntas básicas: qué, cuándo y cuánto?, considerando las características del producto buscado, el plazo de tiempo necesario para la entrega del mismo y cuál será el costo para su obtención.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Imprescindibles para el desarrollo de un proyecto software

- Que el analista y el cliente se reúnan sobre las expectativas del proyecto y se pongan de acuerdo en los puntos de interés para su desarrollo.
- Estimar los recursos requeridos para acometer el esfuerzo de desarrollo de Software, esto simula a una pirámide donde las Herramientas (hardware y Software) son la base que proporciona la infraestructura de soporte al esfuerzo de desarrollo, en segundo nivel de la pirámide se encuentran los Componentes Reutilizables y en la parte más alta se encuentra el recurso primario, las personas: el Recurso Humano.

DSDM va más allá

Para demostrarlo se enumeraran ahora algunos de los principios de DSDM, los que se profundizarán en el CAP. III:

- Los recursos deben ser empleados para el desarrollo de las funciones de mayor valor para el negocio.
- La entrega del proyecto debe ser en el plazo preestablecido, respetando el budget y con alta calidad.
- Las necesidades del negocio cambian constantemente.
- Tanto las técnicas de Desarrollo como las de Gestión de proyecto están incluidas en DSDM.
- Además de desarrollar nuevos SI, DSDM puede ser usado también en proyectos de ampliación de sistemas IT actuales, incluso en proyectos de cambio no específicamente IT.
- La evaluación de riesgos debe centrarse en entregar función de negocio, no en el proceso de construcción.
- La estimación debe basarse en la funcionalidad del negocio en lugar de líneas de código.

DSDM desarrolla también, afianzando la importancia de estos y otros principios, un Business Case y un Business Vision.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Business Case o Caso de Negocio:

consiste del análisis de una propuesta y la comunicación objetiva del mismo al cliente, con los elementos necesarios para la toma de una decisión.

Se presenta en un documento escrito bien estructurado, a veces también en forma de un argumento verbal o presentación.

- Usando las técnicas de DSDM mejora la probabilidad de que los proyectos se entreguen dentro del budget.
- La solución final tiene más posibilidades de cumplir con los requisitos de negocio reales de los usuarios.
- Los usuarios resultan mejor capacitados debido a que sus representantes han definido y coordinado la formación.
- La implementación es más suave debido a las relaciones que se establecen entre los stakeholders.

Business Vision:

Es una visión de negocios. Se trata de cómo una empresa o una organización, proyecta el desarrollo futuro de la misma.

Para poder innovar en un sistema, las empresas, como las personas, requieren tener una visión a largo plazo de qué acontecería en función de alguna innovación. Muchas empresas no han definido planes a largo plazo y sufren luego las consecuencias de la imprevisión. Es importante prever una estrategia política ante distintas factibles contingencias, para plazos corto, mediano y largo, considerando en cada caso que la innovación, por el cambio de un servicio, será sostenible con alto riesgo por los altos niveles de incertidumbre que conlleva.

El alto riesgo financiero es tal vez el mayor de los obstáculos para la mayoría de las organizaciones. Los beneficios económicos son difíciles de predecir.

Enfoque en el Proyecto o Enfoque en el Producto

A continuación se plasmarán opiniones de mundialmente reconocidos consultores, sobre la conveniencia o no de usar Scrum y/o DSDM ante un proyecto de software. Todos ellos remarcan que hay un factor importante a destacar, que no los rivaliza, por el contrario los une, y es que Scrum en general no lo puede cubrir por sí sola: Scrum tiene un enfoque más centrado en el producto y DSDM más centrado en el proyecto. Por eso, en muchos casos Scrum necesita de DSDM.

Scrum de Scrums - DSDM Atern

Escalabilidad

Escalabilidad es una propiedad deseable de un proceso software. Este término, como se dijo más arriba, se refiere a la evolución, al crecimiento del software.

MSDN Microsoft, a través de Visual Studio. Net en la página [http://msdn.microsoft.com/es-es/library/aa292203\(v=vs.71\).aspx](http://msdn.microsoft.com/es-es/library/aa292203(v=vs.71).aspx) expresa: “La escalabilidad debe formar parte del proceso de diseño porque no es una característica separada que se pueda agregar después. Al igual que con otras funciones de aplicación, las decisiones que se tomen durante las primeras fases de diseño y codificación determinarán en gran medida la escalabilidad de la aplicación.

La escalabilidad de una aplicación requiere una pertenencia equilibrada entre dos dominios distintos, software y hardware. Puede avanzar grandes pasos que aumenten la escalabilidad de un dominio sólo para sabotearlos cometiendo errores en el otro. Por ejemplo, la creación de un grupo de servidores Web con equilibrio de carga no beneficiará una aplicación Web que se ha diseñado para ejecutarse un solo equipo. De igual modo, el diseño de una aplicación altamente escalable y su implementación en equipos conectados a una red con poco ancho de banda no controlará bien las cargas pesadas cuando se sature el tráfico en la red...”

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Y agrega: “La escalabilidad es la capacidad de mejorar recursos para ofrecer una mejora (idealmente) lineal en la capacidad de servicio. La característica clave de una aplicación es que la carga adicional sólo requiere recursos adicionales en lugar de una modificación extensiva de la aplicación en sí.

Aunque el rendimiento marca una diferencia a la hora de determinar el número de usuarios que puede admitir una aplicación, la escalabilidad y el rendimiento son dos entidades diferentes. De hecho, las labores de rendimiento pueden ser opuestas a veces a las de escalabilidad”.

Escalar es un problema que enfrenta a proyectos más ágiles incluso en el ambiente Scrum, donde la solución propuesta es el concepto "Scrum-de-Scrums".

En cualquier organización, para resolver un gran proyecto, normalmente se lo particiona en proyectos de menor envergadura que resulten más manejables.

En los grandes proyectos desarrollados en cascada, se pueden tener varios equipos con una habilidad específica, por ejemplo: un proyecto contendría una serie de equipos que contengan sólo los desarrolladores con cada equipo responsable de un dominio particular.

En desarrollos Agile se crean equipos auto-organizados y multi-funcionales, restándose la dimensión de los equipos en habilidades específicas. Queda así un número de equipos de dominio específicos. Son estos equipos lo que hay que coordinar, lo que para Scrum resultaría en un: Scrum de Scrum de Scrum de.... En definitiva resultaría un Scrum de Scrums.

El concepto Scrum de Scrums evolucionó para hacer frente a esta situación. De cada equipo Scrum se elige una persona para representar al equipo en el scrum-de-scrums y coordinar las actividades entre los equipos. Esta es la teoría para escalar y permitir un Scrum-de-Scrum-de-Scrums y así siguiendo.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Jeff Sutherland, Maintenance Engineer en University of Saskatchewan, uno de los creadores de Scrum junto con Ken Schwaber, como ya se mencionó más arriba, define en Agile Alliance, <http://guide.agilealliance.org/guide/scrumofscrums.html>, su creación, el Scrum de Scrums, como: “Una técnica para escalar Scrum hasta grupos grandes (más de una docena de personas), consiste en dividir los grupos en equipos ágiles de 5 a 10 personas. Cada Scrum diario dentro de un sub-equipo termina mediante la designación de un miembro como “embajador” para participar en una reunión diaria con los embajadores de otros equipos, llamada Scrum de Scrums.

Dependiendo del contexto, los embajadores pueden ser colaboradores técnicos o un Scrum Master de cada equipo o incluso directores de cada equipo.

El Scrum de Scrums procede de otra manera pero como una reunión diaria normal, con los embajadores informando terminaciones o próximos pasos e impedimentos en nombre de los equipos que representan. Se espera la resolución de los impedimentos para centrarse en los problemas de coordinación entre los equipos. Las soluciones para esta coordinación pueden implicar la aceptación de interfaces entre los equipos, negociar los límites de responsabilidad, etc.

El Scrum de Scrums hará un seguimiento de estos elementos a través de un backlog propio, donde cada elemento contribuye a la mejora de la coordinación entre los equipos”. Y agrega luego, ante ciertas supuestas ideas, que: “Puesto que definí originalmente el Scrum de Scrums puedo definitivamente decir que el Scrum de Scrums no es un Meta Scrum. El Scrum de Scrums como yo lo he usado es responsable para la entrega de software trabajado de todos los equipos para la Definición de Dado en el final del Sprint o para realizaciones durante el Sprint”. Y aclara que: “... la Meta-Scrum es un mecanismo de financiación”.

El tema es que, en las grandes organizaciones empresariales este concepto, Scrum de Scrums, no inspira confianza.

En la opinión de Mattheu Caine, Partner at M.C. Partners & Associates (Agile Entrepreneur), Managing Director of the Agile Academy Switzerland - Global Head of GTS eCommerce at Deutsche Bank, en *Agile Project Management Scaling*, publicado en <http://www.mcpa.biz/2012/03/scaling-agile-projects/> en marzo de 2012, Agile

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Project Management, basado en DSDM Atern, resuelve este problema como sigue: Además del rol de Project Manager (Director de Proyecto), Atern agrega un Business Visionary (Visionario de Negocios) y un Technical Coordinator (Coordinador Técnico) del proyecto. Los dos últimos son compañeros y responsables de la solución de negocios, en su totalidad, y de mantener la tecnología elegida aplicada en forma coherente.

Así, un individuo tiene una visión general de cómo la solución se mira desde un punto de vista comercial y otro individuo mantiene alineada la tecnología. Son también los encargados de adoptar decisiones definitivas de las escaladas que ocurren en los equipos. Por lo tanto, son los responsables últimos.

Estos roles se destacan a continuación, en un proyecto que tiene un solo equipo:

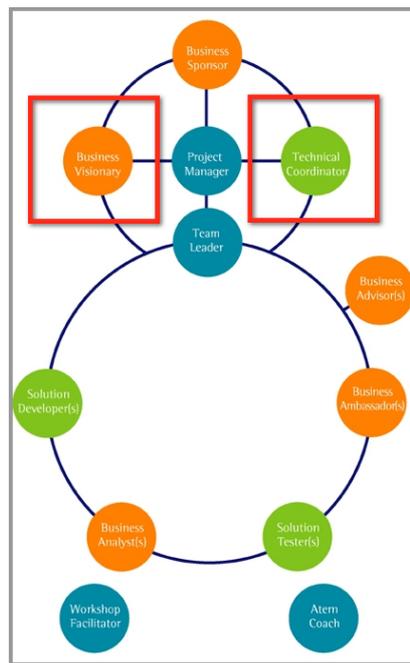


Fig. 3: Esquema de los roles en un equipo que combina Scrum con DSDM Atern

Fuente: *Scaling Agile Projects*. < <http://www.mcpa.biz/2012/03/scaling-agile-projects/> >

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Si un proyecto tiene, por ejemplo, tres equipos, cada uno responsable de un dominio diferente, estos tres equipos tendrían un Business Visionary común y un Technical Coordinator, como se muestra a continuación:

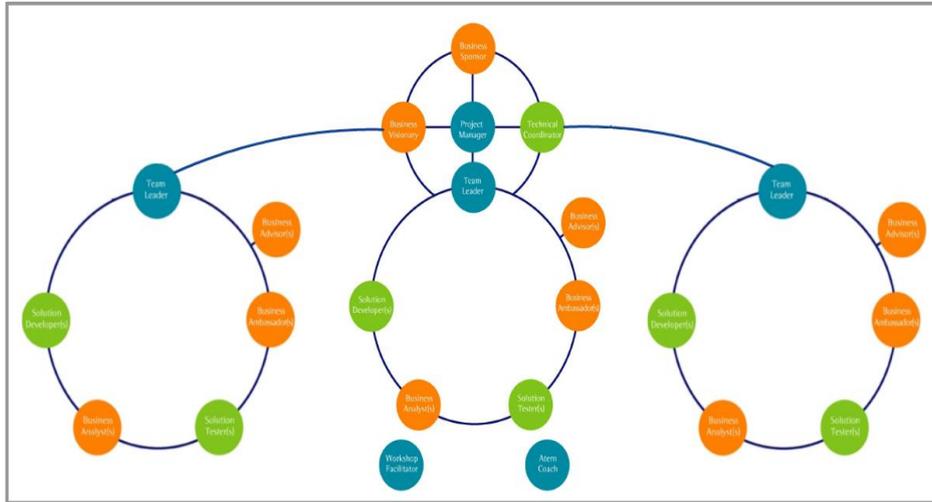


Fig. 4: Esquema de los roles en 3 equipos combinando Scrum con DSDM Atern

Fuente: *Scaling Agile Projects*.< <http://www.mcpa.biz/2012/03/scaling-agile-projects/>>

Como los roles están "incorporados en" el método, no necesitan ser creados artificialmente, y en grandes proyectos estos roles se convierten en trabajos a tiempo completo.

Qué, Cuándo y Cuánto?

Se resumen ahora los pasos del Scrum Process, extractados de *The DSDM Agile Project Framework for Scrum* (2007:3), producido por Andrew Craddock, Director Técnico de DSDM Consortium, en coautoría con Barbara Roberts, Keith Richards, Julia Godwin y Dorothy Tudor, miembros del Technical Team en el DSDM Consortium, para DSDM Consortium, bajo la revisión de José Casal-Giménez, Nick Clare, Mischa Ramseyer, miembros de la comunidad Scrum:

- "Un Product Owner, que representa al cliente, define un conjunto ordenado de requisitos para el producto que será construido, el Product Backlog (lista de requisitos de alto nivel priorizados, que definen el trabajo a realizar): artefacto

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

dinámico que evoluciona con los cambios necesarios de la empresa para la que se construyeron los productos”.

- “El Equipo Scrum (comprendido por el Product Owner, el equipo de desarrollo y el ScrumMaster, quien gestiona y facilita el desarrollo del proceso por parte del equipo) utiliza sus conocimientos, experiencia, entendimiento del producto y los requisitos para acordar un número de los artículos más importantes del Product Backlog para abordar en un Sprint (tiempo en que se realiza una iteración)”
- “En el proceso del Sprint Planning (planificación del Sprint), el equipo de Scrum colabora para averiguar cómo van a representar los elementos que han seleccionado desde el Product Backlog en la entrega del Potentially Releasable Product Increment al final del Sprint. El resultado del Sprint Planning Meeting (reunión de planificación del sprint) es el Sprint Backlog (objetivo a alcanzar en un sprint) que define, a diferentes niveles de granularidad, los trabajos necesarios para entregar el Incremento”.
- “La duración del Sprint se acordó desde el principio y, cada día, el equipo mantiene una reunión de 15 minutos llamada Scrum. El propósito principal del Scrum es permitir al Equipo Scrum sincronizar actividades, identificar obstáculos y acordar un plan para las próximas 24 horas”
- “El Sprint termina con una Sprint Review Meeting (reunión de revisión del Sprint), donde se demuestra el Product Increment (incremento alcanzado del producto) y una retrospectiva en la que el equipo reflexiona sobre la eficacia de la forma en que trabajó”
- “Un Scrum Master está presente en todo para ayudar a mantener el Equipo Scrum centrado en su objetivo y para ayudarles a sacar el mayor provecho de los Scrum Events (Sprint Planning Meeting, Daily Scrum (reunión diaria para revisar el estado del proceso), Sprint Review y Retrospective (para analizar la forma en que se trabajó))”

Y además dice: “Scrum tiene un enfoque de producto en lugar de un enfoque de proyecto por lo que pone más énfasis en la Incremental Release de un producto, en el contexto de un Product Lifecycle, que se focaliza en la tarea de terminar formalmente el desarrollo después de un período de tiempo acordado. Efectivamente, el desarrollo

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

termina cuando el valor a ser entregado por el siguiente incremento del producto no se puede justificar desde una perspectiva empresarial”.

En la Introducción de *DSDM Agile Project Framework for Scrum* (2007:3) se expresa que: “Este libro de bolsillo describe una versión del Agile Project Framework diseñada específicamente para complementar Scrum. Incorpora principios de DSDM enfocados a proyectos, junto con un rico conjunto de roles y responsabilidades de DSDM que son ideales para un entorno de proyecto empresarial. También ofrece las técnicas robustas y totalmente ágiles de DSDM para establecer y demostrar el control de un proyecto. A nivel de entrega, el Agile Project Framework utiliza una versión simplificada del proceso de DSDM, con el fin de trabajar al lado y complementando el proceso de entrega del producto Scrum, sin la necesidad de un cambio de enfoque. El DSDM Agile Project Framework reúne la fuerza de DSDM a nivel de proyecto y la simplicidad racional de Scrum en el nivel de equipo de entrega.

Uno o más aspectos del *DSDM Agile Project Framework for Scrum* se pueden utilizar para complementar Scrum en una base de proyecto por proyecto o una organización puede optar por aplicar algo o todo esto como un estándar.

En cualquier caso, los aspectos del DSDM Agile Project Framework deben utilizarse donde hacen el uso de Scrum más fácil, más eficaz o simplemente más accesible para el proyecto y stakeholders organizacionales”. (Craddock A et al, 2007:2)

Scrum no se ocupa de las cuestiones más básicas que un negocio exige: Qué, Cuándo y Cuánto, y que además necesita no tener que ocuparse de gestionar riesgos ni requerimientos. Es un enfoque simple e incompleto para una gestión de proyectos.

Esto se explicita en el *Agile Project Management* (2011:6) de DSDM Consortium y la APMG-International, con las siguientes palabras: “Scrum provee un excelente equipo con enfoque basado en permitir que el trabajo sea priorizado y entregado, usando el concepto de “backlog” de constante evolución para proporcionar la carga de trabajo del equipo. La fuerza de Scrum es su simplicidad, y puesto que es tan fácil de describir y empezar a usar, que esto ha impulsado su popularidad hasta la fecha.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Sin embargo, en Scrum, no existe el concepto de proyecto, simplemente un Product Backlog de trabajo a realizar. Para aquellos que deseen escalar Scrum para trabajar como un enfoque ágil en toda la empresa o utilizarlo para la gestión de proyectos y lanzamientos, por lo general es necesario trabajo significativo extra para superponer el concepto del project / release en el proceso básico Scrum. Scrum ofrece una versión muy simple de Scrum de toda la empresa ("Scrum de Scrums"), pero en el complejo mundo empresarial, hay poca confianza en la aplicación práctica con éxito de esto.

El resultado final es a menudo que las organizaciones complejas usan Scrum, pero necesitando un enfoque Agile corporativo fuerte, terminando por reinventar la información y orientación ya disponible en DSDM Atern.

En proyectos de IT, Scrum es a menudo vinculado con XP. Scrum proporciona el proceso de gestión del equipo y XP proporciona las técnicas de desarrollo.

Scrum también se combina a menudo con DSDM Atern, donde Scrum se utiliza a nivel de equipo de desarrollo y DSDM Atern se apoya sobre el equipo para posicionar el trabajo dentro de un proyecto y proporcionar los elementos de gestión de proyectos".([[: *Agile Project Management*, 2011:6)

Scrum es muy popular y se utiliza en muchas organizaciones, las que necesitaron llenar el vacío que les quedó cuando sustituyó la gestión tradicional de proyectos. Es por esto que:

- El PMI reconoce que las técnicas ágiles necesitan ser entendidas por la gestión del proyecto
- La APMG (fundadores de PRINCE2) ha fusionado PRINCE2 con DSDM Atern para formar el método Agile Project Management y certificación.

Donde, como se sabe, PMI es el Project Management Institute, organización internacional sin fines de lucro que asocia a profesionales relacionados con la Gestión de Proyectos, cuya página es www.pmi.org y que sirve a profesionales y organizaciones con estándares que describen las buenas prácticas, proporciona credenciales reconocidas a nivel mundial que certifican experiencia en gestión de proyectos y recursos para el

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

desarrollo profesional, networking (actividad socioeconómica de negocios mediante la cual, grupos de empresarios de reconocida mentalidad, crean o actúan sobre las oportunidades de negocio) y comunidad.

En tanto que APMG-International es uno de los principales institutos de examen que acredita a organizaciones de capacitación y consultoría, y gestiona los sistemas de certificación para los profesionales, estando acreditada con los más altos estándares UKAS para trabajar con PRINCE2 , MSP, M_o_R, DSDM Atern – Agile Project Mgmt, etc.. Su página es <http://www.apmg-international.com/>

Scrum - Agile Project Management

El Software no es ingeniería tradicional ni es investigación

Lo que sigue se basa en la opinión de Matheu Caine, en su publicación de setiembre de 2011 en Agile Project Management , DSDM Atern , Gestión , Scrum | Comments Off en www.mcpa.biz M.C.Partners & Associates.

Matthew Caine trabajó en Logica, Swiss Re, Avaloq e Infonic. Experimentó las principales funciones de software, apoyo al análisis, desarrollo, consultoría, ventas de gestión de la oferta, ventas de gestión de grandes co-localizado y equipos virtuales además de programas de mejora de procesos, como la adopción de DSDM Atern en Infonic.

Matthew Caine , en otras palabras, expresa que:

Disciplinas de gestión de proyectos existen desde hace muchos años.

El entrenamiento en gestión de proyectos en la década del 90 era considerado como una Técnica en Gestión de Proyectos y estaba dado por Gestores de Ingeniería de Proyectos. Se trataba de proyectos pre-deterministas en el sentido de que la fecha determinada, el camino crítico, etc. podrían ser manejados fácilmente.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

También era fácil de realizar un análisis del desempeño de redes, examinar el diagrama de Gantt, las tareas de choque, etc. Los errores en estos casos se solían adjudicar al ser humano (por ejemplo por la comunicación) o a fenómenos naturales (terremotos, inundaciones).

Lo contrario a un proyecto pre-determinista es la investigación. Dado un presupuesto, una voluntad y una visión, hay que explorar cómo se podrá concretar la visión. No siempre se puede alcanzar la meta deseada, pero en muchos casos, en esa búsqueda se llega a otros descubrimientos, hasta por accidente (ejemplos famosos incluyen 3M Post-It Notes, Viagra, la sacarina, la plastilina, etc.).

Pero el software no es ingeniería tradicional como tampoco producto de la investigación.

En los últimos 10 años, los desarrolladores de software han comenzado a adoptar metodologías ágiles que mejoran en tiempo, en budget, en calidad y en cuestiones de alcance. Pero no siempre con mucho éxito, aunque los enfoques ágiles sin duda conciben una mejor manera de la entrega de software.

Entregar software es, obviamente, sólo un aspecto de un negocio. El software es sólo el facilitador que permite al negocio llegar a los clientes de forma más fácil, más rápida, consistente y fiable. Amazon, por ejemplo, está en el negocio de hacer dinero de la venta de bienes, libros, pero, es el software el que lo hace posible. El software es la técnica que habilita y Agile ayuda a crear un mejor software.

“Pero el tema ahora es el negocio, que todavía necesita saber de IT: Cuándo se conseguirá, qué y cuánto va a costar”. (Caine Matheu, 2011)

“Los jóvenes especialistas del software, los principales motores de Agile, en general, no se dan cuenta de esto” (Caine Matheu, 2011)

Los desarrolladores de software no pueden ignorar, por ejemplo, las necesidades de otros departamentos en el negocio. Las áreas de un negocio tienen que estar en línea con

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

el ritmo de entrega de los envíos de software. Esto es lo que se llama “Maximizar alineación Agile”.

En concreto, Matheu Caine ejemplifica:

- “Marketing y Ventas necesitan saber qué está en la “hoja de ruta” de forma de poder iniciar el mercadeo, esto es qué funcionalidad y a cuáles clientes enfocar (target), ver si están interesados en pagar (al menos parcialmente) por las características”. (Caine Matheu, 2011)
- “Los clientes de Productos de Software (tales como, soluciones de core-banking) necesitan saber qué van a conseguir y cuándo para que ellos también pueden programar proyectos para mejorar y tener su propia gente de venta enfocados a sus clientes propios”. (Caine Matheu, 2011)
- “Servicios profesionales necesita saber que, cuando están en el sitio, se les informa con antelación, para que puedan promover o responder a solicitudes de los clientes sin tener que volver a la casa-oficina e interrumpirlos y volver a la oficina”. (Matheu Caine, 2012)
- “Soporte necesita saber cómo ellos pueden responder preguntas de los clientes existentes como de Account Managers”. (Caine Matheu, 2011)
- “La alta dirección necesita saber qué necesitan presupuestar todavía y ver el Cash Flow (flujos de efectivo). No sólo que podrían estar preparando el plan de inversión o el cálculo del ROI en nuevos productos”. (Caine Matheu, 2011)

En definitiva, es necesario, para poder cumplir los planes generales para el resto de la empresa, que se les informe de manera proactiva sobre qué viene, y cuándo. La administración también tendrá que saber cuáles son los riesgos y cuánto va a costar.

“Para ser ágil y atender las necesidades de la empresa, tenemos que combinar las técnicas ágiles y disciplinas tradicionales de gestión de proyectos. Esto es posible y las dos organizaciones más populares de gestión de proyectos están alentando el uso de técnicas ágiles”. (Caine Matheu, 2011)

El DSDM Agile Project Management abarca los valores de ejecución de proyectos con los fundamentos de DSDM y alinea plenamente con la filosofía de desarrollo de productos inherentes a Scrum.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

DSDM sostiene, como parte de su filosofía, que cualquier proyecto debe alinearse con la táctica del negocio y abocarse a la pronta entrega realmente provechosa para el negocio.

Esto puede lograrse cuando los stakeholders comprenden claramente los objetivos del negocio, se los capacita a un buen nivel y se disponen a colaborar para alcanzar una solución adecuada. La misma será entregada dentro de los plazos convenidos, de acuerdo a las prioridades indicadas por el negocio. En la proporción que el proyecto avance, los interesados tienen que aceptar que el cambio es ineludible y comprender que se ahonda la solución.

Siguen otras opiniones sobre DSDM vs Scrum

¿Por qué preferir DSDM a Scrum?

“¿Por qué preferir DSDM a Scrum?” es una pregunta que formuló Richard Robinson (2010, mayo), propietario de Hames Robinson Associates, cuya pagina es <http://hamesrobinson.blogspot.com.ar/2010/05/why-prefer-dsdm-to-scrum.html>.

Robinson es especialista en negocios y en IT governance, gestión y aseguramiento para organizaciones, proyectos y programas de toda clase. Trabajó en la mayoría de los sectores y para las más prestigiosas organizaciones mundiales, tales como: Accenture, Citibank, AIG, American Express, RBS, LloydsTSB, entre otras.

Robinson expresa que los profesionales saben que no es necesario elegir a Scrum o DSDM en forma excluyente. Se pueden emplear en un enfoque híbrido que incluya a ambos y que se adapte a sus necesidades. Pero, es importante que tengan una idea de qué los diferencia y lo que pueden aportar por separado y en forma integrada, para dar al cliente/usuario definiciones y criterios claros que ayuden a tomar una decidida elección “...esencial para la venta de cualquier forma de ágiles”, dice Robinson (2010, mayo).

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Por ende, en relación al tamaño, complejidad y carácter innovador del proyecto y considerando las dificultades de requisitos técnicos y reglamentarios, Robinson expresa, en otras palabras, algunas de las razones para preferir DSDM a Scrum:

En la etapa Fundación se necesita que todo el mundo tome conciencia de que se está haciendo, para no perjudicar el proyecto. “Agregar un 'Sprint 0 'no es la respuesta, salvo que se parezca mucho a Fundamentos de DSDM”. Robinson (2010, mayo).

DSDM es un proceso cierto durante el desarrollo de todo el proyecto. Esto es muy importante porque al inicio del proceso, en forma inmediata, se cometen errores.

La definición de los roles en la forma completa en que lo hace DSDM es siempre de gran ayuda, incluso para proyectos de escasa envergadura. Siempre se descuidó este ítem clave en IT, pero DSDM lo considera esencial

La definición de productos como BAD (BAD Sector Scan, freeware, aplicación para la exploración de sectores en dispositivos de almacenamiento), SAD (Sistema de Ayuda a la Decisión, es un sistema interactivo que ayuda al proceso de decisión de problemas mal estructurados), etc., es muy útil, para no depender de prototipos y de equipos facultados.

Agrega Robinson:” Como esta lista indica, DSDM es mucho más adecuado para un entorno corporativo, donde lo lúdico, la autonomía y la espontaneidad de Scrum choca más fuertemente con la demanda de definición formal, la rendición de cuentas y la previsibilidad. Es difícil obtener todos los beneficios de Scrum sin perder de vista formalidades corporativas dictadas en materia de gobierno y rendición de cuentas, priorización de inversiones, inteligibilidad y la visibilidad a un grupo más amplio de interesados, gestión de la calidad, elaboración de informes y la arquitectura”.

Anders Larson (2010), actualmente Solution Architech en Softronic y Chairman of the Swedish DSDM Consortium, colaboró con Robinson señalándole la comparación, que oportunamente se realizó, entre DSDM y Scrum desde el sitio de DSDM Consortium,

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

cuya autoría se atribuye a Andrew Craddock, Technical Director de DSDM Consortium, quien resumió la situación muy bien, después de enumerar los principios del Manifiesto para el Desarrollo Agile de Software, ya mencionados más arriba, al que contribuyeron y suscribieron profesionales de DSDM y Scrum:

- Gente e Interacciones sobre Procesos y Herramientas
- Trabajar Software sobre Documentación Exhaustiva
- Colaboración con el cliente sobre Negociación de Contratos
- Responder al Cambio sobre Seguir un Plan

Craddock señaló, oportunamente, que DSDM, en los cuatro principios, reconoce más que Scrum el peso de los ítems en la derecha, poniendo igual el valor más alto en los componentes de la izquierda. Y es por eso que DSDM encaja más fluidamente con la naturaleza de las grandes organizaciones, dando pie a las diferencias entre estas dos metodologías ágiles.

DSDM Scrum Roadmap

De la página de Orange Fortune, <http://orangefortune.com/DSDMScrumRoadmap.html>, se extractó el DSDM Roadmap, que se verá con detalle en el CAP. IV, publicado por Chris Fortuin quien aclara que: “El DSDM Scrum Roadmap identifica los temas críticos al combinar DSDM Atern y Scrum para que se pueda aprovechar al máximo la sinergia mediante la comprensión de la alineación, los solapamientos e incluso algunos elementos en conflicto”. (Fortuin, 2010)

Cabe aclarar que Orange Fortune es una empresa de Capacitación y consultoría en Agile Project Management Training usando Scrum, DSDM y PMI-ACP (PMI Agile Certified Practitioner, certificado que reconoce conocimiento de los principios ágiles, prácticas y herramientas y técnicas a través de las metodologías ágiles) y que Chris Fortuin, es M.Sc. (Magister en Ciencias), con las siguientes certificaciones internacionales: PMP (Project Management Professional, que otorga PMI y acredita sobrada experiencia para liderar y dirigir proyectos), PMI-ACP, PSM I & II (Professional Scrum Master, niveles I y II, que otorga Scrum.org), AgilePM / DSDM (DSDM Agile Project Management,

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

que acredita APM), CSM (Certified ScrumMasters, que entrega Scrum Alliance), CSP (Certified Scrum Professionals certificado que da Scrum Alliance).

Chris Fortuin es Gerente de Proyecto de Orange Fortune. Como Management Consultant, aprendió sus habilidades en gestión de proyectos en Ernst & Young, en los Países Bajos, usando DSDM y continuó entregando soluciones de software complejas en grandes y también complejas organizaciones.

Lean y PMBOK

Es oportuno, por lo que se verá enseguida, dar un breve esbozo sobre Lean y PMBOK

Lean:

Es una metodología de desarrollo de software, translación de los principios y prácticas de la manufactura esbelta, lo que es un modelo de gestión enfocado a la creación de flujo para poder entregar el máximo valor para los clientes, utilizando los mínimos recursos necesarios, esto es ajustados. Adaptado del Sistema de producción Toyota, cuya meta es eliminar los "desperdicios", distinguiendo siete tipos de posibles desperdicios:

- Defectos
- Exceso de producción
- Transporte
- Esperas
- Inventarios
- Movimiento
- Procesos innecesarios

Este sistema de producción es un ejemplo de la filosofía Kaizen (o mejora continua) de mejora de la productividad.

Para la metodología Lean, todo lo que no añade valor al cliente se considera un desperdicio:

- Código y funcionalidades innecesarias

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Retraso en el proceso de desarrollo de software
- Requisitos poco claros
- Burocracia
- Comunicación interna lenta

PMBOK:

Siendo la sigla de la Guía Project Management Body of Knowledge desarrollada por el PMI, es un conjunto de conocimientos en Dirección/Gestión/Administración de Proyectos generalmente reconocidos como «buenas prácticas», que se constituye como estándar de Administración de proyectos. Comprende dos grandes secciones, la primera sobre los procesos y contextos de un proyecto, la segunda sobre las áreas de conocimientos específicos para la gestión de un proyecto.

Volviendo a Chris Fortuin, se puede decir que hoy en día utiliza PM-BOK, PRINCE2, en combinación con Scrum, las prácticas de Lean y XP. Como Project Manager esto implica iniciación rápida y estilo de desarrollo de software iterativo e incremental de alta calidad. Como entrenador ofrece cursos de formación en Agile y presentaciones de la industria, compartiendo su conocimiento y experiencia en Agile Project Management, entrega de productos Scrum y Earned Value Management (EVM: Earned Value Management o Gestión del Valor Ganado), técnica de gestión de proyectos que consiste de un conjunto completo de indicadores de gestión para controlar la ejecución de un proyecto mediante su presupuesto y su calendario de desarrollo. Tales indicadores son por ejemplo BAC (Budget at Completion: Presupuesto previsto al inicio para todo proyecto), EAC (Estimate at Completion: previsión del coste total al terminar el proyecto de acuerdo a los datos actuales), CPI, SPI, VAC (Variation at Completion o Variación de la Terminación ($VAC=EAC-BAC$) que indica lo que nos hemos excedido del presupuesto autorizado), etc, indicadores de gestión financiera.

Fortuin afirma, en otros términos, que un gerente de proyectos Agile es responsable de la entrega de proyectos exitosos en entornos complejos. Dice que Scrum es la metodología ágil más común, pero no es un marco para el desarrollo y mantenimiento de productos complejos, pues el concepto de proyecto no existe en Scrum.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

La solución alternativa podría ser adoptar DSDM Atern. Este es el único enfoque ágil para centrarse en la entrega de proyectos de una manera totalmente Agile. Pero, para Fortuin, es aún mejor la combinación de los puntos fuertes de DSDM Atern y Scrum, para crear una sinergia que proporciona una metodología completa para la gestión de un proyecto Agile y entrega de productos impresionantes a tiempo y dentro del budget, en un entorno en constante cambio.

El consejo de Fortuin para un gestor de proyecto es que si se embarca en un proyecto Scrum utilice también DSDM Atern, para abordar temas de ejecución de proyectos tales como: términos de referencia, modelo de negocio, plan de implementación, plan de realización de beneficios, revisión de registros, etc. Y recuerda que DSDM Atern también incluye un Project Approach Questionnaire (PAQ), para gestionar los riesgos de la utilización de un enfoque ágil en una empresa. Finalmente, esto también lo preparará para lo que su PMO (PMO es la Oficina de Gestión de Proyectos, responsable de aplicar los conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del proyecto) requiera al utilizar PRINCE2 como la metodología general del proyecto.

Scrum & DSDM Atern: AND, not XOR

Nicola Hildred Hill, Senior Project Manager, News Technology en Thomson Reuters, dice:

“Soy algo así como un entusiasta de DSDM Atern, una versión reducida de la que se comercializa ahora como Agile Project Management. Parece que hay una creencia de que si tienes Scrum en el lugar no se puede utilizar DSDM Atern. Y ¿por qué querrías hacerlo?.

Hay algunas cosas que Scrum no hace ni hace explícito, que aún necesitan ser hechas en un contexto donde el dinero se invierte para producir un resultado esperado. En el momento de empezar con Scrum, se ha tomado la decisión de inversión, y el Product Owner y el equipo están mágicamente facultados para entregar, incluso antes de que hayan elaborado la cantidad de inversión (tiempo) que necesitarán. Tampoco hay

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

control explícito del proyecto (su rendimiento). Y la incapacidad de Scrum para proporcionar una fecha de lanzamiento puede ser un verdadero problema para algunas empresas: Si tiene que notificar a los clientes de un cambio con suficiente antelación, por razones contractuales o el tiempo de reserva con un proveedor subcontratado o incluso asegurarse de que tiene el producto listo a tiempo para el trade fair (canje comercial) en particular, es necesario un enfoque más firme para las fechas y el seguimiento. Todo esto es posible sin dejar de ser ágil, si agrega DSDM Atern". (Hill, 15 de mayo de 2012)

El diagrama siguiente, producido por RADTAC ilustra esto (RADTAC es una especialista en métodos ágiles. Provee entrenamiento y servicios de formación y consultoría para las organizaciones que buscan adoptar y aplicar las prácticas esbeltas y ágiles para entregar más rápido, más barato, mejores soluciones). Muestra cómo diferentes métodos ágiles pueden utilizarse en diferentes niveles operacionales. Es conveniente aclarar que RADTAC es una empresa que realiza coaching en la empresa y desarrollo de software, con clientes como British Airways y Friends Provident.

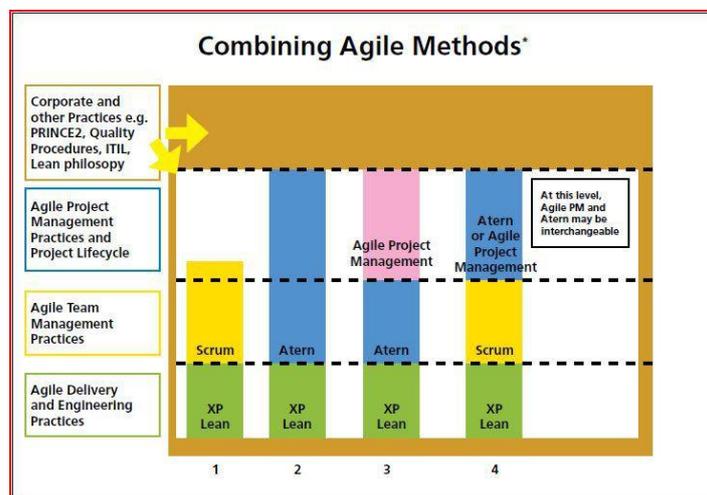


Fig.5: Diferentes métodos ágiles pueden utilizarse en diferentes niveles operacionales

Fuente: *Scrum & DSDM Atern: AND, not XOR* <<http://abithon.wordpress.com/2012/05/15/scrum-dsdm-atern-and-not-xor/>>/

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Compuertas de decisión y etapas de preparación

DSDM Atern cumple dos etapas, Feasibility (Factibilidad o Viabilidad) y Foundations (Fundación), antes de iniciar el trabajo de desarrollo, con tomas de decisión después de cada una de ellas, como forma de comprobar que vale la pena invertir en el proyecto.

Cada una deberá ser breve y pasarse rápidamente. Feasibility (Viabilidad) es exactamente eso, una breve evaluación de alto nivel de que el proyecto es factible y comprende: un caso de negocio, esbozo de solución y esbozo de plan. Una vez que se ha demostrado que la inversión vale la pena, la etapa de la Fundación define cómo será enfocado y controlado el trabajo, produciéndose el Product Backlog (Prioritised Requirements List), y los planes en un nivel de esquema que entrará en el número de sprints. Cuando todo esto se ha esbozado, una decisión puede ser tomada para proceder al desarrollo.

Decisión de la Jerarquía

DSDM Atern define tres niveles de jerarquía: equipo, proyecto y dirección. El equipo tiene la facultad de entregar y decidir sobre sus actividades directas, de la misma manera como para Scrum. El proyecto tiene la facultad de entregar el Business Case, tal como fue aprobado en Fundaciones. El grupo de dirección está para decidir sobre las excepciones al plan y no necesita que se le consulte si todo se va a planificar.

Usando MoSCoW para fijar timescales

La priorización de los requisitos en DSDM se reglamenta por MoSCoW, donde las letras significativas son M, S, C y W, las o sirven sólo para completar la palabra y las reglas para el éxito del proyecto son: Must Haves (Debe tener estos requisitos), Should Haves (debería tener estos requisitos), Could Haves (Podría tener estos requisitos), Won't Have (No va a tener estos requisitos).

Una diferencia clave en el enfoque DSDM es difundir el Must Haves a través de los sprints de entrega, en lugar de forward-loading como en Scrum. Cada sprint debe

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

contener 60% de Must Haves, 20% Should Haves y 20% Could Haves. Se puede comprometer una fecha fija de terminación. En el sprint, los Must Haves son entregados primero. Mientras cada sprint entrega sus Must Haves, el proyecto avanza según lo previsto, y se puede estar seguro de cumplir con su fecha límite. La desventaja de esto es que no se consigue antes el ROI y una organización necesita decidir qué es más importante para ellos: la entrega temprana de toda la funcionalidad más importante o la entrega a tiempo para satisfacer un caso de negocio acordado.

Un rol para Project Management

En la opinión de Nicola Hildred Hill: "Tal vez los jefes de proyecto no parecían ser muy útiles antes de Agile. La idea no es preservar un rol obsoleto para asegurar la continuidad del empleo, sino proporcionar una mejor certeza a la empresa de que lo correcto será entregado cuando sea necesario. Esto incluye hacer cosas tediosas como la gestión de riesgos. DSDM Atern fue diseñado para integrarse con PRINCE2 y por lo tanto proporciona un vínculo entre el control del proyecto completamente desarrollado (si se desea) y métodos ágiles de trabajo, sin comprometer ninguno de ellos. Mi observación de Scrum es que se trata de un montón de gente corriendo alrededor como gerentes de desarrollo, tratando de hacer la planificación, seguimiento y presentación de informes. Si se pone de nuevo la gestión del proyecto de una manera estructurada, esto será mucho más fácil". (Hill, 2012, 15 de mayo)

DSDM is Agile's Best Kept Secret?



Fig. 6: DSDM ¿es el secreto ágil mejor guardado?

Fuente: *DSDM is Agile's Best Kept Secret?*, <<http://agilescout.com/dsdm-is-agiles-best-kept-secret/>>

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Es cierto que DSDM se utiliza más en Europa, pero se ha ido introduciendo lentamente en Estados Unidos. Durante una conferencia de Agile Business, el propio Keith Richards habló sobre DSDM, remarcando lo que significa en la comunidad ágil. En sus comentarios sobre Scrum y XP señalaba que estos enfoques están más centrados en el producto y marcaba a DSDM como más centrado en el proyecto.

"Es necesario distinguir entre el término general 'ágil', la frase 'métodos ágiles' y el concepto de 'gestión de proyectos ágiles'. En el Reino Unido hay sólo 3 "métodos ágiles" reconocidos: XP, SCRUM y DSDM. DSDM sólo se puede utilizar "tal cual" para proyectos. Scrum y XP son técnicas de entrega del producto no tienen ningún concepto de proyecto tan definido por enfoques como PRINCE2". (Keith Richards, 2012)

Puntos de interés

Lo que sigue se trata de un resumen de la charla de Keith Richards por Elizabeth Harrin, creadora de A Girl's Guide To Project Management, publicación que se inició en 2006:

- "DSDM es diferente de la gestión de proyectos tradicional – se acuerdan el tiempo y el costo y luego varía el alcance acordado para asegurarse de llegar a esos objetivos".
- "El enfoque DSDM a tiempo y en presupuestos hace a ágil un muy rentable método - pero corre el riesgo de no conseguir todo lo que se quiera del proyecto en una primera vez".
- "DSDM utiliza un montón de facilitated workshops para mejorar el flujo de comunicación – la gente se involucra en el proyecto y ayuda a que las cosas avancen rápidamente".

Los beneficios

"Los Proyectos se entregan a tiempo. El enfoque crea propiedad. Es más fácil implementar y soportar soluciones. Pero toma mucho trabajo hacerlo bien. Mientras que el método es holístico, se necesita poder aplicarlo en forma inteligente."(Keith

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Richards). Donde la holística es aquello perteneciente al holismo (del griego ὅλος [*hólos*]: "todo", "entero", "total"), posición metodológica y epistemológica que postula que los sistemas, físicos, biológicos, sociales, económicos, mentales, lingüísticos, etc., y sus propiedades, deben ser analizados en su conjunto y no a través de las partes que los componen, considerando a éstas separadamente.

Defining Agile Quality: Scrum versus DSDM - June 14, 2012

Wernham Brian, consultor independiente, Project Governance Expert y autor de “Agile Project Management for Government” publicado por Maitland and Strong el 31 de Julio de 2012, opina, en otros terminos, que:

La terminología Agile puede variar de un método a otro, pero tanto el principio como la práctica son similares. Así como en DSDM, el trabajo de desarrollo se denomina "actividad de ingeniería", y la salida de cada iteración se llama la "solución emergente", en Scrum, la salida se denomina “incremento potencialmente liberable”.

En Scrum se habla de Product Backlog, mientras que en DSDM Prioritized Requirements List, pero cada elemento desarrollado, del Product Backlog o de la Prioritized Requirements List, necesita una definición de finalización cierta y precisa, para asegurar que se alcanza la calidad correcta.

En Scrum un incremento se establece como “Done” cuando se acuerda una “Definición de Hecho” para el equipo de Scrum, precondition de cada incremento, la que debe acordarse antes de que el equipo pueda planificar el desarrollo en el tiempo establecido.

Los requisitos más rigurosos en calidad necesitarán tanto más tiempo por ítem del product backlog y por lo tanto menos elementos deberían intentarse para esa iteración.

Lo que no está claro para Brian Wernham, al menos no especificado, en la Guía Scrum es cuándo se acuerda esta “Definición de Hecho”, lo que implica que los criterios de estricto sólo se deben acordar cuando agregan valor y no retrasan la solución. Se puede tomar la decisión sobre la calidad necesaria en el "último momento responsable" cuando la calidad / tiempo trade-offs se puede entender en base a los progresos realizados hasta la fecha y el feedback real de la implementación. Pero en DSDM, en cambio, el trabajo

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

de definición se llevó a cabo durante la fase de Fundaciones del Proyecto. Lo que puede acarrear sobre-especificación de calidad basada en suposiciones sin fundamentos.

La guía de DSDM advierte que cualquier "definición de hecho" por adelantado al trabajo, durante la fase de Fundaciones, deberá revisarse periódicamente a lo largo del ciclo de vida del proyecto, asegurando que no se cae en el 'Big Design Up-Front' (BDUF, Gran Diseño Inicial), característico de los enfoques de "cascada", antítesis de la agilidad.

Investigaciones sobre Metodologías Ágiles

Se ha realizado un análisis de las investigaciones para tesis de maestría y de doctorado sobre Metodologías Ágiles.

Se presentan algunas de las mismas en una tabla en el ANEXO III Tabla de Tesis de Maestría y de Doctorado sobre Metodologías Ágiles y sobre DSDM y Scrum.

No se encontraron tesis relacionadas basadas en el análisis comparativo de las Metodologías DSDM y Scrum.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

CAPITULO III - PLANTEAMIENTO DEL PROBLEMA

Ante un proyecto de software, el problema consiste en determinar, entre DSDM y SCRUM, cuál metodología elegir para su desarrollo.

En el CAP. II se han expuesto las diferencias esenciales entre ambas metodologías.

En este capítulo se profundiza en los fundamentos, roles, técnicas y artefactos de ambas metodologías y se detallan aspectos de DSDM Atern (Características Básicas, Principios y Productos), para resumir, en el CAP. IV, sus concordancias y diferencias, y concluir así la conveniencia o no de la aplicación de alguna u otra de las mismas, en función de las necesidades empresariales y el tipo de proyecto.

¿Qué es Scrum?

Bases de Scrum

- Priorización de los requisitos por valor para el cliente y coste de desarrollo en cada iteración
- El control empírico del proyecto
- La potenciación del equipo de desarrollo
- El timeboxing de las actividades del proyecto, donde **Timebox** es una técnica que consiste en fijar el tiempo máximo para conseguir unos objetivos, tomar una decisión o realizar unas tareas, y hacer lo mejor que se pueda en ese intervalo. La duración estimada de un timebox es menor a 4 horas.

Requisitos para poder utilizar Scrum

Cultura de empresa basada en trabajo en equipo, con características como: delegación, creatividad y mejora continua

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Compromiso del cliente en la gestión de los resultados del proyecto y del ROI (Return On Investment: es una razón financiera que compara el beneficio o la utilidad obtenida en relación a la inversión realizada, vale decir, es una herramienta para analizar el rendimiento que la empresa tiene desde el punto de vista financiero) y de la disponibilidad para colaborar con el equipo en forma directa durante todo el desarrollo del proyecto

Compromiso de la dirección de la organización para formar equipos autogestionados y multidisciplinarios, fomentando una cultura de gestión basada en la colaboración y en la facilitación a cargo de los líderes del equipo.

Compromiso conjunto y colaboración de los miembros del equipo.

Relación entre proveedor y cliente basada en ganar-ganar, colaboración y transparencia. Se aclara que los juegos ganar-ganar, en Dinámicas de Grupo, acentúan la importancia de la cooperación, diversión, distribución, el cuidado y sobre todo el éxito total del grupo en contraste con la dominación, comportamiento egoísta y mejora personal.

Desarrollo Iterativo e Incremental

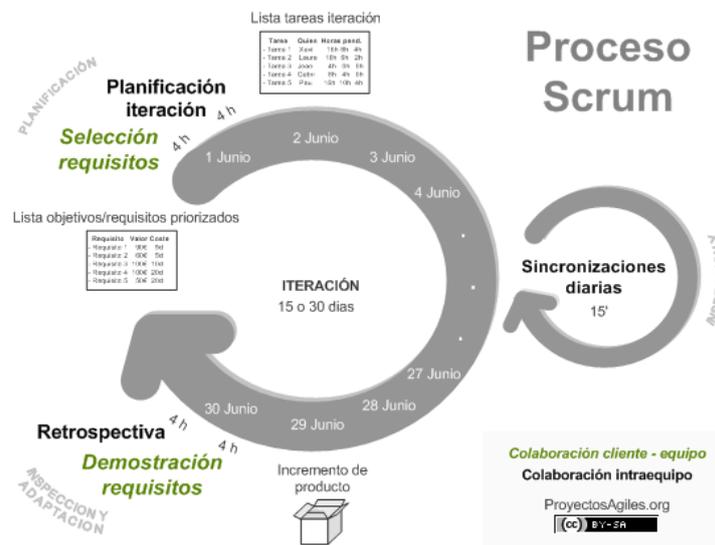


Fig. 7: Proceso Scrum

Fuente: *Qué es Scrum* <<http://www.proyectosagiles.org/que-es-scrum>>

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Sprint o Iteración:

El proceso se realiza en bloques temporales cortos y fijos llamados Sprint, de un mes y hasta de 2 semanas. Cada uno proporciona un incremento del producto final. Estas iteraciones se repiten como miniproyectos en los que el equipo debe realizar las tareas necesarias, incluyendo pruebas y documentación, para completar cada requisito en una única iteración, quedando preparado para ser entregado al Product Owner o Cliente, de forma tal que al final del proyecto no quede actividad arriesgada alguna relacionada con la entrega de requisitos.

Sprint Planning:

Cada iteración se planifica en una reunión, determinando un Timebox y en base a un Product Backlog o Lista de objetivos/requisitos del producto, proporcionada por el cliente y conformada al valor que aporta a su coste, consultada por el equipo previamente, según las dudas que surjan, para priorizar los objetivos/requisitos que se compromete a completar en cada iteración, de manera que puedan ser entregados si el cliente lo solicita.

Product Backlog Items:

Son los puntos destacados del Product Backlog, gestionables por la Scrum Taskboard o Pizarra de Tareas: tablero que actúa como radiador de información.

Sprint Backlog o Lista de Tareas de la Iteración:

Es elaborada por el equipo en el Sprint planning. Constituye un plan para completar los objetivos/requisitos seleccionados para la iteración, los que el equipo se compromete a demostrar al cliente al finalizar la iteración, en forma de incremento de producto preparado para ser entregado.

Scrum Daily Meeting o Reunión de Sincronización:

Se ejecuta diariamente en cada una de las iteraciones. Cada miembro del equipo inspecciona lo que está realizando el resto del equipo (relación de dependencia entre las tareas, evolución hacia la meta de la iteración, impedimentos para el logro del objetivo) para hacer las adaptaciones necesarias y poder cumplir con el compromiso adquirido.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

En toda reunión de sincronización, cada miembro, en un timebox de a lo sumo 15 min. tiene que responder:

¿Qué hizo desde la última reunión? ¿Si llevó a cabo su plan? ¿Cuál fue el inconveniente?

¿Qué va a hacer desde este momento?

¿Qué obstáculos tiene o tendrá en esta iteración y en el proyecto como para cumplir con su compromiso?

El Sprint Backlog sirve de apoyo a la reunión.

El equipo de desarrollo, guiado por el Scrum Master o Facilitador (en Roles se detallará su función), evoluciona hacia el producto final, haciendo una entrega incremental a partir de los resultados obtenidos en iteraciones anteriores, en los que fue agregando nuevos objetivos/requisitos o mejorando los ya completados.

Sprint Review o Revisión de la Iteración:

Es una reunión que se efectúa en el último día del Sprint y que consta de:

- Demostración: en un timebox 4 horas a lo sumo. Se presentan al cliente sólo los requisitos completados en la iteración, en función de los cuales y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, desde la primera iteración, replanificando el proyecto. Pero sólo se pueden mostrar los requisitos completados, para que el cliente no se haga falsas expectativas y pueda tomar decisiones correctas y objetivas en función de la velocidad de desarrollo y el resultado realmente completado. Un requisito no completado quedará como un requisito más a replanificar.
- Product Backlog Refinement o Grooming o Replanificación: que realiza el cliente, de manera objetiva, añadiendo requisitos, modificándolos, eliminándolos, repriorizándolos por valor y coste, cambiando el contenido de iteraciones y definiendo un calendario de entregas que se ajuste mejor a sus nuevas necesidades, de acuerdo a los resultados y los cambios que haya habido en el contexto del proyecto. Esto configura un fundamental soporte al control empírico, según su valor en ese momento y su coste estimado de desarrollo, pudiendo llegar a un punto en

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

que no valga la pena desarrollar los requisitos restantes, de acuerdo al poco ROI que tengan.

Sprint Retrospective o Restrospectiva:

Dura 4 hs. como máximo. En la misma el equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad.

Release Sprint o Iteración de Entrega:

Se añade cuando el cliente solicita una entrega de los objetivos/requisitos completados hasta ese momento. Es una iteración más corta que las iteraciones habituales, donde se realiza alguna tarea que no ha sido necesaria o posible hasta el momento de la entrega final y acaba de corregir defectos detectados en la última demostración.

Burndown Chart o Gráfico de Trabajo Pendiente:

Muestra el progreso de la iteración y su velocidad con respecto a tareas u horas pendientes. De la misma forma muestra el progreso del proyecto y su velocidad con respecto a requisitos completados.

Terminación anormal de la iteración:

Se puede solicitar sólo en situaciones excepcionales. Puede ocurrir que el contexto del proyecto haya cambiado enormemente y no sea posible esperar al final de la iteración para aplicar cambios o que el equipo encuentre que es imposible cumplir con el compromiso adquirido. Entonces se dará por finalizada la iteración y se iniciará otra mediante una Sprint Planning.

Roles de Scrum

Scrum Master o Facilitador:

Es el gestor del equipo, quien logra:

- Que todos los participantes del proyecto sigan las reglas y proceso de Scrum, acordes a la cultura de la organización,

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Guiar la colaboración intraequipo y con el cliente, aumentando al máximo las sinergías, consiguiendo:
 - Que el Product Backlog esté preparado antes de la siguiente iteración
 - Facilitar las reuniones logrando que sean productivas y alcancen sus objetivos.
 - Enseñar al equipo a autogestionarse. No dar respuestas, guiar al equipo con preguntas para que ellos mismos encuentren las soluciones.
 - Quitar impedimentos, protegiendo y aislando al equipo de interrupciones externas en la ejecución de cada iteración

Product Owner o Cliente:

Representa a todas las personas interesadas en los resultados del proyecto, internas o externas a la organización, promotores del proyecto y usuarios finales o consumidores finales del producto, actuando como interlocutor único ante el equipo, con autoridad para la toma de decisiones

El Team o Equipo de desarrollo del producto del proyecto:

Es responsable del trabajo que realiza y su calidad, en cada iteración y en el proyecto, como de la información.

Está integrado por 5 a 9 personas. Con menos de 5 personas se corre el riesgo de no cumplir con el compromiso adquirido. Con más de 9 integrantes se complican la comunicación y la colaboración entre todos los miembros, formándose subgrupos. Pero se puede hacer Scrum con 3 personas y también con 250 personas agrupadas en varios equipos distintos, con técnicas como Scrum de Scrums, ya mencionada en Cap.II.

Es multidisciplinar. Tiene las skills o habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc).

Definition of Done o Definición de completado:

El cliente y el equipo acuerdan juntos la Definition of Done de los objetivos /requisitos en el proyecto. Esta definición:

- Debe asegurar al cliente, en cada iteración, que el incremento del producto es potencialmente entregable, y que no hay tareas pendientes que puedan impedir

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

utilizar los resultados del proyecto lo antes posible. Así, cuando al final de cada iteración el equipo le haga una demostración de los requisitos completados, el cliente podrá tomar decisiones, como por ejemplo:

- cambiar las prioridades en función de la velocidad de desarrollo,
- solicitar una entrega del producto desarrollado hasta ese momento, etc.
- Debe incluir lo necesario para asegurar que el cliente obtendrá lo que necesita según sus criterios de entregables y de calidad: producto construido, probado, documentado, refactorizado para conseguir calidad interna/mantenibilidad, etc., donde Refactorización de código es una técnica disciplinada para la reestructuración de una entidad de código existente, alterando su estructura interna sin cambiar su comportamiento externo

Condiciones de Satisfacción:

Hay que asociarlas a cada objetivo, además de la definición de completado, en forma de casos de prueba de aceptación, al momento de crear el Product Backlog.

La definición de completado sirve como base, en el Sprint Planning, para identificar las tareas necesarias para conseguir cada objetivo/requisito. Para cada objetivo se crearán más (ó menos) tareas que la definición de completado y con mayor significado. Por ejemplo, “respecto a que el objetivo tiene que estar “construido”, pueden aparecer varias tareas, del estilo “construir el componente X”, “modificar la pantalla Y”, “modificar la BBDD”, “preparar el script de carga”, etc.”.

Métricas para Scrum

La métrica más importante en Scrum es el valor que se está dando al cliente. Permite conocer la velocidad con que retorna su inversión y si no es necesario seguir con el proyecto, cuando los beneficios pendientes de obtener ya no compensan sus costes.

¿Qué es DSDM?

DSDM significa Dynamic Systems Development Method= Método de Desarrollo de Sistemas Dinámicos. Pero en realidad no es un método en un sentido concreto, es un marco centrado en entregar una rápida solución de calidad.

Filosofía de DSDM

- El desarrollo es un esfuerzo conjunto de equipo: formado por el equipo de DSDM junto al usuario/cliente, combinando el conocimiento de los clientes/usuarios, en cuanto a los requisitos del negocio, con las capacidades técnicas de los profesionales de IT.
- El desarrollo puede ser incremental, efectivizándose entregas parciales, no es necesario entregar todo a la vez.
- Los recursos deben ser empleados para el desarrollo de las funciones de mayor valor para el negocio.
- Ningún sistema es construido a la perfección en el primer intento. Implementar la totalidad de requerimientos generalmente provoca que un proyecto supere plazos y presupuestos.
- La entrega del proyecto debe ser en el plazo preestablecido, respetando el budget y con alta calidad.
- DSDM requiere que cada paso del desarrollo se complete lo suficiente antes de que empiece el siguiente. Así una nueva iteración del proyecto puede comenzar sin tener que esperar que la anterior se complete íntegramente. Con cada iteración el sistema se mejora.
- Tanto las Técnicas de Desarrollo como las de Gestión de Proyecto están incluidas en DSDM Framework.
- Además de desarrollar nuevos SI, DSDM puede ser usado también en proyectos de ampliación de sistemas IT actuales, incluso en proyectos de cambio no específicamente IT.
- La evaluación de riesgos debe centrarse en entregar función de negocio, no en el proceso de construcción.
- La estimación debe basarse en la funcionalidad del negocio en lugar de líneas de código.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Principios de DSDM

| | |
|---|--|
| 1 | La participación activa del usuario es imperativa |
| 2 | El equipo debe estar autorizado para tomar decisiones |
| 3 | El foco está en la entrega frecuente de productos |
| 4 | La aptitud para el propósito del negocio es el criterio esencial para aceptar los entregables. |
| 5 | El desarrollo iterativo e incremental es necesario para converger a una solución exacta de negocios. |
| 6 | Todos los cambios durante el desarrollo son reversibles |
| 7 | Los requisitos están referenciados a un alto nivel. |
| 8 | El testeo es integrado a través del ciclo de vida. |
| 9 | Colaboración y cooperación entre todos los stakeholders es esencial. |

Tabla 7: Principios de DSDM

Fuente: *DSDM Tour* <www.dsdm.org>

Roles

| | |
|-------------------------------|--|
| Executive Sponsor | El Campeón del Proyecto |
| Visionary | Generalmente responsable de conseguir un proyecto ya iniciado |
| Ambassador User | Generalmente viene del área de negocio que está siendo tratada |
| Advisor User | Trae el conocimiento cotidiano del trabajo que es automatizado |
| Project Manager | Puede venir de la comunidad de usuarios o de IT |
| Technical Co-ordinator | Coordina los equipos individuales del desarrollo |
| Team Leader | Asegura que un equipo de desarrollo funcione en su conjunto |
| Developer | Modela e interpreta requisitos del usuario, y los convierten en prototipos y código entregable |
| Tester | Ejecuta la prueba del no usuario |
| Scribe | Anota en todas las reuniones, workshops para registrar requisitos, etc. |
| Facilitator | Independiente del equipo de proyecto, gestiona los workshop |
| Specialist Roles | Business Architect, Quality Manager, System Integrator... |

Tabla 8: Roles en DSDM

Fuente: *DSDM Tour* <www.dsdm.org>

Técnicas básicas

El uso de técnicas por un proyecto DSDM cae en dos agrupaciones principales: En principio las técnicas de proyectos que se llevan a cabo dentro de cualquier proyecto

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

usando DSDM sin importar el tipo específico de desarrollo. Estas incluyen Timeboxing, Moscow, etc.

En segundo lugar, las distintas técnicas de acuerdo al proyecto en sí, considerando normas existentes, soporte de herramientas de software y Best Practice in the Marketplace o las Mejores Prácticas de un Determinado Mercado, por ejemplo, modelos de procesos de negocio, modelado orientado a objetos, modelos de navegación web.

Best Practice:

Es un método o técnica que demostró consistentemente mejores resultados a los ya obtenidos de otra forma, y que se usa como un benchmark, técnica que sirve para medir el rendimiento de un sistema o componente del mismo,

Timeboxing:

Es el empleo del timebox.

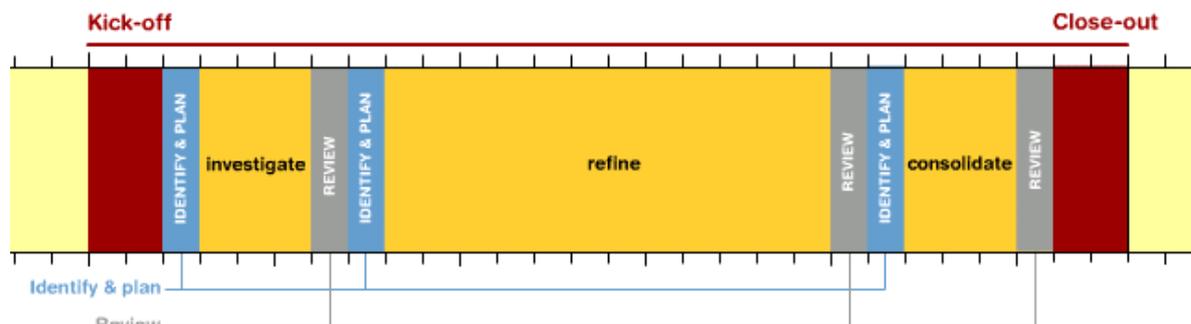


Fig. 8: Timebox

Fuente: *DSDM Tour* <www.dsdm.org>

Timebox:

Permite el manejo de la flexibilidad de los requisitos en DSDM. En todo proyecto, se fija una fecha de terminación del mismo, en forma inamovible. Se determina así un timebox total al que se le asigna un conjunto de requisitos prioritarios, algunos obligatorios y otros no mandatorios, lo que proporciona un margen de maniobra cuando el proceso no se ajusta al plan o cuando los requisitos son nuevos. DSDM subdivide el timebox en otros cortos timeboxes de anidación, cada uno de una duración de dos a seis semanas dentro del plazo total del proyecto.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Cada timebox consta de tres fases o etapas:

- Investigación: plazo breve en el que se ve si el equipo está tomando la dirección correcta.
- Refinamiento: en la que se construye en función de los comentarios resultantes de la investigación.
- Consolidación: es la parte final del timebox, tiene por objeto que nada quede descuidado.

La priorización de las necesidades en todo el timebox se comprueban y posiblemente se reasignan mediante las MoSCoW Rules o Reglas de MoSCoW.

MoSCoW Rules o Reglas de MoSCoW:

Constituyen la base sobre la cual se toman las decisiones sobre la totalidad del proyecto, y durante cualquier timebox. En la palabra Moscow las letras significativas son M, S, C y W, las “o” son para la “diversión”, podrían representar “on time” (“a tiempo”) y “on Budget” (“en presupuesto”).



Fig. 9: Reglas MoSCoW

Fuente: *DSDM Tour* <www.dsdm.org>

Los timeboxes son fijos, por lo que los entregables del timebox pueden variar de acuerdo a la cantidad de tiempo restante.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| MoSCoW Rules o Reglas de MoSCoW | |
|--|---|
| Must Haves Fundamental to the projects success | Debe tener estos requisitos Fundamental para el éxito del proyecto |
| O | O |
| Should Haves Important but the projects success does not rely on these | Debería tener estos requisitos Importante pero el éxito del proyecto no depende de esto |
| Could Haves can easily be left out without impacting on the project | Podría tener estos requisitos Se pueden postergar fácilmente sin impactar en el proyecto |
| O | O |
| Won't Have this time round can be left out this time and done at a later date. | No va a tener estos requisitos Esta vez se pueden omitir y considerarlos en una fecha posterior |

Tabla 9: Significado de las Reglas MoSCoW

Fuente: *DSDM Tour* <www.dsdm.org>

Se priorizan ciertos requisitos para asegurar que el trabajo esencial sea completado en el plazo acordado.

Modelado:

DSDM entiende mejor el dominio del negocio modelando. Sin embargo, para modelar con el uso de DSDM es necesario hacer ciertas consideraciones en las técnicas:

- Es un desarrollo ágil, luego no tiene que imponer una innecesaria burocracia
- Debe ayudar a la comunicación propiciando que los desarrolladores realicen las preguntas correctas y que los usuarios obtengan una forma de comprobar que les asegure que la solución que se está desarrollando es la requerida.
- Tiene que achicar la brecha semántica entre usuario final y desarrollador: el usuario final se preocupa por las tareas y los escenarios del negocio, mientras que los desarrolladores ven al sistema desde una mirada técnica, de forma tal de que no haya malos entendidos y se cumplan las expectativas de las empresas. Para lograr esto DSDM centra su foco en el usuario y fija su visión desde la perspectiva del usuario/negocio.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Prototipado:

La estrategia y los requisitos de alto nivel son definidos por DSDM en los Facilitated Workshops. Sin embargo los prototipos son necesarios para que los usuarios se aseguren sobre la corrección del detalle de los requisitos.

Que se demuestre un prototipo aumenta en los usuarios el conocimiento de las posibilidades y colabora con ellos para dar un feedback a los desarrolladores. Así se acelera el desarrollo y crece la confianza en el resultado.

Un prototipo puede no ser completo ni probado respecto a todos sus requisitos funcionales y no funcionales afectados, pues los prototipos DSDM son incrementales, evolucionan.

DSDM considera cuatro categorías de prototipo:

1. Negocios: para demostrar los procesos que están siendo automatizados en el negocio,
2. Usabilidad: para analizar los aspectos de la interfaz de usuario que no afecten la funcionalidad,
3. Rendimiento y Capacidad para asegurar que el sistema podrá manejar exitosamente todas las cargas de trabajo,
4. Capacidad / Técnica con el fin de probar un particular enfoque de diseño o comprobar un concepto

Testeo

DSDM testea el desarrollo durante todo el ciclo de vida.

Cuando el tiempo es imperativo en muchos casos se utiliza DSDM, por eso las pruebas deben basarse en el proyecto y ser reconocidas por todos los involucrados.

Es importante mencionar que, proporcionalmente, es mayor que en los proyectos tradicionales la cantidad de pruebas realizadas por usuarios no técnicos y que está probado que, si el testeo no es sistemático, una gran extensión de software continúa sin

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

ser testado, no pudiéndose planificar ni calibrar la amplitud de pruebas o detectar áreas de incertidumbre. Aún pueden usarse los métodos y herramientas de pruebas tradicionales, pero deben adaptarse a fin de que el usuario pueda consustanciarse fácilmente con ellos y ser posibles en los plazos de tiempo y dentro del budget.

Software Configuration o Configuración del Software:

Es el conjunto de todos los Elementos de Configuración del Software (ECS) que constituyen el software, esto es la información y los productos usados y/o producidos en un proceso software. Cada ECS es la información creada como parte del proceso software.

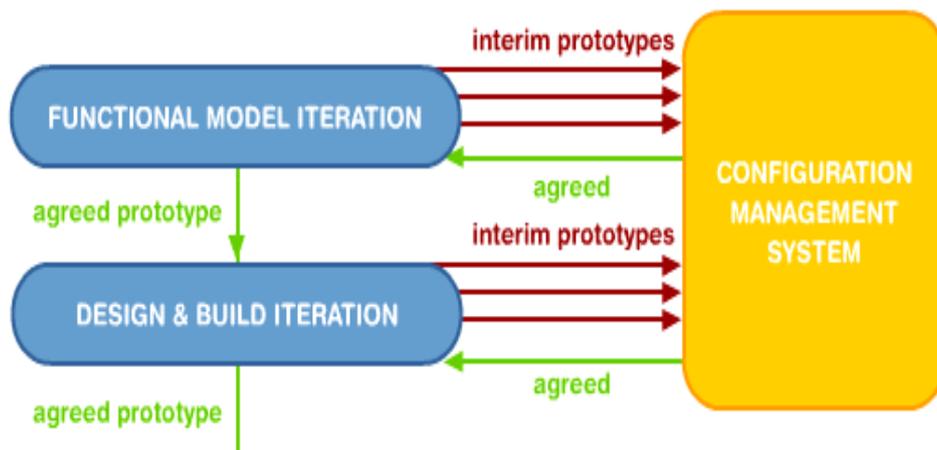


Fig. 10: Sistema de Gestión de la Configuración del Software
Fuente: *DSDM Tour* <www.dsdm.org>

La única representación tangible de un sistema software es la configuración de software por eso es necesario controlarla para conservar su exactitud, para actualizar la información y lograr que sea clara, concisa y segura, al desarrollarse un determinado proceso, más aún en un desarrollo de software iterativo donde el cambio es ineludible:

- Los usuarios necesitan modificar los requerimientos.
- El equipo de desarrollo necesita/quiere modificar el enfoque técnico.
- Los gestores desean modificar el enfoque del proyecto.
- Configuration Managment (CM) o Gestión de la Configuración del Software (GCS):

Es un conjunto de actividades para gestionar los cambios a lo largo del ciclo de vida. Resulta imprescindible el control estricto de estos productos especialmente por la

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

velocidad en que se los va entregando. La gestión garantiza la calidad del software y se desarrolla en todas las fases del proceso software.

En un proyecto de DSDM los siguientes puntos, algunos de los cuales son corolario de los principios de DSDM, son fundamentales para la gestión de la configuración:

- Los miembros del equipo de DSDM deben poder tomar decisiones sin gastos administrativos innecesarios
- Los métodos y herramientas de Configuration Management no deben trabar el desarrollo
- Todos los cambios que se produzcan en el desarrollo de un incremento son reversibles y toda liberación / prototipo se puede reproducir, al igual que los scripts de prueba y los resultados esperados, etc.

Fases del Delivery Process o Proceso de Entrega de un proyecto software para DSDM

1. Factibilidad,
2. Estudio del Negocio,
3. Modelo Funcional de la Iteración,
4. Diseño e Iteración de la Construcción,
5. Implementación en el ambiente de trabajo.

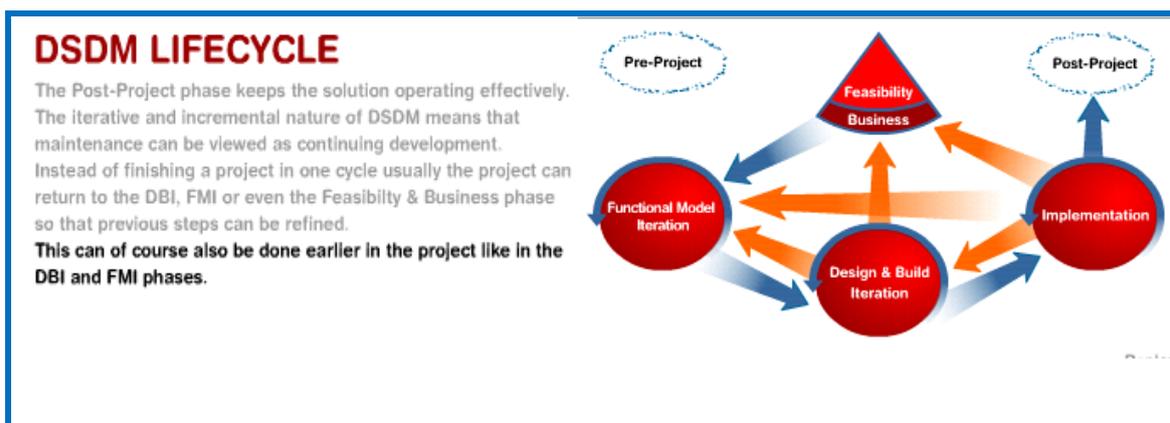


Fig. 11: Fases del proceso de un proyecto software
Fuente: *DSDM Tour* <www.dsdm.org>

Las 5 fases son precedidas por el Pre-Project o Preproyecto y finalizan con el Post-Project o Post- Proyecto. Luego son siete las fases del proceso de un proyecto DSDM.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

El Preproyecto junto con las fases 1 y 2 se realizan en forma secuencial, asegurando las reglas básicas para el resto del desarrollo.

Cuando el proyecto ha entregado la solución al problema de negocio, el equipo del proyecto se disuelve. Es el momento en que se inicia la fase Post-Proyecto, compuesta por actividades que guardan la efectiva operatividad de la solución y controlan que se hayan satisfecho los beneficios de negocio esperados.

¿Qué es DSDM Atern?

Como ya se dijo en el CAP. II, DSDM Atern es una versión de DSDM que surgió en el 2007.

Características básicas de DSDM Atern

- El equipo del proyecto y los stakeholders se mantienen centrados en los resultados de negocio
- La entrega es dentro del tiempo acordado, se garantiza un pronto ROI.
- Todas las personas involucradas en el proyecto trabajan en forma conjunta para ofrecer una solución óptima
- El trabajo se prioriza considerando la necesidad comercial y la capacidad de los usuarios para acomodar los cambios en el plazo acordado
- Atern no se compromete en la calidad, es decir, la solución no es excesiva ni insuficientemente ingenieril
- Se aprovecha el conocimiento, la experiencia y la creatividad de los usuarios finales.
- Se utiliza un ciclo de vida iterativo para desarrollar la solución más adecuada para satisfacer los objetivos del proyecto
- Se particiona el proyecto en períodos de tiempo cortos (timeboxes), cada uno con resultados claramente especificados y acordados, el control se ejerce en todo momento por el director de proyecto y por los propios miembros del equipo.

Principios de DSDM Atern

Cabe destacar los 8 principios de DSDM Atern, originados en las Best Practices:

1. Centrarse en las necesidades del negocio:
Entender las prioridades de la empresa con un Business Case sólido.
2. Estar al día:
No se cambian las fechas. Se busca llegar a la entrega en la fecha límite estipulada
3. Colaborar:
El compromiso y la colaboración de los equipos ayuda a la comprensión y la velocidad de desarrollo. Los equipos incluyen a los representantes de las empresas
4. Nunca renunciar a la calidad:
Establecido desde un principio el nivel de calidad, se conserva porque los proyectos se prueban temprana y continuamente, además de ser revisados en forma constante.
5. Construir de forma incremental:
Los incrementos facilitan a la empresa el uso del trabajo aunque no se haya completado el producto final, promueven la confianza de los interesados y el feedback.
6. Desarrollar Iterativamente:
Porque al principio se pueden cometer errores. El uso de timeboxes facilita los cambios pero se debe aseverar constantemente que se está logrando la solución esperada.
7. Comunicar continuamente y en forma clara:
Fomentar la comunicación face to face, realizar workshops y standups diarios, efectuar presentaciones, modelos y prototipos.
8. Demostrar control:
El equipo debe centrarse en el seguimiento y control de los avances para mantenerlos alineados con la Fase de Fundamentos, evaluando permanentemente la viabilidad del proyecto en función de las metas del negocio.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Atern Products o Productos de Atern

Los Atern Products son las entregas y están asociadas con cada fase del ciclo de vida.

No todos los productos son necesarios en cada proyecto, variando también su formalidad, de acuerdo al proyecto en sí y la empresa u organización. Son ineludibles los factores contractuales y standards corporativos, como en los casos de medio ambiente, calidad, seguridad y salud ocupacional.

Algunos productos son especiales de una fase en particular, pero otros se continúan desarrollando a través de las fases siguientes.

Las grandes corporaciones reguladas, como bancos y laboratorios farmacéuticos, necesitan documentación para cumplir con normas estatales.

Scrum no provee una guía sobre documentación por lo que se le adjudican los documentos típicos de DSDM Atern Products.

Hay documentos que son necesarios, otros a veces y otros no tanto., por eso algunos se crean: Una única vez, Por períodos, Para cada realización (ya sea de uso externo o de uso interno), Si es necesario e Innecesarios

A continuación y en cada caso se detallan cuáles son los documentos y en qué momento dentro del ciclo de vida se creó.

Documentos creados “Una única vez”:

- Business Case en detalle
- Testing Strategy o Estrategia de prueba
- Antes y después de la Arquitectura de Alto Nivel

Tanto para un proyecto como para un producto nuevo estos artefactos se producirán una única vez pero a un alto nivel, porque la información que contienen será válida durante todo el ciclo de vida del proyecto, por lo que se leerán y serán referidos por muchas

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

personas. En el tiempo podrían hacerse cambios, obviamente, pero los factores críticos de decisión acompañan una larga vida a muchos lectores.

Un Test Strategy viene bien como para saber que se espera del equipo antes de entrar en los timeboxes. De tal forma de no enterarse al terminarse el time o, lo que sería peor, en el medio del desarrollo del proyecto que son necesarios requisitos de prueba no considerados.

Gráficamente:

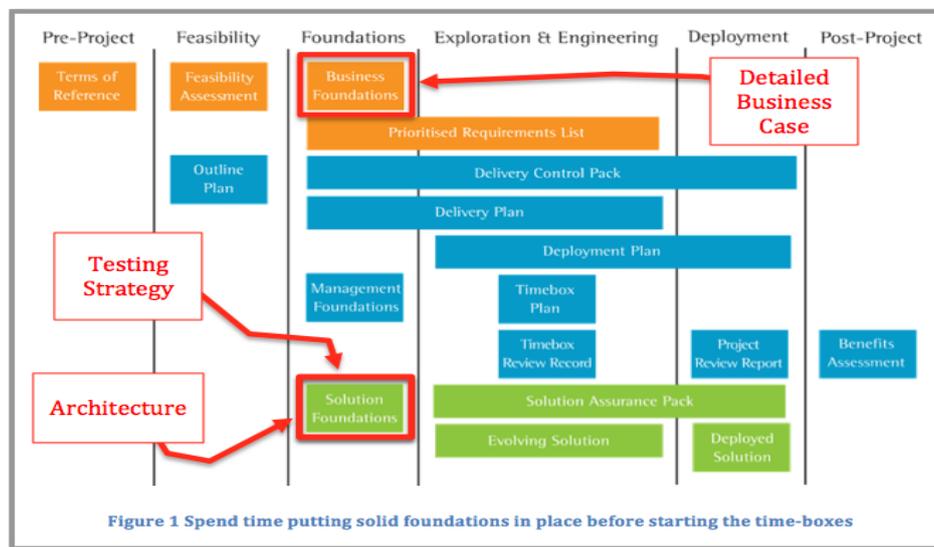


Fig. 12: Documentos creados Una única vez.

Fuente: *Mapping documents to DSDM Atern* <<http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/>>

Documentos creados “Por períodos”:

- Delivery Plan o Plan de Entrega, que incluye el alcance de alto nivel
- Financial Reports o Informes Financieros

Se trata de una documentación necesaria aunque esté dirigida a pocos lectores. Se crea mensualmente y muestra el estado del proyecto. Lo mejor es un conjunto de estadísticas en tiempo real. Y si se exhibe centradamente y se lo actualiza en forma constante, no serán necesarios informes mensuales.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Es importante que haya un acuerdo para informar mensualmente y en forma conjunta con el Sponsor, el Visionary, el Architech y los Team Leaders.

Gráficamente:

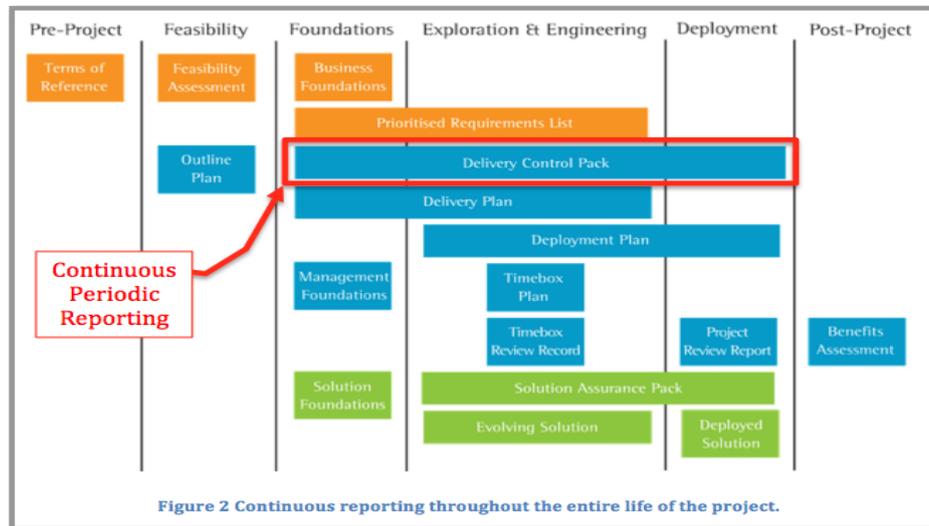


Fig. 13: Documentos creados Por Períodos.

Fuente: *Mapping documents to DSDM Atern* <<http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/>>

Documentos creados “Para cada realización externa”:

- Guía del usuario
- Guías de Instalación y Actualización
- Notas de la versión
- Guía de funcionamiento

Estos documentos deben actualizarse en forma constante. Lo mejor es hacerlo en el mismo timebox en que se produce el producto, pues se deben entregar como parte de lanzamientos Alpha o Beta para pruebas. De hecho, los operadores seguramente querrán hacer pruebas durante la noche anterior a la puesta en funcionamiento y, obviamente, desearán probar scripts de actualización antes del real cut-over del sistema nuevo.

El formato deberá elegirse de acuerdo a que los usuarios sean internos o externos. Los sistemas Wiki resultan efectivos pero los usuarios externos necesitarán acceder si hay relación proveedor/cliente, sobre todo cuando son muchos clientes.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

En lugar de las Guías de usuario es mejor realizar videos, que son más sencillos de producir y permiten comprobar el uso real.

Gráficamente:

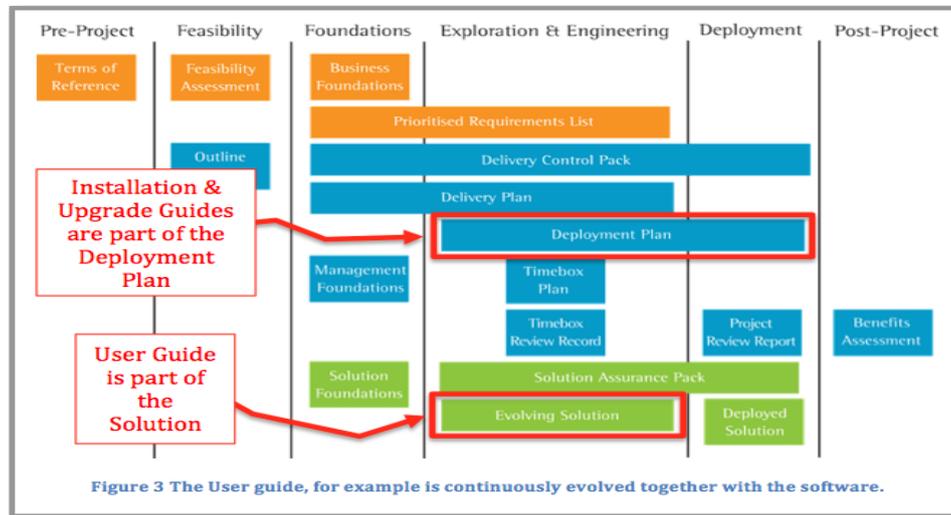


Fig.14 Documentos creados Para cada realización externa.

Fuente: *Mapping documents to DSDM Atern* <<http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/>>

Documentos creados “Para cada realización interna”:

- Datos de Prueba
- Casos de Prueba
- Automatización de Pruebas (Unit & Negocios)
- Resultados de la prueba

Se popularizó el Test Driven Development (Ver sitio: datadrivendevelopment.com/, Data Driven Development: Desarrollo Impulsado por Datos) y también los tests de aceptación Driven Development, técnica desarrollada o redescubierta por Kent Beck.

Test Driven Development es un proceso de desarrollo de software basado en la repetición de un ciclo de desarrollo corto: el desarrollador escribe primero un caso de prueba automatizado que define o bien una mejora deseada o una nueva función, luego produce la mínima cantidad de código para superar esa prueba, y para terminar refactoriza el nuevo código de normas aceptables.

Estos artefactos son creados en los mismos timeboxes que el software. Se usan frecuentemente y durante todo el tiempo del proyecto. Con la automatización de

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

pruebas y las prácticas de integración, los resultados pueden registrarse como vídeos para ser utilizados además para formación interna y como muestra de un producto de trabajo.

“Los datos correctos en el lugar correcto en el momento adecuado”. Frase de Data Driven Development

Es común que los desarrolladores rechacen pensar en objetos de prueba como integrante de su responsabilidad o peor aún, como una parte del software. Estos artefactos deben considerarse “importantes”, tal vez más que el código. Se considera que si el Equipo acepta las pruebas esto indica que es maduro.

Gráficamente:

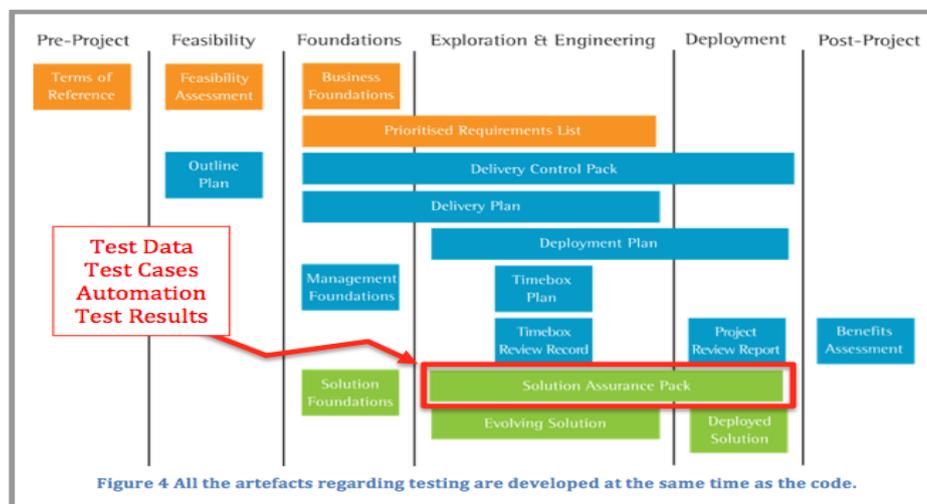


Fig. 15: Documentos creados Para cada realización interna.

Fuente: *Mapping documents to DSDM Atern* <<http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/>>

Documentos creados “Si es necesario”:

- Pantalla y el informe de diseño Diseños

Se puede invertir mucho tiempo en el detalle de cada pantalla de un sistema y el diseño de cada informe. Cuando el código está escrito toda la documentación será obsoleta.

Pero, estos artefactos ayudan a la comprensión.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Gráficamente:

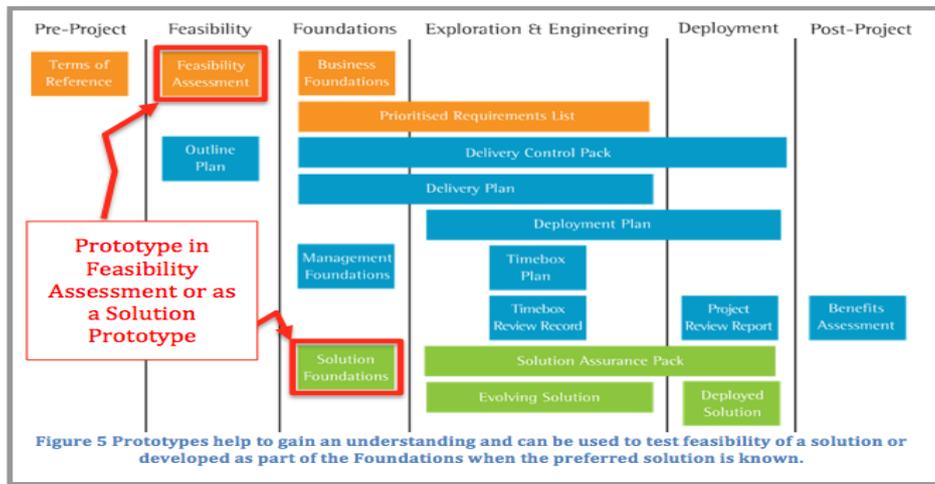


Fig.16: Documentos creados Si es necesario.

Fuente: *Mapping documents to DSDM Atern* <<http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/>>

Documentos “Innecesarios”:

- Requisitos detallados
- Especificaciones detalladas
- Diseños detallados

Prescriben algo que aún no existe, luego son pasibles de cambios. Deben evitarse.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

CAPITULO IV – PROPUESTA DE ANALISIS COMPARATIVO

En este capítulo se profundizará en los aspectos de Scrum y DSDM, indicando, en algunos casos, los beneficios y/o restricciones y/o recomendaciones respecto de los mismos, para luego presentar el DSDM Roadmap, definido por Chris Fortuin, mencionado en el CAP. II.

Se plasmarán las características esenciales de cada una de estas metodologías. El análisis de estas, permitirá extraer un resumen de las mismas y presentar un Cuadro Comparativo, que revele las diferencias que mantienen.

Finalmente, a partir de las hipótesis planteadas en el CAP.I, se podrá plantear una Propuesta de Análisis Comparativo que colabore, ante un determinado proyecto de software, para una organización específica, en la toma de la decisión de emplear una u otra metodología o ambas.

Scrum - Aspectos Fundamentales

Personas - Roles de Scrum y skills necesarios

Scrum Master o Facilitador:

Como se explicó en el punto sobre Desarrollo Iterativo e Incremental y en Roles de Scrum, CAP. III, el Facilitador vela por que se cumpla el proceso de Scrum, colaborando en forma directa con el equipo para que alcance su objetivo final, liberándolo de eventuales impedimentos y protegiéndolo de interrupciones externas.

El gestor de proyecto pasa a ser un Facilitador, elemento clave del equipo.

Tiene que conseguir el mejor resultado para el proyecto del cliente, que el equipo sea hiperproductivo y simultáneamente disfrute de su trabajo. Tiene que ser líder del equipo, esto es una forma de pensar y actuar, es tener un estilo de vida que consigue:

- Que las cosas se hagan, en tiempo y forma
- Que se cambien, cuando sea necesario

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Que se mejoren, en forma continua
- Que se innove.

Existen dos modelos de gestión de equipo:

- Un modelo de gestión tradicional, basado en la autoridad, dirección y control por parte de un superior que planifica y dicta las tareas a hacer y las asigna. Con las siguientes consecuencias:
 - No hay sentimiento de responsabilidad compartido en el equipo
 - Se producen esperas y pasos hacia atrás
 - Son necesarias correcciones por incumplimiento
 - Fomenta la competitividad en detrimento de la sintonía y colaboración
- El modelo que favorece el trabajo en equipo, en proyectos emergentes, es decir, proyectos complejos y de alto riesgo donde el alcance se va precisando durante la ejecución del mismo y/o que necesitan creatividad por parte de sus integrantes, los que colaboran codo a codo y comparten la responsabilidad del resultado del proyecto.

Este segundo modelo requiere un gestor/líder que se comporte como un Facilitador, un líder al servicio del equipo, lo que en Scrum se denomina Scrum Master.

El Scrum Master es una persona que debe:

- Entender el negocio del cliente, la metodología de trabajo, la tecnología
- Tener dotes de trato interpersonal, con el cliente y con el equipo, y ser capaz de:
- Observar, escuchar, preguntar mucho y parafrasear para comprender las necesidades, motivaciones y sentimientos de los demás, poniéndose en lugar del otro antes de dar una opinión propia (siempre que sea necesario que la dé) evitando se prejuzgue al otro y buscando tener empatía.
- Negociar, comunicar adecuadamente la información correcta en el momento preciso, adaptándola a las necesidades de la audiencia.
- Enfocar al equipo, orientarlo para avanzar y cumplir con las expectativas del cliente, a la vez que cuidar la calidad del producto, sin explicitar (dictar) cómo hacerlo.
- Motivar al equipo.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Esta manera alternativa de conceptualizar el trabajo en equipo se utiliza desde hace varios años en miles de proyectos. Actualmente hay más de 50.000 Scrum Masters certificados a nivel mundial, a los que hay que sumar los que no están certificados, que trabajan en empresas donde la competitividad de mercado, la innovación, la calidad y la productividad son fundamentales, desde startups a grandes empresas como Nokia, Toyota, Google, etc., donde una startup es una organización humana con gran capacidad de cambio, que desarrolla productos o servicios, de gran innovación, altamente deseados o requeridos por el mercado, donde su diseño y comercialización están orientados completamente al cliente. Que opera con costos mínimos, pero obtiene ganancias que crecen exponencialmente, que mantiene una comunicación continua y abierta con el cliente y se orienta a la masificación de las ventas, aprovechando la comunicación que brinda Internet y sus plataformas.

Cualidades de un buen gestor de equipo:

- Confía en el equipo y lo potencia. No impone su autoridad. No exige al equipo que lo siga
- No planifica las tareas del equipo ni determina cuál tarea debe realizar cada integrante ni cómo debe hacerlo
- No dirige los resultados del proyecto con sus decisiones (esto es responsabilidad del cliente).
- No hace micro gestión. Esto es el hábito de algunos jefes de interferir innecesariamente en las actividades de sus trabajadores.
- Asume que el equipo es el experto, no lo cuestiona, no lo desautoriza
- Ayuda al equipo a autoorganizarse para alcanzar los objetivos del proyecto, a comunicarse, a compartir ideas, a colaborar y a optimizar su forma de trabajo para una mayor productividad, siendo más flexible y adaptativo a cambios en las necesidades del cliente.
- Fomenta la colaboración entre los miembros del equipo en los procesos de decisión y con el cliente, en las reuniones de planificación de la iteración o Sprint, en las demostraciones de los resultados de cada iteración y en las retrospectivas.
- No da respuestas, guía al equipo con preguntas para que este descubra por sí mismo una solución y aprenda. Motivando de esta forma la creatividad del mismo, facilitando así que el equipo tome decisiones consensuadas (sostenibles en el

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

tiempo), que encuentre las mejores soluciones posibles mediante la sinergia de enfoques, conocimientos y experiencias, a partir de los cuales pueda incluso combinar diferentes opciones.

- Quita los impedimentos que el equipo no puede resolver por sí mismo, logrando así que su equipo avance, no se bloquee, se mantenga focalizado en su trabajo, elimine ineficiencias y maximice su productividad. Hasta es capaz de intuirlos y por eso anticipar la mitigación de riesgos. Si es necesario busca recursos para que el equipo pueda autovalerse en este sentido (formación, soporte experto, etc.).
- Protege al equipo de interrupciones externas de forma tal que pueda cumplir con el compromiso de objetivos que adquirió al inicio de la iteración.
- Aporta seguridad y confianza para promover que las conversaciones sean sinceras, se comparta información y la comunicación sea frecuente entre los miembros del equipo.
- Ayuda a escuchar, a ponerse en lugar del otro, a entender sus razones, a consensuar el problema, a que se propongan soluciones alternativas, con la conformidad del resto, a evaluarlas y a llegar a conclusiones y acuerdos, para mantener un ambiente constructivo, sin actitudes recriminatorias, atacantes y/o acusadoras desde adentro y afuera del equipo.
- Logra crear un sentido de identidad del equipo, de comunidad con un proyecto. Fomenta que unos aprendan de otros y establezcan vínculos, desechando la competitividad individualista.
- Promueve la confianza entre el cliente y el equipo
 - Facilitando las reuniones de planificación del Sprint y las demostraciones, se asegura de que tanto el cliente como el equipo comparten el mismo objetivo del proyecto, con una visión conjunta de la lista del Product Backlog y colaboran para alcanzar el máximo valor del proyecto.
 - En las reuniones Planning Sprint y en las demostraciones reconoce la bondad del trabajo realizado por el equipo y lo presenta como verdadero protagonista delante del cliente.
- Tolera errores y no busca culpables, por el contrario, mejorar el proceso de trabajo
 - No busca culpables de los errores que se hayan producido. No acusa ni recrimina porque no quiere que el equipo se desmotive e incluso se paralice.
 - Asume que es el contexto quien provoca los errores. Con intensión constructiva, se centra en ayudar al equipo a descubrir, a través de un análisis causal, qué debe

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

ser mejorado del proceso de trabajo, de interrelaciones y de colaboración, en intraequipo como con terceros, para que el error no se repita.

- Acepta imperfecciones (desde su óptica) en el trabajo de los integrantes del equipo y procura que el equipo encuentre la manera de ir mejorando. Hasta llega a dejar que el equipo cometa errores para que aprenda a partir de la reflexión sobre la equivocación.
- Cuando se produce un problema, lo primero que debe pensar el gestor de un equipo ágil es que la responsabilidad es suya, no de otros. Se debe preguntar qué debería haber hecho o estar haciendo para que no ocurriera el problema y/o cómo podría haberse anticipado.
- Es humilde, autocrítico, acepta que siempre hay algo por aprender y se esfuerza en ello.
- Acepta las observaciones de los otros y asume sus errores como oportunidades de mejora.
- Da siempre el ejemplo especialmente en la manera de interrelacionarse con los demás.
- No habla de "yo", sino de "nosotros".
- No eleva el tono de voz ni grita. Habla calmado, sin excitarse, tiene buen humor y sonríe.

Product Owner (ya interno o externo de la organización):

Responsabilidades del Product Owner:

- Define los objetivos del producto o proyecto.
- Dirige los resultados del proyecto y maximiza su ROI. Es el propietario de la planificación del proyecto: crea y mantiene la lista priorizada con los requisitos necesarios para cubrir los objetivos del producto o proyecto, conoce el valor que aportará cada requisito y calcula el ROI a partir del coste de cada requisito que le proporciona el equipo.
- Antes de iniciar la primera iteración, debe tener definida la meta del producto o proyecto y el Product Backlog creado, la que no tiene que estar completa, aunque sí es necesario que los requisitos más prioritarios estén identificados y con suficiente detalle para que el equipo empiece a trabajar.
- Reparte los objetivos/requisitos en iteraciones y establece un calendario de entregas.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- En las siguientes iteraciones replanifica el proyecto en función de los requisitos que aportan más valor en ese momento (demandas del mercado, movimientos de la competencia, etc.), de los requisitos completados en la iteración anterior y del contexto del proyecto. Puede llegarse a un punto del proyecto en que no valga la pena analizar ni desarrollar los requisitos restantes, en función del poco ROI que tengan.
- Colabora con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración:
 - Participa en el Sprint Planning, proponiendo los requisitos más prioritarios, respondiendo a las dudas del equipo y detallando los requisitos que el equipo se compromete a hacer.
 - Responde durante toda la iteración a las preguntas que puedan surgir.
 - No cambia los requisitos que se están desarrollando en una iteración, una vez iniciada.
 - Participa en la reunión de Demostración de la iteración, revisando los requisitos completados.

El Team o Equipo de desarrollo del producto del proyecto:

Autoorganizado y autogestionado, realiza de manera conjunta las siguientes actividades:

- Selecciona los requisitos que se compromete a completar en una iteración.
- Estima la complejidad de cada requisito en el Product Backlog
- En el Sprint Planning decide cómo va a realizar su trabajo:
 - Selecciona los requisitos que puede completar en cada iteración, realizando al cliente las preguntas necesarias.
 - Identifica todas las tareas necesarias para completar cada requisito.
 - Estima el esfuerzo necesario para realizar cada tarea.
 - Cada miembro se autoasigna ciertas tareas.
- Durante la iteración, trabaja de manera conjunta para conseguir los objetivos de la iteración. Cada especialista lidera el trabajo en su área y el resto colabora, si es necesario, para poder completar un requisito.
- Al finalizar cada Sprint, participa en la Sprint Review, realizando una Demostración y una Sprint Retrospective.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Tiene que depender en una mínima expresión de personas externas al equipo, de manera que el compromiso que adquiere en cada iteración no se ponga en peligro.

Se crea una sinergia que permite que el resultado sea más rico al nutrirse de las diferentes experiencias, conocimientos y habilidades de todos. Colaboración creativa.

La dedicación de cada miembro del equipo es a tiempo completo y trabajan todos en la misma localización física, para maximizar la comunicación mediante conversaciones cara a cara, diagramas en pizarras blancas, etc. Mientras dure el proyecto el equipo debe ser estable, es decir que no deben cambiarse sus integrantes, por el capital intelectual adquirido, por las relaciones interpersonales logradas, etc.

Potenciación del equipo:

En Scrum los equipos son autogestionados. Esto potencia al equipo en su trabajo en equipo, lo motiva y consigue que sus miembros se comprometan al utilizar las técnicas aplicables en las actividades de Scrum:

En la reunión de planificación de la iteración:

- por la responsabilidad que afronta al tener que decidir qué requisitos, objetivos o tareas puede desarrollar en cada iteración
- por el compromiso que asume de manera conjunta en esa decisión
- por la autoridad para gestionar que se otorga al equipo
- por establecer criterios comunes. El equipo realiza conjuntamente:
 - la identificación de las tareas necesarias a realizar en la iteración.
 - la estimación del esfuerzo de las tareas.
 - la autoasignación de tareas entre los miembros del equipo.

En el Sprint Review:

- por la satisfacción de mostrar los resultados que va obteniendo, sin necesidad de esperar meses para poder exhibir su obra

En el Sprint Retrospective:

- porque se respeta su capacidad de analizar las mejoras a realizar en su modo de trabajar, preguntándose:

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Qué cosas han funcionado bien
- Cuáles hay que mejorar.
- Qué cosas quiere probar hacer en la siguiente iteración.
- Qué ha aprendido.
- Cuáles son los problemas que podrían impedirle progresar adecuadamente.

Planificación del producto o proyecto

Timeboxing:

En todas las actividades de Scrum se determina un tiempo de realización, como se indicó en Desarrollo iterativo e incremental, CAP III. Las actividades son timeboxed, esto es la determinación a priori de un tiempo máximo para conseguir ciertos objetivos determinados. Esto favorece la priorización de objetivos/tareas y se fuerza la toma de decisiones. Así se fomenta la productividad y el aprendizaje del tiempo necesario para realizar una tarea.

Product Backlog o Lista de objetivos / requisitos priorizada:

Cuando se trata del desarrollo de un producto, la lista evoluciona en toda la vida del producto. En caso de un proyecto, a medida que éste avance van apareciendo los requisitos menos prioritarios faltantes.

- La lista consta de los objetivos/requisitos de mayor nivel del producto o proyecto, que se expresan generalmente en forma de Historias de Usuario, esto es un requerimiento de negocios visto desde el punto de vista de un usuario que se escriben con el siguiente formato: "Como xxx, quiero hacer yyy con el objetivo de zzz", donde, xxx es el tipo de Usuario (quién), yyy es lo que el sistema debe permitir realizar (el qué) y zzz es el beneficio o valor buscado (el por qué).

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM



Fig. 17: Tarjeta Historia de Usuario

Fuente: *Objetivos como historias de usuario*

<<http://www.proyectosagiles.org/introduccion-estimacion-planificacion-agil#historias-usuario>>

- A cada objetivo se le adjudica el Valor que aporta al cliente y el Coste Estimado de alcanzarlo. La lista se prioriza cotejando el valor que cada requisito aporta al negocio contra el coste estimado que implica su desarrollo, esto es basándose en el ROI.
- Se indican las iteraciones posibles y los releases o entregas que espera el cliente
- Se consideran también los Riesgos del proyecto, incluyéndose los requisitos o tareas necesarios para reducir su impacto.

| Área de requisitos | Requisitos | Origen | Valor | Estimación inicial | Factor Ajuste | Estimación ajustada | Iteración: | | | | | |
|--------------------|------------------------|------------|-------------|--------------------|---------------|---------------------|------------|------------|------------|-----------|----------|----------|
| | | | | | | | 1 | 2 | 3 | 4 | 5 | |
| Área X | Requisito A | Marketing | 2000 | 15 | | 15 | | | | | | |
| Área Z | Requisito B | Producción | 1750 | 20 | | 20 | | | | | | |
| Área Y | Requisito C | Ventas | 1500 | 20 | | 20 | | | | | | |
| | Iteración 1 | | 5250 | 55 | | 55 | 0 | 0 | 0 | 0 | 0 | 0 |
| Área Z | Requisito C | Producción | 1250 | 15 | 0,2 | 18 | 18 | 18 | 0 | | | |
| Área X | Requisito D | Producción | 1250 | 20 | | 20 | 20 | 20 | 0 | | | |
| Área Z | Requisito E | Marketing | 1000 | 15 | 0,2 | 18 | 18 | 18 | 0 | | | |
| | Iteración 2 | | 3500 | 50 | | 56 | 56 | 56 | 0 | 0 | 0 | 0 |
| | Primera entrega | | 8750 | 105 | | 111 | 111 | 56 | 0 | 0 | 0 | 0 |
| Área X | Requisito F | Marketing | 1250 | 20 | 0,2 | 24 | 24 | 24 | 24 | 0 | | |
| Área Y | Requisito G | Marketing | 750 | 15 | | 15 | 15 | 15 | 15 | 0 | | |
| Área Y | Requisito H | Ventas | 750 | 15 | 0,2 | 18 | 18 | 18 | 18 | 0 | | |
| | Iteración 3 | | 2750 | 50 | | 57 | 57 | 57 | 57 | 0 | 0 | 0 |
| Área Z | Requisito I | Producción | 700 | 15 | 0,2 | 18 | 18 | 18 | 18 | 18 | | |
| Área Y | Requisito J | Marketing | 500 | 10 | 0,5 | 15 | 15 | 15 | 15 | 15 | | |
| Área Y | Requisito K | Ventas | 500 | 20 | 0,2 | 24 | 24 | 24 | 24 | 24 | | |
| | Iteración 4 | | 1700 | 45 | | 57 | 57 | 57 | 57 | 57 | 0 | 0 |
| | Segunda entrega | | 4450 | 95 | | 114 | 114 | 114 | 114 | 57 | 0 | 0 |

Fig. 18: Product Backlog

Fuente: *Lista de objetivos / requisitos priorizada* (Product Backlog)

<<http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog>>

- Antes de comenzar la iteración inicial, el cliente debe tener definido el objetivo final del producto o proyecto y la lista de requisitos determinada, aunque no esté

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

completa ni se hayan detallados todos los requisitos a un mismo nivel. Sí es necesario que los requisitos más prioritarios estén identificados con el detalle suficiente para que se inicie el trabajo.

- En las siguientes iteraciones los requisitos pueden ser más amplios y generales.
- Se evita analizar en detalle requisitos no prioritarios, porque podrían ir variando
- Puede llegarse a un punto del proyecto en que no valga la pena analizar ni desarrollar los requisitos restantes, en función del poco ROI que tengan.

Uso del Product Backlog o Lista de objetivos / requisitos priorizada:

- Se debe poder medir la velocidad de desarrollo, ver una evolución constante y extrapolar fechas de entrega, para lo cual los requisitos deben significar un esfuerzo semejante para ser completados y poder completarse antes de los 10 días, siempre que las iteraciones sean de 20 días laborables. Tanto el progreso del proyecto como la velocidad de completitud de requisitos se exhiben en un Burndown Chart o Gráfico de Trabajo Pendiente (ver más abajo).
- A cada requisito se le asocia factor de complejidad, para poder ajustar su coste
- Cuando un requisito depende de otro, se lo coloca debajo del que depende.
- Cuando un requisito no se completa en una iteración, se lo vuelve a poner en alguna próxima iteración, señalando el coste pendiente de desarrollo.
- El "origen" indicado en la lista, sirve para saber quién participaría en la definición de un objetivo/requisito, siendo conveniente su presencia al momento de su demostración.

Burndown Chart o Gráfico de Trabajo Pendiente:

Se pueden utilizar dos Burndown Chart:

- Product Burndown Chart: que indica los días pendientes para completar los requisitos del producto o proyecto, que se hace a partir del Product Backlog.
- Sprint Burndown Chart: que muestra las horas pendientes para completar las tareas de la iteración, confeccionado con la Iteration Backlog

Con este tipo de gráfico se pueden hacer simulaciones: añadir requisitos para ver cómo se postergan fechas de entrega, comprobar avances quitando requisitos o añadiendo otro equipo, etc.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

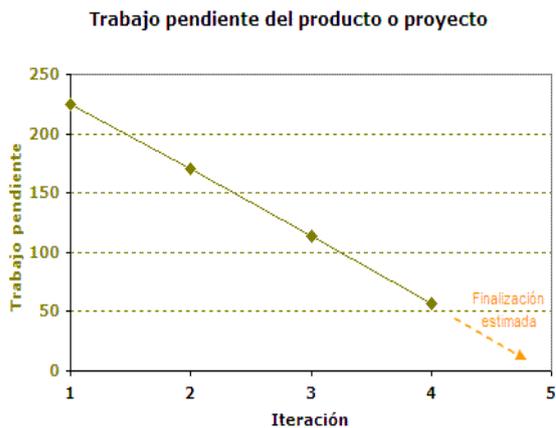


Fig. 19: Product Burndown Chart

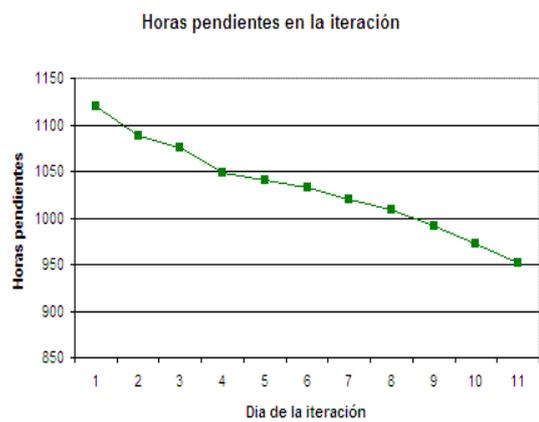


Fig.20: Sprint Burndown Chart

Fuente: *Gráficos de trabajo pendiente* (Burndown charts) <<http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts>>

Planificación de la iteración

Sprint Backlog o Lista de tareas de la Iteración:

Como se vió en el CAP. III, es elaborada por el equipo en el Sprint Planning. Permite ver en cuales tareas tiene problemas el equipo y no avanza, y así tomar decisiones en consecuencia.

Para cada objetivo/requisito muestra: las tareas, el esfuerzo pendiente para completarlas y la autoasignación de las mismas que hicieron los miembros del equipo.

| Requisito | Tarea | Quien | Estado (No iniciada / en progreso / completada) | Dia: | | | | | | | | | | | |
|-------------|----------|-------|--|---------------------|------|------|------|------|------|------|------|------|-----|-----|----|
| | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| | | | | Horas pendientes | 1120 | 1088 | 1076 | 1048 | 1040 | 1032 | 1020 | 1008 | 992 | 972 | |
| Requisito A | Tarea 1 | Joao | Completada | 16 | 8 | | | | | | | | | | |
| Requisito A | Tarea 4 | Laura | Completada | 4 | | | | | | | | | | | |
| Requisito A | Tarea 5 | Laura | Completada | 4 | | | | | | | | | | | |
| Requisito A | Tarea 3 | Gabri | Completada | 8 | | | | | | | | | | | |
| Requisito A | Tarea 2 | Laura | Completada | 16 | 8 | 4 | | | | | | | | | |
| Requisito A | Tarea 6 | Gabri | Completada | 8 | 8 | 8 | | | | | | | | | |
| Requisito A | Tarea 7 | Joao | Completada | 16 | 16 | 16 | 8 | | | | | | | | |
| Requisito A | Tarea 8 | Laura | Completada | 8 | 8 | 8 | | | | | | | | | |
| Requisito A | Tarea 9 | Laura | Completada | 8 | 8 | 8 | 8 | 8 | | | | | | | |
| Requisito A | Tarea 10 | Laura | Completada | 8 | 8 | 8 | 8 | 8 | 8 | | | | | | |
| Requisito A | Tarea 11 | Joao | Completada | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 8 | | | | |
| Requisito B | Tarea 12 | Gabri | Completada | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 8 | | |
| Requisito B | Tarea 13 | Laura | Completada | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 8 | |
| Requisito B | Tarea 14 | Joao | En progreso | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 4 |
| Requisito B | Tarea 15 | Gabri | En progreso | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Requisito B | Tarea 16 | Laura | En progreso | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Requisito C | Tarea 17 | Joao | No iniciada | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| Requisito C | Tarea 18 | Gabri | No iniciada | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Requisito C | Tarea 19 | Laura | No iniciada | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| Requisito C | Tarea 20 | Joao | No iniciada | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |

Fig.. 21: Planificación de la Iteración.

Fuente: *Planificación de la iteración* (Sprint Planning) <<http://www.proyectosagiles.org/planificacion-iteracion-sprint-planning>>

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Uso del Sprint Backlog:

- Los objetivos/requisitos están ordenados según la priorización que haga el cliente
- Se ordenan las tareas, una debajo de otra en función de la dependencia entre ellas
- Se identifican las tareas por coste, entre 4 y 16 horas.

Sprint Planning o Planificación de la Iteración:

Se planifica en dos partes las tareas a realizar en una iteración determinada:

- Primera parte:
en un timebox de a lo sumo 4 horas, el cliente presenta la lista de necesidades y luego el equipo la analiza, pregunta dudas, agrega condiciones y selecciona los más prioritarios objetivos/requisitos asumiendo el compromiso de completarlos en la iteración.
- Segunda parte:
en un timebox de a lo sumo 4 horas, el equipo planifica la iteración, la táctica necesaria para llevar a cabo las tareas que considera necesarias para completarlas, conformado el Sprint Backlog, estima el esfuerzo necesario para cada tarea y cada integrante del equipo se autoasigna las tareas que puede cumplir.

Scrum Taskboard o Tablero de Tareas:

Permite gestionar el Product Backlog Items comprobando las tareas completadas. Al lado de cada objetivo se anotan las tareas para completarlo, utilizando post-its, que se van moviendo hacia la derecha para indicar el cambio de estado (no iniciadas, en progreso, terminadas). Cada miembro del equipo puede emplear pequeños adhesivos de colores para colocar sobre cada tarea, de forma de denotar en qué tareas trabaja cada uno.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Curso, para evitar el multitasking. Y, a la vez, deberá ser el mínimo posible el número de objetivos abiertos en curso (WIP: Work In Progress) y resueltos de arriba hacia abajo de acuerdo a la prioridad implícita en el Product Backlog.

- En Impedimentos se volcarán los obstáculos que impiden el avance del equipo u otros riesgos que deben tenerse especialmente en cuenta. Se conviene indicar quién es el responsable de su solución, ya sea un miembro del equipo o el Scrum Master o Facilitador.
- En Retrospectiva se van anotando lo que está funcionando bien (+, Pluses) y los problemas (Δ , Deltas)
- En el espacio libre de la derecha, sobre Impedimentos y Retrospectiva, encima, alguna información referida al proyecto y, debajo, alguna referente a la iteración

Beneficios del Sprint Planning:

- Mayor productividad gracias a la comunicación y a la sinergia creada
- Se potencia el compromiso asumido
- Se confía más en una estimación del equipo en su conjunto

Durante la ejecución del Sprint Planning:

- Scrum Daily Meeting o Reunión diaria de sincronización del equipo

Esta reunión facilita al equipo el traspaso de información y la mutua colaboración entre sus integrantes.

Beneficios del Scrum Daily Meeting:

- Como cada miembro del equipo expone ante el resto, tanto sus logros como sus impedimentos, la productividad aumenta y el compromiso del equipo se potencia:
- Los miembros pueden ayudar en sus tareas a aquellos que lo necesitan. El Scrum Master o Facilitador colaborará para solucionar los problemas que el equipo no pueda resolver solo o que les resta tiempo para cumplir con el desarrollo de requisitos asumidos.
- Más problemas pueden identificarse a tiempo

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Se evidenciará la realización de tareas no planificadas, que el resto del equipo no conoce y que pueden no estar en línea con el compromiso del equipo
- Se conocerán las necesidades de cada uno, su ritmo de trabajo, los criterios que utiliza para cumplir con sus tareas y si estos están o no alineados con el plan del resto del equipo.
- Los integrantes del equipo aprenden, de los otros miembros al ver cómo trabajan, conforme a su especialidad y su experiencia y se contactan con el estado de la iteración, al asumir si es posible completar los objetivos a los que se comprometieron, en función de las desviaciones y tareas pendientes.

Restricciones del Scrum Daily Meeting:

No se resuelven los problemas en el Scrum daily meeting, hasta después de la reunión:

- Existen miembros que no se interesan por algún tema determinado.
- Se programan reuniones en el Scrum daily meeting para sincronizar tareas, resolver temas, etc.

El proceso de ejecución de las tareas debe basarse en criterios consensuados del equipo.

Recomendaciones sobre el Scrum Daily Meeting:

- El Scrum daily meeting se debe realizar de pie, para que los miembros no se relajen ni extiendan demasiado en detalles.
- Las reuniones de colaboración deben realizarse exactamente después del Scrum daily meeting

Recomendaciones para la ejecución del Sprint:

Para completar en la iteración la mayor cantidad de requisitos asumidos, se minimiza la cantidad de objetivos/requisitos en los que el equipo trabaja en forma simultánea (WIP: Work In Progress), completando los que dan más valor al cliente primero, lo que aumenta la capacidad de reacción ante cambios o situaciones no previstas.

Restricciones para la ejecución del Sprint:

No pueden cambiarse los objetivos/requisitos de la iteración.

- Un cambio dificultaría la concentración del equipo y bajaría su compromiso

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Así, se obliga al cliente a cumplir con su responsabilidad de saber qué es más prioritario para desarrollar, antes de iniciar el Sprint. Aunque los objetivos/requisitos que se están desarrollando se determinaron como prioritarios antes de iniciar el Sprint y la posibilidad de cambios es mínima porque las iteraciones en Scrum son cortas.

Terminación anormal del Sprint:

Se solicita sólo en casos muy especiales. Por ejemplo si cambió mucho el contexto del proyecto y no se puede llegar al final del Sprint para implementar los cambios o si el equipo descubre que no puede de manera alguna cumplir con lo prometido. En este caso se da por concluida la iteración y se comienza otra mediante un Sprint Planning.

Métricas para Scrum

Métricas ágiles y cuadro de mandos integral para Scrum:

Es necesario considerar un conjunto de métricas para no descuidar aspectos importantes como la calidad, los costes, los riesgos, la sostenibilidad de la velocidad con que se obtienen objetivos, etc., creando así un Agile BSC o Agile Balanced Scorecard, esto es un Cuadro de Mandos Integral, herramienta de gestión de rendimiento estratégico, que permite medir el desempeño. El concepto BSC fue creado por los Dres. Kaplan y Norton en 1990.



Fig. 24: Cuadro de mandos integral para Scrum

Fuente: *Métricas ágiles y cuadro de mandos integral para Scrum*

<<http://www.proyectosagiles.org/metricas-agiles-cuadro-mandos-balanceado-scrum>>

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Selección y uso de las métricas:

Es recomendable que el valor aportado al proyecto se base en un único indicador económico que sea clave con el cual el cliente mida los objetivos del proyecto, para la toma de decisiones y la forma de invertir en el proyecto. Para el resto del cuadro de mandos se debe escoger el mínimo número de métricas que permita tomar decisiones sobre los resultados del proyecto y las necesidades del cliente.

Métricas ágiles del cuadro de mandos:

Las métricas más importantes surgen de las herramientas de Scrum: el Product Backlog, el Sprint Backlog y los Burdown Charts, por lo que el momento adecuado para actualizar el cuadro de mandos es luego de la Sprint Retrospective.

Las métricas a considerar son las de:

- productividad y efectividad de la entrega
- resultados del proyecto
- situación financiera
- calidad
- riesgos, impedimentos, proceso y mejora continua

Sprint Review o Revisión de la Iteración - Inspección y adaptación

Demostración de los requisitos completados

Como se explicó en el CAP. III, es una reunión informal en la que el equipo presenta al cliente los requisitos que fueron completados en el Sprint, como incremento de producto. De acuerdo a los resultados mostrados y de los cambios que se hayan hecho en el tenor del proyecto, el cliente hace las adaptaciones que considere necesarias en forma objetiva, desde la primera iteración, para el Product Backlog Refinement o Replanificación del proyecto. Se realiza en un timebox de no más de 4 horas.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Beneficios del Sprint Review:

- El cliente puede ver, en forma objetiva, cómo se desarrollaron los requisitos que oportunamente proporcionó, si se han cumplido sus expectativas, entender mejor qué es lo que realmente necesita y asumir nuevas determinaciones en relación al proyecto.
- El equipo puede ver claramente si efectivamente entendió cuáles eran los requisitos que solicitó el cliente y en qué aspectos hay que mejorar la comunicación con el cliente.
- El equipo siente mayor satisfacción cuando puede mostrar resultados alcanzados sin necesidad de esperar el transcurso de meses de trabajo para mostrar sus logros

Restricciones del Sprint Review:

Se pueden mostrar únicamente los requisitos completados, para evitar falsas expectativas al cliente y que pueda asumir decisiones correctas y objetivas en relación a la velocidad de desarrollo y respecto del resultado efectivamente completado. Los requisitos que no se completaron se dejarán para una replanificación.

¿Por qué son buenas las demostraciones en Scrum?

El tema más crítico en Scrum es el fin del Sprint.

Al terminarse una iteración se hace una demostración del producto. Para lograrlo se para el desarrollo durante una semana, tiempo en que el equipo se concentra en llevar adelante la demostración y que le permite ver varios temas de calidad o imprevistos, por ejemplo cuando han trabajado varios equipos aisladamente, en diferentes aspectos/componentes del producto o proyecto, pudiendo desatender la interacción entre ellos. La preparación de la demostración les ayuda a articularlos.

En principio se crearán “parches” que se utilizaran en la demostración, pero luego se planificarán soluciones concretas para el siguiente Sprint.

Se busca la practicidad con la creación de un producto que puede entregarse al cliente con el mínimo esfuerzo.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Los miembros del equipo son quienes realizan la demostración, no así el Scrum Master, lo que los compromete más en la misma.

La demostración se hace para toda la empresa. Demostrar que se cumplieron los objetivos determinados para el Sprint levanta la moral del equipo que puede haber decaído en el esfuerzo del Sprint.

El cliente puede corroborar que se progresa gracias a la demostración y conformar todos sus requisitos y sugerir los cambios que considere necesarios.

Sprint Retrospective o Retrospectiva

Esta reunión se realiza después de la demostración al cliente.

Preguntas clave que deberá hacerse el equipo

- Qué funcionó bien.
- Qué hay que mejorar.
- Qué intenta hacer en la siguiente iteración.
- Qué aprendió.
- Cuáles son los impedimentos para un adecuado progreso

La participación del Facilitador es muy importante para que el equipo pueda sortear los obstáculos que no pudo solucionar solo.

Beneficios del Sprint Retrospective:

Aumenta la productividad en el proyecto, la calidad del producto y se favorece la experiencia del equipo

- Incrementa la motivación del equipo porque participa en la mejora del proceso, es escuchado, decide de conformidad eliminando lo que impide ser más productivo.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Restricciones del Sprint Retrospective:

- Tanto el equipo como el Facilitador tienen que disponer de autoridad, medios y herramientas para mejorar su manera de trabajar y el entorno del proyecto. Desmoraliza encontrar siempre los mismos impedimentos sin poder solucionarlos.

Beneficios de Scrum

Beneficios: Calidad y competitividad

- Gestión de **expectativas del cliente**, basada en **resultados tangibles**.
- **Flexibilidad a cambios**: adaptación a las necesidades del cliente, a cambios en el mercado, etc.
- Gestión del **ROI**.
- Soporte a la **innovación**.
- Resultados anticipados (**time to market**).
- **Mantenibilidad** y **productividad**
- **Mitigación de riesgos**.
- **Equipo motivado** implicado, creativo y que **disfruta**.

REGULAR

MONEY MAKER

www.agile-spain.com

Fig. 25: Beneficios de Scrum.

Fuente: *Beneficios de Scrum* <<http://www.proyectosagiles.org/beneficios-de-scrum#expectativas>>

Relación Cliente Proveedor:

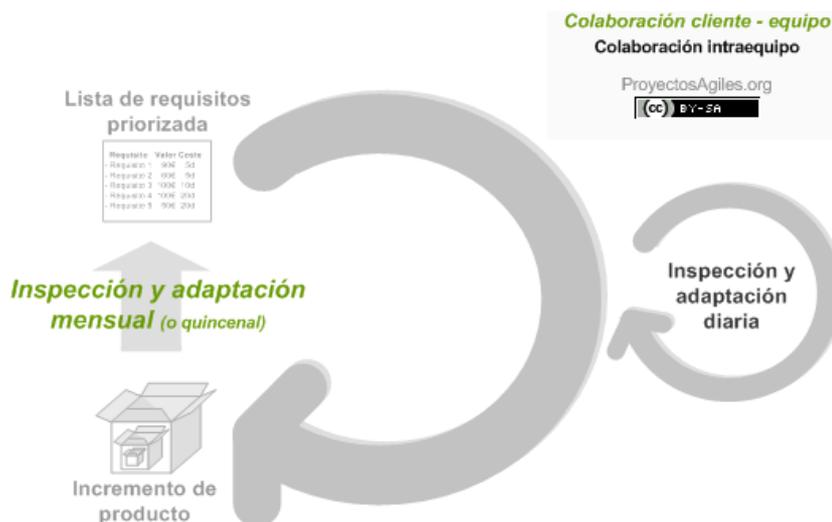


Fig. 26: Relación Cliente Proveedor

Fuente: *Beneficios de Scrum* <<http://www.proyectosagiles.org/beneficios-de-scrum#expectativas>>

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| Beneficios de Scrum | Cómo se consiguen |
|---|---|
| <p>Gestión regular de las expectativas del cliente El cliente establece sus expectativas indicando el valor que le aporta cada requisito del proyecto y cuando espera que esté completado.</p> <p>El cliente comprueba de manera regular si se van cumpliendo sus expectativas, da feedback, ya desde el inicio del proyecto puede tomar decisiones informadas a partir de resultados objetivos y dirige estos resultados del proyecto, iteración a iteración, hacia su meta. Se ahorra esfuerzo y tiempo al evitar hipótesis.</p> | <p>Lista de requisitos priorizada El cliente crea y gestiona la lista de requisitos del producto o proyecto, donde quedan reflejadas sus expectativas a nivel de requisitos, valor, coste y entregas.</p> <p>Demostración de los resultados de proyecto en cada iteración Al final de cada iteración el equipo demuestra al cliente los requisitos que ha conseguido completar. Tras una inspección del resultado real del proyecto hasta ese momento, y considerando el esfuerzo que ha sido necesario para realizarlo, el cliente solicita los cambios que necesita y replanifica el proyecto.</p> |
| <p>Resultados anticipados (“time to market”)</p> <p>El cliente puede empezar a utilizar los resultados más importantes del proyecto antes de que esté finalizado por completo.</p> <p>Siguiendo la ley de Pareto (el 20% del esfuerzo proporciona el 80% del valor), el cliente puede empezar antes a recuperar su inversión (y/o autofinanciarse) comenzando a utilizar un producto al que sólo le faltan características poco relevantes, puede sacar al mercado un producto antes que su competidor, puede hacer frente a urgencias o nuevas peticiones de clientes, etc.</p> | <p>Priorización de requisitos por valor y coste</p> <p>Al inicio de cada iteración el cliente prioriza la lista de requisitos del producto o proyecto en función del valor que le aportan, su coste de desarrollo y los riesgos del proyecto, cambiando los requisitos previstos para reaccionar a cambios de contexto en el proyecto.</p> <p>El progreso del proyecto se mide en función de los requisitos que el equipo completa en cada iteración.</p> |
| <p>Flexibilidad y adaptación De manera regular el cliente redirige el proyecto en función de sus nuevas prioridades, de los cambios en el mercado, de los requisitos completados que le permiten entender mejor el producto, de la velocidad real de desarrollo, etc. Al final de cada iteración el cliente puede aprovechar la parte de producto completada hasta ese momento para hacer pruebas de concepto con usuarios o consumidores y tomar decisiones en función del resultado obtenido.</p> | <p>Replanificación en el inicio de cada iteración Se asume que los cambios son parte natural del proyecto. Toda iteración comienza con una replanificación del proyecto. Esta replanificación no es traumática puesto que Scrum minimiza el número de objetivos/requisitos en que el equipo trabaja (WIP, Work In Progress) a los que caben en una iteración. Todavía no se ha hecho ningún esfuerzo en desarrollar los requisitos de las siguientes iteraciones. El hecho los requisitos se completen en función del valor que aportan al cliente minimiza la probabilidad de que se produzcan grandes cambios en el transcurso del proyecto.</p> |
| <p>Retorno de inversión (ROI) De manera regular, el cliente maximiza el ROI del proyecto. Cuando el beneficio pendiente de obtener es menor que el coste de desarrollo, el cliente puede finalizar el proyecto.</p> | <p>Priorización de requisitos por valor Cada iteración el cliente dispone de unos requisitos completados y replanifica el proyecto en función del valor que le aportan los requisitos pendientes respecto del coste de desarrollo que tienen.</p> |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | |
|---|---|
| <p>Mitigación de riesgos</p> <p>Desde la primera iteración el equipo tiene que gestionar los problemas que pueden aparecer en una entrega del proyecto. Al hacer patentes estos riesgos, es posible iniciar su mitigación de manera anticipada. "Si hay que equivocarse o fallar, mejor hacerlo lo antes posible". El feedback temprano permite ahorrar esfuerzo y tiempo en errores técnicos.</p> <p>La cantidad de riesgo a que se enfrenta el equipo está limitada a los requisitos que se puede desarrollar en una iteración. La complejidad y riesgos del proyecto se dividen de manera natural en iteraciones.</p> | <p>Desarrollo iterativo e incremental</p> <p>Un requisito se debe completar en una iteración. El equipo debe realizar todas las tareas necesarias para completarlo y que esté preparado para ser entregado al cliente con el esfuerzo mínimo necesario. De esta manera no se deja para el final del proyecto ninguna actividad arriesgada relacionada con la entrega de requisitos.</p> |
| <p>Productividad y calidad</p> <p>De manera regular el equipo va mejorando y simplificando su forma de trabajar.</p> <p>Los miembros del equipo sincronizan su trabajo diariamente y se ayudan a resolver los problemas que pueden impedir conseguir el objetivo de la iteración. La comunicación y la adaptación a las diferentes necesidades entre los miembros del equipo son máximas (se van ajustando iteración a iteración), de manera que no se realizan tareas innecesarias y se evitan ineficiencias.</p> <p>Las personas trabajan más enfocadas y de manera más eficiente cuando hay una fecha límite a corto plazo para entregar un resultado al que se han comprometido. La consciencia de esta limitación temporal favorece la priorización de las tareas y fuerza la toma de decisiones.</p> <p>Las iteraciones (Sprints) son regulares y de un mes para facilitar la sincronización sistemática con otros equipos, con el resto de la empresa y con el cliente. El equipo minimiza su dependencia de personas externas para poder avanzar (dependen de la disponibilidad de otros puede parar tareas).</p> <p>La estimación de esfuerzo y la optimización de tareas para completar un requisito es mejor si la realizan las personas que van a desarrollar el requisito, dadas sus diferentes especializaciones, experiencias y puntos de vista. Asimismo, con iteraciones cortas la precisión de las estimaciones aumenta.</p> <p>Las personas trabajan de manera más eficiente y con más calidad cuando ellas mismas se han comprometido a entregar un resultado en un momento determinado y deciden cómo hacerlo, no cuando se les ha asignado una tarea e indicado el tiempo necesario para realizarla.</p> | <p>Mejora continua</p> <p>Cada iteración el equipo realiza una retrospectiva para analizar su manera de trabajar e identificar los obstáculos que le impiden avanzar al mejor ritmo posible.</p> <p>Comunicación diaria del equipo</p> <p>Todo miembro del equipo conoce cómo el trabajo de los otros miembros impacta en el suyo y cuáles son las necesidades de los otros.</p> <p>TimeBoxing Cada actividad de Scrum siempre tiene la misma duración (1 mes, 4 horas, etc.), con lo que las personas aprenden lo que pueden conseguir en este tiempo, cómo organizarse, priorizar tareas y tomar decisiones.</p> <p>Equipo multidisciplinar El equipo está formado por todas las personas con las especialidades necesarias para llevar a cabo el proyecto</p> <p>Estimación de esfuerzo conjunta En el inicio de la iteración los miembros del equipo estiman de manera conjunta el esfuerzo necesario para completar requisitos y sus tareas.</p> <p>Compromiso del equipo En el inicio de cada iteración el equipo selecciona los requisitos que se compromete a completar y entregar al final de la iteración (responsabilidad). El propio equipo se organiza (autoridad) identificando las tareas necesarias, su esfuerzo y autoasignándose cada miembro las tareas que se compromete a realizar</p> |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | |
|---|---|
| <p>El equipo se evita caminar mucho tiempo por un camino equivocado que le obligue a realizar un gran esfuerzo para llegar al objetivo esperado</p> | <p>Demostración de resultados preparados para ser utilizados y velocidad sostenida</p> <p>Por un lado, al final de cada iteración el equipo demuestra al cliente los requisitos que ha conseguido completar, de manera que están completamente operativos. Por otro lado, para tener una velocidad de desarrollo sostenida, el equipo necesita desarrollar cada incremento de producto sin tener que visitar aspectos mal resueltos en iteraciones anteriores.</p> |
| <p>Alineamiento entre cliente y equipo</p> <p>Los resultados y esfuerzos del proyecto se miden en forma de objetivos y requisitos entregados al negocio. Todos los participantes en el proyecto conocen cuál es el objetivo a conseguir. El producto se enriquece con las aportaciones de todos.</p> | <p>Cliente y equipo trabajando “en equipo”</p> <p>Cada iteración el equipo y el cliente trabajan juntos en la creación de los requisitos del proyecto (en la estimación de la lista priorizada de requisitos del proyecto), en darles detalle (en la reunión de planificación de la iteración) y en el análisis del resultado obtenido (en la demostración de los requisitos completados).</p> |
| <p>Equipo motivado</p> <p>Las personas están más motivadas cuando pueden usar su creatividad para resolver problemas y cuando pueden decidir organizar su trabajo.</p> <p>Las personas se sienten más satisfechas cuando pueden mostrar los logros que consiguen.</p> | <p>Equipo autogestionado</p> <p>El equipo es quien se compromete a completar unos requisitos determinados en una iteración y quien mejor sabe cómo desarrollarlos. Por ello es el equipo quien se autoorganiza y quien planifica cómo trabajará en la iteración.</p> <p>Demostración</p> <p>Cada iteración el equipo muestra al cliente los resultados que consigue. No está meses trabajando sin poder exhibir su obra.</p> |

Tabla 10: Cómo se consiguen Beneficios de Scrum.

Fuente: *Beneficios de Scrum* <<http://www.proyectosagiles.org/beneficios-de-scrum#expectativas>>

DSDM Framework - Características Destacadas

El framework de DSDM maneja controles dedicados a la entrega de productos de alta calidad en función de las necesidades de negocio considerando el lugar y el tiempo. Esto se logra por usar las variadas técnicas en el framework además de requisitos flexibles.

El propósito consiste en tratar las necesidades actuales e inminentes del negocio más que atacar todas las posibilidades.

DSDM considera que no puede construirse nada perfecto en una primera instancia, pero que, si se acepta la Regla de Pareto, con un 80% del costo del desarrollo, en tiempo y recursos, se producirá un 20% del producto de software, y que el 80% restante demandará el 20% del esfuerzo. El Principio 4, enunciado en la tabla de Principios de DSDM, que figura en el Cap III, es fundamental para la aceptación de entregables. Esto marca una gran diferencia con otros enfoques pues no se remite a una especificación de requisitos.

Requisitos flexibles

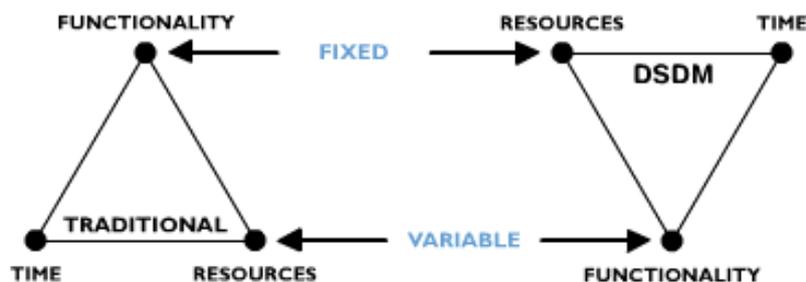


Fig. 27: Requisitos flexibles. Fuente: *DSDM Tour*. <www.dsdm.org>

En las metodologías tradicionales se busca satisfacer los requisitos incluidos en un documento, de acuerdo a entregables o prestaciones anteriores. Pero los requisitos son frecuentemente inexactos, podrían ser defectuosas las prestaciones anteriores y las necesidades de toda empresa son dinámicas por lo que pueden cambiar, incluso durante el proyecto. Las variaciones de tiempo y recursos que pueden ser permitidas, producen

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

retraso en la entrega y mayores gastos. DSDM fija el tiempo para la duración de un proyecto y trata de fijar los recursos, cuando puede. Por lo tanto los requisitos que lleguen a ser satisfechos pueden cambiar así como las necesidades del negocio a podrán cambiarlos a ellos.

People & Team o Gente y Equipo

Cuando DSDM es adoptado ampliamente, puede alcanzar un cambio cultural en la corporación, logrando un ambiente de trabajo más comunicativo, productivo y franco.

El framework pone el énfasis fundamentalmente en el factor gente, Principio 1 de la Tabla de Principios de DSDM, presentada en el CAP..III, porque la gente es la mayor razón por la cual fracasan algunos proyectos.

Hay dos roles fundamentales para un proyecto DSDM: Equipo y Ambassador User

Equipo:

Consiste de desarrolladores y usuarios que trabajan juntos en forma colaborativa.

Ambassador User o Usuario Embajador:

Son clave en un proyecto DSDM. Funcionan de igual forma que los diplomáticos. Se mantienen por algún tiempo junto al equipo del proyecto y facilitan la comunicación entre el negocio y el equipo de desarrollo. Proporcionan información para el proyecto y aseguran que toda la comunidad de negocio se actualice con el progreso. Trabajan cerca del Team Leader o Líder del Equipo, los Desarrolladores, los Testadores etc., y traen consigo información e ideas de usuarios. Cumplen también el rol de usuario coordinador asegurando que la solución sea la efectiva en función de las necesidades concretas de los usuarios.

Facilitated Workshops o Talleres Facilitados

Los Facilitated Workshops constituyen una técnica base en DSDM, que asegura decisiones de alta calidad del equipo de base. Son breves y eficientes. Ayudan a efectivizar el cambio cultural en una organización, porque promueven la aceptación y autonomía de los participantes. Pueden fijar el tono para el proyecto entero. Pero, corresponde a los miembros del proyecto mismos decidir si un taller es necesario o más aplicable otra técnica, como por ejemplo de entrevistas o de investigación. Son gestionados por el Facilitator o Facilitador de Taller: responsable del contexto del taller pero no del contenido ni del resultado. Acuerda con el responsable del Taller el alcance y el plan del taller. Se familiariza con el área temática. Participa del taller para entender las principales áreas de interés o preocupación. Ayuda a que el taller cumpla sus objetivos.

Beneficios de los Workshops:

Usar workshops trae beneficios directos e indirectos a un proyecto:

- Rápida, toma de decisiones que propenden a la calidad: todos los stakeholders están presentes al mismo tiempo, existe una gran confianza en el resultado. El grupo se centra en los objetivos de la sesión para la recopilación de información y el ciclo de revisión se lleva a cabo a una velocidad mayor. Los malentendidos y desacuerdos se pueden resolver y las preocupaciones planteadas son resueltas o señaladas al final del taller
- Gran usuario de buy-in: Talleres, efectivamente ejecutados, llevan a los participantes a sentirse más involucrados en el proyecto y decisiones que se toman. Ellos construyen y mantienen el entusiasmo
- Construir el espíritu de equipo. Se trata de una forma controlada de la construcción de relaciones. Puede promover la comprensión y la cooperación entre los departamentos, lo cual es particularmente importante cuando se trata del desarrollo de muchos grupos.
- Rediseño del Proceso por la comunidad de usuarios. Si las prácticas se revisan como resultado de un taller, los participantes pueden obtener un mayor entendimiento de las entradas y las implicaciones de su trabajo.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Aclaración de los requisitos cuando no están claros: Esclarecimiento de los requisitos cuando no están claros. Los usuarios de negocios pueden conducirse a través de sus objetivos y procesos para definir lo que puedan necesitar.

Técnicas básicas

Dado un determinado proyecto son varias las técnicas de desarrollo que pueden aplicarse. Ninguna técnica es aplicable de la misma manera a todos los proyectos así como no todas las técnicas pueden o deben utilizarse en todos los proyectos. DSDM Framework plantea técnicas que son explicadas en términos de lo que es importante considerar al aplicarlas en el DSDM Business Centred Development, mencionado en el CAP. II, proporcionándose una orientación práctica sobre cómo y cuándo deben ser utilizadas en los proyectos.

Enumeración de las técnicas básicas que se presentaron en el CAP. III

En cualquier proyecto:

1. Business Case: Es una tarea muy importante que realiza DSDM, como se vio en el CAP.II, ayuda al cliente a analizar una propuesta, considerando el impacto financiero que resultará de tomar una u otra decisión, con mayor conocimiento de causa/efecto.
2. Timeboxing: el timebox es subdividido en pequeños timebox. Facilita la flexibilidad de los requisitos en DSDM.
3. MoSCoW: para clasificar los requisitos.
4. Prototipado: permite descubrir, con anterioridad, deficiencias del sistema
5. Testeo: se crea un sistema de buena calidad. Se aplican tests a lo largo de cada iteración.
6. Facilited Workshops o Talleres Facilitados: ayuda a las partes interesadas a discutir necesidades y funcionales y colabora con la comprensión mutua.
7. Modelado: técnica esencial utilizada para visualizar una representación gráfica de aspectos específicos del sistema o área de negocio que será trabajada.
8. Gestión de la Configuración: conjunto de actividades para gestionar todos los cambios a lo largo del ciclo de vida.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

En algunos proyectos en particular:

1. Best Practice: sirve para medir el rendimiento de un componente, como se mencionó en Técnicas, de DSDM, Cap. III.
2. Exámenes: técnica independiente que sirve para medir lo alcanzado en cada iteración.

¿Cuándo usar DSDM?

Dado un proyecto de desarrollo, decidir si utilizar o no DSDM es difícil e importante. Algunos proyectos resultarán ideales para DSDM otros deberán ser sometidos a una cuidadosa consideración para determinar si será beneficioso para el negocio el uso de DSDM.

Cada organización, tiene una cultura definida y rutinas de trabajo aceptadas y en algunos casos surgirán, de la misma, opositores a la introducción de DSDM. En este sentido, es importante que trabajen IT y la Business Management para mostrar que los beneficios de negocio serán mayores que los costos.

Suitability Checklist o Listado de Control de Aptitudes

DSDM facilita un conjunto de criterios, aptitud/riesgo, para diagnosticar los riesgos que deben asumirse al aplicar el enfoque DSDM Framework.

Que el proyecto no cumpla todos los criterios no implica el descartar el uso de DSDM. Podría implementarse una gestión de riesgo, con el fin de afrontar los riesgos.

Tailoring o Adaptación

Para cada tipo de proyecto DSDM facilita un marco de ejecución acomodado a la organización en cuestión.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| Tipos de Proyecto | |
|---|---|
| Pequeño | DSDM tiene como objetivo la entrega rápida y es muy bien adaptado para pequeños proyectos. Puede haber una necesidad de reducir el conjunto de productos y de combinar los roles. |
| Grande | Cuando los proyectos son "grandes" alguna consideración especial debe prestarse a la gestión y control de los aspectos tales como el aumento de la formalidad de la estructura organizativa, procedimientos, comunicaciones, etc., manteniendo conceptos de trabajo de DSDM, como por ejemplo equipos facultados. |
| Híbrido | No todas las partes del proyecto podrán abarcarse plenamente por DSDM: algunas partes no lograrán respuestas afirmativas suficientes en la lista de aptitud / riesgo. Estos proyectos necesitarán utilizar técnicas DSDM y cascada en conjunto para lograr los mejores resultados. |
| Business Change Project o Proyecto de Cambio de Negocio | DSDM es aplicable en este tipo de proyectos, pero son necesarios algunos cambios del marco general en los objetivos, actividades y productos. |

Tabla 11: Tipos de Proyectos. Fuente: *DSDM Tour* < www.dsdm.org>

Usando DSDM con otros Métodos

El DSDM Framework asegura que los proyectos se entreguen a tiempo y dentro del budget. No gestiona proyectos. A menudo se utiliza con otros enfoques para facilitar la entrega del proyecto y gestionar la intervención del usuario.

The DSDM Agile Project Framework for Scrum

El DSDM Agile Project Framework for Scrum, publicado en el 2012, ya mencionado en el CAP. II, introduce principios de DSDM focalizados en proyectos, agrega un conjunto de roles y responsabilidades de DSDM muy importantes para un entorno de proyecto

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

organizacional y plantea el uso de sólidas y ágiles técnicas para fundar y probar el control de un proyecto.

Para la entrega utiliza una versión acotada del proceso de DSDM, con la idea de complementar la forma de entrega del producto Scrum, sin cambiar de enfoque.

The DSDM Agile Project Framework for Scrum mantiene la misma línea de DSDM sobre los cuatro principios del Manifiesto Desarrollo Ágil de Software, mencionados en el CAP. II.

DSDM apuntala a The DSDM Agile Project Framework manteniendo su filosofía con los 8 principios de DSDM Atern que se enumeran en el CAP.III.

Variables

Gestionar proyectos implica considerar tres parámetros clave que deben estar balanceados: tiempo, costo y calidad. Agile Project Framework sigue a DSDM, luego tiempo, costo y calidad se fijan al establecer las Fundaciones para el proyecto. Se permite la variación del alcance.

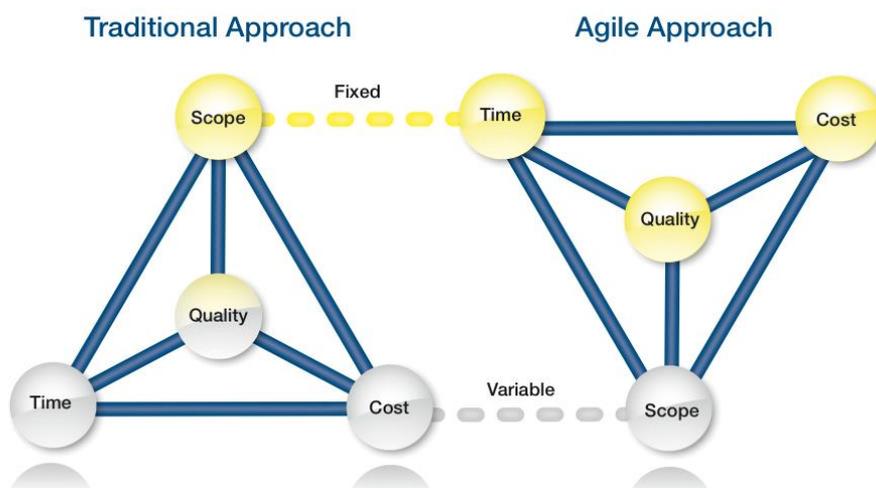


Fig. 28: Parámetros balanceados. Fuente: *The DSDM Agile Project Framework for Scrum* (2012)

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Lifecycle

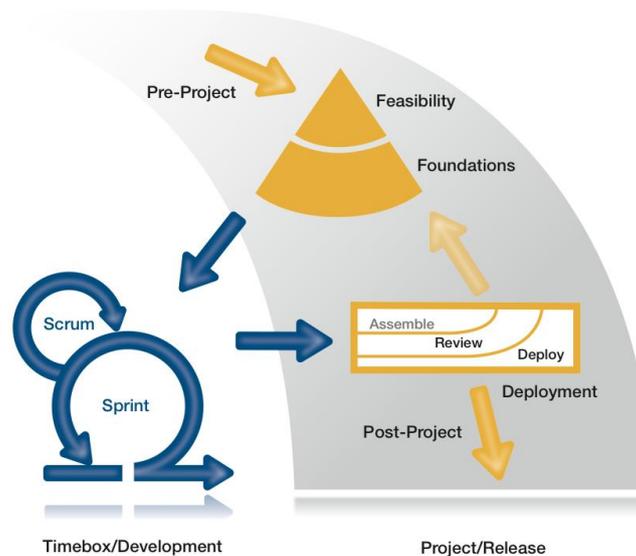


Fig. 29: Ciclo de Vida. Fuente: *The DSDM Agile Project Framework for Scrum* (2012)

El DSDM Agile Project Framework focaliza en dos elementos: la gestión del proyecto y la entrega del producto.

El Lifecycle consta de siete fases:

- Pre-Project: formaliza una propuesta y prioriza en el entorno de otros trabajos realizados por la empresa en función de sus propósitos estratégicos.
- Feasibility o Factibilidad: para analizar la posibilidad del proyecto, aunque la viabilidad debe evaluarse continuamente para asegurar los beneficios previstos.
- Foundations o Fundaciones o Implantaciones: instaura una base sólida y duradera de comprensión del proyecto, desde tres perspectivas: negocio, solución y gestión, de manera tal de suministrar un enfoque robusto y flexible.
- Evolutionary Development o Desarrollo Evolutivo: tiene un encuadre iterativo e incremental para que la solución evolucione como un todo.

“En todo momento, la participación continua del Product Owner proporciona una oportunidad para orientar el desarrollo y validar la aptitud para el propósito del producto. Los roles del negocio desde DSDM (el Business Sponsor, el Business Visionary y los Business Advisor(s)), a menudo serán de gran valor en un complejo entorno corporativo donde no es posible para una sola persona, el Product Owner, comprender todos los detalles de todos los aspectos de los requisitos de un producto de

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

software, así como la comprensión de todas las implicancias de la forma en que el negocio puede necesitar cambiar con el fin de explotar esto. La perspectiva y aportes de los otros roles de DSDM bien pueden ser valiosos para el Product Owner y el resto de la Scrum Team quienes pueden confiar en el Business Visionary para proporcionar un contexto de alto nivel y manejar los aspectos de cambios del negocio de un proyecto más amplio y en los Business Advisors proporcionar profundidad y detalle verdadero en torno a requisitos especiales tales como el cumplimiento de legislación específica o en la práctica de trabajo”.(The DSDM Agile Project Framework for Scrum, 2012)

- Deployment o Despliegue o Implementación: El foco está puesto en lograr la solución operativa o lista para el mercado. Hay que considerar tres aspectos: el ensamblaje final (configuración/envases) del producto, una revisión del producto final para ser entregado y la implementación real.
- Post-Project: se realiza después del último despliegue previsto. Consiste de la evaluación del rendimiento del proyecto.

Roles

Los roles son los mismos que los de Atern, ver ANEXO I DSDM Atern, divididos en tres grupos: del Proyecto, de Desarrollo de Soluciones y otros.

En el DSDM Agile Project Framework for Scrum:

- El rol del Team Leader lo desempeña el Scrum Master, asegurando un enfoque iterativo y colaborativo en el Scrum Team.
- El rol de Business Ambassador lo realiza el Product Owner
- El rol de Atern Coach de Atern es el de Agile Coach y lo realiza en parte el Scrum Master: consiste en ayudar al equipo, si corresponde, a por ejemplo aumentar su conocimiento sobre Agile, cuando la experiencia del equipo está limitada sobre el uso de Agile Project Framework.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

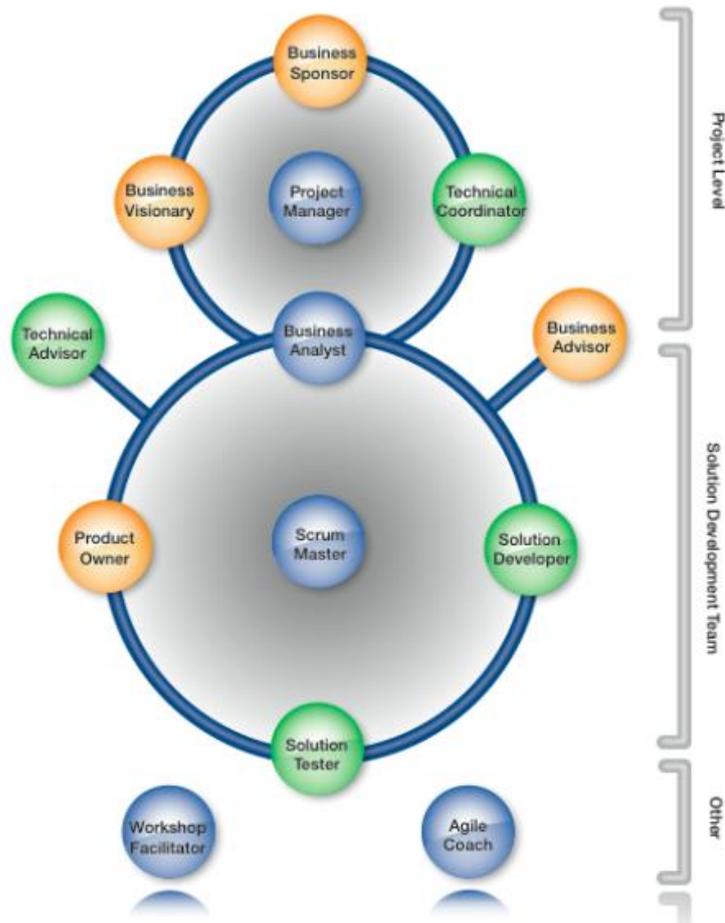


Fig. 30: Tres grupos de roles

Fuente: *The DSDM Agile Project Framework for Scrum* (2012)

Artefactos

Son ocho los artefactos del DSDM Agile Project Framework for Scrum, que a veces varían de proyecto en proyecto “pero que deben ser activamente considerados para cada proyecto” (*The DSDM Agile Project Framework for Scrum*, 2012). Algunas organizaciones necesitan suplir este conjunto de artefactos por otros.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

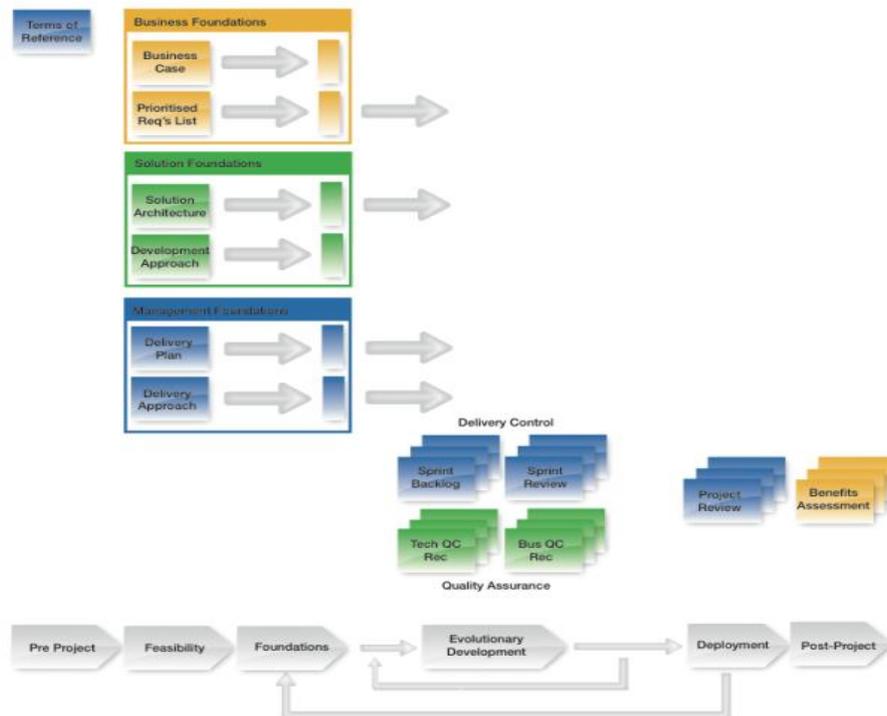


Fig. 31: Artefactos. Fuente: *The DSDM Agile Project Framework for Scrum* (2012)

- Terms of Reference o Términos de Referencia: “Definen en un alto nivel los objetivos y controladores del negocio para el propósito del proyecto”. (Craddock Andrew et al., 2012) Incluyen especificaciones técnicas, objetivos y estructura sobre cómo implementar un proyecto dado o también: un estudio o trabajo o conferencia o negociación, etc.. Detallan, describen y/o definen: el ámbito espacial, el objetivo general, los objetivos específicos, el tiempo disponible, los requisitos de los técnicos que van a participar (especialidades y tiempos mínimos de dedicación), los resultados esperados, contenidos de los informes, visión, componentes, roles, responsabilidades, recursos, finanzas, plan de calidad, desglose del trabajo, calendarización y factores de éxito y/o riesgos y las restricciones.
- Business Foundations o Fundaciones del Negocio: al principio, durante la fase de Feasibility, se bosquejan, luego son refinadas para lograr una firme y duradera base del proyecto todo hasta el final sin ser demasiado específicas ni limitadas.

Comprende:

- Business Case
- Product Backlog, equivalente a Prioritised Requirements List o Lista Priorizada de Requerimientos.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Solution Foundations o Fundaciones de la Solución: durante la fase de Feasibility se bosquejan como las Business Foundations.

Comprende:

- Solution Architecture Definition o Definición de la Arquitectura de la Solución
- Development Approach Definition o Definición del Enfoque del Desarrollo.
- Management Foundations o Fundaciones de la Gestión: durante la fase de Feasibility se bosquejan como las Business Foundations.

Comprende:

- Delivery/Release Plan o Plan de Entrega/Release
- Delivery Approach Definition o Definición del Enfoque de la Entrega.
- Delivery Control o Control de Entrega: los artefactos para definir y controlar el resultado de un Sprint.

Comprende:

- Sprint Backlog
- Sprint Review record o Registro del Sprint Review.
- Quality Assurance o Aseguramiento de la Calidad: Scrum propugna una Done Definition o Definición de Dado para guiar al Equipo Scrum hacia el éxito en el incremento del producto.

Comprende:

- Technical Quality Review Records o Registros de las Revisiones de Calidad Técnica, artefacto que suministra un registro de las actividades principales de revisión y tests técnicos y cualquier trazabilidad para la resolución esperada para problemas y defectos,
- Business Quality Review Records o Registros de las Revisiones de Calidad del Negocio, todos los registros relativos a revisión del negocio y aceptación de la solución como it evolves
- Project Review o Revisión del Proyecto: a la terminación de cada Release con el objeto de reflejar el cumplimiento y aprendizaje del proyecto
- Benefits Assessment: Reseñan cómo los beneficios pronosticados por el Business Case se cumplen en la Deployed Solution o Solución Entregada

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Techniques

“Las siguientes prácticas de DSDM son técnicas comprobadas que pueden ser utilizadas por el equipo del proyecto para ayudar en forma práctica y predecible en una gestión de gobierno, proyecto corporativo, programa o portfolio. Esto permite proporcionar el nivel de escalabilidad y rigor adecuado, ya sea que el proyecto sea pequeño y simple o grande y complejo”.(Craddock Andrew et al., 2012:19)

- Facilitated Workshops: Resultan valiosos para actividades como: identificación y refinamiento de requisitos, priorización, Release o Sprint Planning, análisis de riesgo, solución de problemas, revisión de productos y retrospectivas.
- Modelling: Los modelos pueden más o menos formales en función de la índole del proyecto y de la capacidad y experiencia del equipo en técnicas de modelado. Variando también de acuerdo al tipo de proyecto, normas vigentes y Best Practice. Las reglas más simples para optar por qué modelo crear y cuándo, sugeridas por DSDM Agile Framework for Scrum son, en otras palabras:
 - Tener la aptitud de testimoniar el valor del modelo para optimizar la comprensión del tema estipulado
 - Usar un enfoque efectivo para el cliente y su organización
 - Hacer sólo lo necesario para alcanzar el objetivo del modelo.
- MoSCoW Prioritisation applied to the Product Backlog
Son las mismas detalladas en el CAP. III.
- DSDM Timeboxing aplicado a un Sprint: Toda la actividad a describir a continuación se requiere para cualquier Sprint. Cada paso colabora en la construcción de una estructura que permite al equipo controlar el trabajo.



Fig. 32: Timeboxing Aplicado a un Sprint.

Fuente: *The DSDM Agile Project Framework for Scrum* (2012)

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Kick-Off: En este momento el equipo acuerda qué considerar del Product Backlog.
- Sprint Planning and Investigation o Planificación del Sprint e Investigación: Ahora el Scrum Team comprende los detalles de los requisitos que debe cumplir y va a afrontar como parte de la Evolving Solution o Solución de la Evolución. El equipo entregará uno o más productos al final del Sprint, que se basarán en su investigación detallada considerando los recaudos a tomar para un buen logro. Al terminar la investigación el Scrum Team a pleno y otros interesados deben concurrir a una revisión, la que tiene por objeto aseverar que tanto el Equipo como cualquier interesado que haya asistido tengan una completa y acordada comprensión de lo que el Equipo planea hacer sobre los requisitos del Sprint.
- Product Refinement: Aquí es donde se lleva a cabo la mayor parte del trabajo del Sprint. El desarrollo y las pruebas correspondientes deberían estar casi terminados al final de esta etapa. En ese momento se realiza una importante revisión para ver qué entregable se creó y cuáles arreglos se emplearon para lograr los criterios de aceptación. A la misma deben asistir el Product Owner y cualquier otra persona implicada en la aceptación del producto de forma tal de proporcionar sus feedbacks y colaborar en priorizar el trabajo en el Sprint Consolidation.
- Sprint Consolidation: Toda acción acordada al final del Refinement se hará simultáneamente con cualquier trabajo final solicitado para cumplimentar los estándares de la organización o proyecto. Se completa la prueba final y si algún producto no aprobó los tests no será entregado.
- Sprint Review: Comúnmente el Sprint Review es como una Demostración del incremento del producto entregado (ver CAP. III).

Todo lo que no llegue a entregarse efectivamente, a pesar de haber sido prevista la entrega, será considerado para algún Sprint posterior en la siguiente sesión de planeamiento.
- Sprint Retrospective: Ofrece al equipo la posibilidad de comprobar la eficacia de la labor por ellos desarrollada en el Sprint, atendiendo:
 - si se alcanzaron o no los cánones de calidad propuestos y/o si cualquier otro resultado los lleva a entender que ciertos aspectos de la solución no son lo suficientemente aptos para la propuesta.
 - los motivos de por qué los requisitos no fueron alcanzados.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

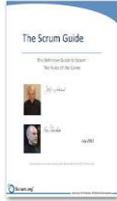
Se volcará todo esto en un Sprint Review Record o Registro de la Revisión de la Iteración con un resumen de lo entregado incluyendo las carencias o deficiencias notorias incurridas.

DSDM Roadmap

En la siguiente página se presentará el DSDM Roadmap, ya mencionado en el CAP. II como parte de la forma de trabajo por parte de DSDM Atern, ver ANEXO DSDM Atern, que puntualiza Chris Fortuin y publicó en su página de Orange Fortune. Justamente el DSDM Roadmap revela la afirmación de Fortuin en el sentido de ante un proyecto es mejor aplicar la sinergia entre Scrum y DSDM Atern.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

DSDM Scrum Roadmap

| | DSDM® Atern | Scrum.org™ |  |
|---|---|--|--|
| Propósito | Agile Project Management Framework proporciona la guía de Best Practice para la entrega de proyectos a tiempo y en budget en cualquier Business Sector. | Agile Process Framework para desarrollo y mantenimiento de productos complejos. | Lo Mejor de Ambos Mundos Entrega de proyectos de productos complejos en un entorno complejo. |
| Organismo Emisor | El DSDM Consortium es el guardián de la metodología DSDM® Atern la cual está libre para ver y usar. | Scrum.org es el hogar de Scrum y mantiene la Scrum Guide la cual está libre para ver y usar. | |
| Cuerpo de Conocimiento | Agile Project Management Handbook  | Scrum Guide  | |
| Entrenamiento y Certificación | AgilePM (Agile Project Management) formación y certificación proporcionada por APMG. | Agile Scrum Master Formación y certificación proporcionada por Yellowhouse. | |
| Responsabilidad de la entrega a tiempo y en budget *** Conflicto *** | El Project Manager es responsable de "todos los aspectos de la entrega de la solución " y de "la planificación de proyectos de alto nivel y la programación". | El Product Owner sigue "el avance hacia completar el trabajo previsto por el tiempo deseado para el objetivo". El Scrum Master "sirve al Product Owner en [...] la comprensión a largo plazo del planeamiento del producto en un entorno empírico". | Alineación: use Scrum Beneficio: dominio complejo implica muchas "incógnitas desconocidas" y responsabilidad para alcance/tiempo/costo necesarios para ser un representante de negocios siendo el Product Owner en Scrum Herramientas: Iron Triangle ⁽¹⁾ |
| Timeboxing con alcance de la | Priorizar el trabajo del proyecto con | Priorizar el trabajo usando orden | Alineación: use Scrum Beneficio: más |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | | |
|--------------------------------------|--|--|--|
| variable | MoSCoW (Must, Should, Could and Won't). | relativo. | flexible, simple y menos confuso |
| Niveles de Planificación | Tres niveles de planificación: releases, incrementos y timeboxes . | Dos niveles de planificación: releases e incrementos. | Alineación: use Scrum Beneficio: factible, más simple y predecible Herramientas: Agile Scrum Compass ⁽²⁾ |
| Gente versus roles | Proyectos con muchos posibles roles de proyecto y posiblemente con varias personas cumpliendo un rol. | Proyecto limitado a tres roles clave (Product Owner, Scrum Master and Developer) y con solo el rol de Desarrollador a cumplir por varias personas. | Alineación: use Scrum Beneficio: mejora de la comunicación |
| Timebox del Desarrollo | 1. Kick-Off (reunión de arranque del proyecto) 2. Investigación 3. Refinamiento 4. Consolidación 5. Close-Out (Liquidar) | 1. Sprint Planning - Parte 1 2. Sprint Planning - Parte 2 3. Sprint 4. Sprint Review 5. Sprint Retrospective | Alineación: use Scrum Beneficio: mejor definición |
| Timebox Diario | Diario Stand-Up | Diario Scrum | Alineación: use Scrum Beneficio: mejor definición |
| Roles del Proyecto | 1. Business Sponsor 2. Business Visionary 3. Business Ambassador 4. Project Manager 5. Workshop Facilitator 6. Atern Coach 7. Team Leader 8. Business Advisor 9. Solution Developer 10. Solution Tester | 1. N/A 2-4. Product Owner (nota: incluyendo la responsabilidad del product owner al DSDM Project Manager) 4-6. Scrum Master (nota: incluyendo la responsabilidad del process owner al rol del DSDM Project Manager) 7-10. Developer | Alineación: use Scrum Beneficio: mejor comunicación |
| Productos del Pre-Proyecto | 1. Términos de Referencia | N/A | Alineación: use DSDM® Atern |
| Factibilidad de los Productos | 1. Evaluación de la Factibilidad: Esquema del Business Case y esbozo de la Solución 2. Esquema del Plan | N/A | Alineación: use DSDM® Atern Beneficio: cubierto por DSDM® Atern Herramientas: Scrum- |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | | |
|--|--|---|---|
| | incluyendo Project Approach Questionnaire (PAQ) ⁽³⁾ | | But Test ⁽⁴⁾ , Agile Inception Deck ⁽⁵⁾ |
| Fundación de Productos | <ol style="list-style-type: none"> 1. Fundación del Negocio: Business Vision y Business Case 2. Fundamentos de la Gestión 3A. Fundaciones de la Solución: Business Area Definition (BAD)⁽⁶⁾, System Architecture Definition (SAD)⁽⁷⁾ y Prototipo de la Solución 3B. Fundación de la Solución: Development Approach Definition (DAD)⁽⁹⁾ 4. Lista priorizada de Requerimientos [basado en un alto nivel] 5. Plan de Entrega 6. Paquete de Control de Entrega | <ol style="list-style-type: none"> 1-3A. N/A 3B. Definición de Dado [base] 4. Product Backlog [línea de base con "Undone Work"⁽⁸⁾ para despliegue(s)] 5-6. Release Burn-Up/Down Chart [base] | <p>Alineación: use Scrum y complete con DSDM® Atern</p> <p>Beneficio: cubierto por DSDM® Atern o Scrum</p> <p>Herramientas: Scrum-But Test⁽⁴⁾, Agile Inception Deck⁽⁵⁾</p> |
| Exploración y Productos de Ingeniería | <ol style="list-style-type: none"> 1. Lista Priorizada de Requerimientos [actualizada] 2. Plan de Entrega 3. Paquete de Control de Entrega 4. Timebox Plan 5. Revisión del Historial del Timebox 6. Paquete de Seguridad de la Solución: Revisión del Historial de la Solución, Paquete de Pruebas de Negocio y Paquete de Pruebas Técnicas 7. Solución Evolucionando 8A. Plan de | <ol style="list-style-type: none"> 1. Product Backlog [preparado] 2-3. Release Burn-Up/Down Chart [actualizado] 4. Sprint Backlog 5. Product Backlog [actualizado para feedback del producto y mejora del proceso] 6. Definición de Dado [actualizado] 7. Incremento del Producto [Listo para su lanzamiento] 8A. Product Backlog [actualizado con "Undone Work"⁽⁸⁾ para despliegue(s)] | <p>Alineación: use Scrum y complemente con DSDM® Atern</p> <p>Beneficio: cubierto por DSDM® Atern o Scrum</p> <p>Herramientas: Scrum-But Test⁽⁴⁾</p> |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | | |
|-----------------------------------|---|---|--|
| | Despliegue ⁽¹⁰⁾ 8B. Plan de Despliegue ⁽¹⁰⁾ : Beneficios del Plan de Realización | 8B. N/A | |
| Productos de Despliegue | 1. Informe de Revisión del Proyecto: Revisión del Incremento, Resumen de Beneficios Habilitados y Fin de Evaluación de Proyectos 2. Solución del Despliegue 3. Plan de Despliegue ⁽¹⁰⁾ | 1. N/A 2. Incremento del Producto [lanzado] 3. Product Backlog [actualizado con "Undone Work" ⁽⁸⁾ para despliegues(s)] | Alineación: use Scrum y complemente con DSDM® Atern Beneficio: cubierto por DSDM® Atern o Scrum |
| Productos del Post-Project | 1. Evaluación de los Beneficios | 1. N/A | Alineación: use DSDM® Atern Beneficio: cubierto por DSDM® Atern |

Tabla 12: DSDM Scrum Roadmap.

Fuente: *DSDM Scrum Roadmap*. <<http://orangefortune.com/DSDMScrumRoadmap.html>>

Aclaraciones al Roadmap:

⁽¹⁾ Iron Triangle o Triángulo de Hierro, es tenido en cuenta por ambas metodologías

Considera las tres variables más importantes de todo proyecto:

- El alcance: cuántos requisitos o tareas hay que realizar
- El tiempo o planificación: cuánto durará el proyecto
- El coste o recursos: cuanto dinero, personas, etc. se dedicarán al proyecto.

Para mantener determinados objetivos de calidad, la modificación alguna de las tres variables implicará la modificación de alguna/s de las otras dos.

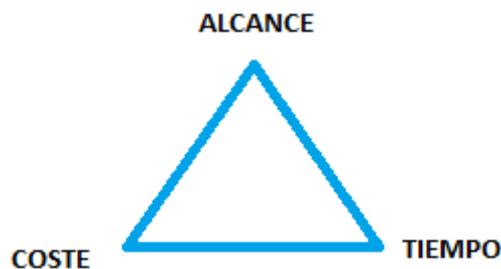


Fig. 33: Triángulo de Hierro Fuente: Elaboración propia

⁽²⁾ Agile Scrum Compass: es una herramienta para el Agile Scrum Master que sirve Product Owner, al Scrum Team y a los participantes del proyecto y seguimiento de los

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

proyectos Scrum Agile y sobre la que Chris Fortuin sugiere, en otras palabras, que uno se sienta libre de usarla y de transmitirla. Contiene:

Guía de usuario, en tres niveles de uso: fácil, intermedio y sofisticado

Un cuadro de mandos del proyecto para la planificación, el seguimiento y la inspección Burndown Charts, gráficos del trabajo pendiente, utilizables cuando la situación lo requiera.

⁽³⁾ Project Approach Questionnaire –PAQ: Se utiliza para gestionar los riesgos de la utilización de un enfoque ágil en una empresa. Consiste de 21 preguntas que revisan el Project Manager y el Business Sponsor, ver ANEXO I DSDM Atern.

⁽⁴⁾ Scrum-But Test: son las razones por las cuales los equipos no alcanzan a aprovechar al máximo a Scrum para solucionar problemas y efectivizar los beneficios de desarrollo de productos con el uso de Scrum. Los roles, reglas y timeboxes de Scrum fueron diseñados para obtener pretendidos beneficios y encarar reiterados problemas predecibles. Los ScrumButs indican que Scrum manifestó una disfunción y está coadyuvando al problema, aunque es muy difícil de solucionar. ScrumBut conserva el problema al rectificar Scrum de modo que no sea visible y así la disfunción no mortifica al equipo.

Ejemplos de ScrumBut que facilitó Ken Schwaber:

“(Usamos Scrum, pero)... (Tener un Daily Scrum cada día es demasiado trabajo), (de manera que tenemos uno por semana)”

“Usamos Scrum, pero... algunas veces nuestros managers nos dan alguna tarea especial, de manera que no siempre tenemos el tiempo para encontrar nuestra definition of done”

A veces las organizaciones hacen los cambios a corto plazo a Scrum para darles tiempo para corregir las deficiencias. Por ejemplo, "done" puede no incluir inicialmente regresión y pruebas de rendimiento, ya que tomará varios meses para desarrollar pruebas automatizadas. Para esos meses, la transparencia se ve comprometida, pero restaurada tan rápidamente como sea posible.

⁽⁵⁾ Agile Inception Deck: o Inception Deck o sólo Inception, es un conjunto de dinámicas con el fin de orientar a todas las personas comprometidas en un proyecto hacia un único objetivo, aplacando incertidumbres, explicitando los riesgos obvios y creando una expectativa común a todos. Se publicó en el libro “The Agile Samurai”, de Jonathan Rasmusson y en la conferencia “Agile 2011”. Se emplea al inicio de cualquier proyecto, no sólo de software, pero se destaca especialmente en el inicio de un proyecto

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

de emprendimiento porque ayuda a la definición del MVP o *Mínimum Viable Product* o *Producto Mínimo Viable*, que es una estrategia para la prueba de rapidez y cuantificación en el mercado. Este término se debe a Frank Robinson (2012) y difundido por Eric Ries para aplicaciones web. En la práctica es un previo análisis de mercado.

⁽⁶⁾ *Business Area Definition (BAD)* o *Definición del Área de Negocio*: “Identifica los requisitos de alto nivel y proporciona una descripción del proceso del producto final”.
<<http://www.philadelphia.edu.jo/>>

⁽⁷⁾ *System Architecture Definition (SAD)*: Como se sabe, la arquitectura de un sistema es “La organización fundamental de un sistema, que incluye componentes, las relaciones entre sí y el ambiente y los principios que gobiernan su diseño y evolución” (From ANSI/IEEE 1471-2000). DSDM “esboza la arquitectura del sistema final”. (<<http://www.philadelphia.edu.jo/>>) Probablemente esto evolucionará a lo largo de la vida del proyecto.

⁽⁹⁾ *Undone work* o *Trabajo sin hacer*: “La última guía de Scrum de Ken Schwaber habla sobre esto, es el material que no se hace como parte de la historia, pero que todavía hay que hacer antes del release. Son las cosas que se hacen en un Sprint de liberación. Obviamente cuanto más pequeño es mejor, ya que está más cerca de 'potencialmente entregable'. Siempre he tenido un problema con este término. Entiendo la necesidad de que, sobre todo en equipos nuevos a Scrum, que, por ejemplo, podría no tener una suite de pruebas de regresión totalmente automatizada que se pueden ejecutar en cada historia. Sin embargo, nunca me ha gustado el hecho de que fuera un término formal en el Scrum Framework. Para mí esto un poco dice que está bien tener el trabajo sin hacer, mientras que me gustaría un equipo que luche por tener cero trabajo *undone work* o sin hacer cuando se termine una historia”. (Greaves Karen, 2010, 4 de febrero)

⁽⁹⁾ *Development Approach Definition (DAD)* o *Definición del Enfoque de Desarrollo*

⁽¹⁰⁾ *Plan de Despliegue*: Es la planificación de las tres acciones que abarca el despliegue: entrega, soporte y retroalimentación. “Como el Software moderno es evolutivo por naturaleza, el despliegue no se presenta una sola vez, sino varias veces conforme el software avanza hacia su terminación. Cada ciclo de entrega le proporciona al cliente y a los usuarios finales un incremento de software operativo que provee funciones y características útiles. Cada ciclo de soporte proporciona documentación y asistencia humana para todas las funciones y características introducidas durante todos los ciclos de despliegue que se han presentado hasta la fecha. Cada ciclo de retroalimentación

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

ofrece al equipo de software una guía importante que conduce a modificaciones en las funciones, características y el enfoque que se toma para el siguiente incremento”.(Pressman, 2006:128)

Características esenciales de las metodologías DSDM y Scrum

Ambas metodologías coinciden en presentar características propias del Desarrollo Agile:

- Equipos autogestionados
- Compromiso conjunto y colaboración de los miembros del equipo, incluyendo los stakeholders.
- Facilidad para realizar cambios en el proyecto.
- Desarrollo iterativo e incremental.
- Control empírico del proyecto.
- Entrega frecuente de productos.
- Cambios reversibles.

Características destacadas de Scrum:

- simplicidad
- ligereza de trabajo
- prontitud en la entrega
- las características de comunicación entre ellos y con el Scrum Master y el Product Owner.
- la autonomía del equipo,
- la falta de burocracia,
- la sencillez y continuidad de la comunicación,
- la realización de un BSC para medir el rendimiento estratégico.

Características destacadas de DSDM:

- fortaleza
- la solidez de su labor,
- realización de un Business Case, un Business Vissio y un PAQ,

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- definición del Business Área,
- la definición del System Architecture
- y el Prototipar la Solución
- los roles: Ambassador User, Business Visionary y Technical Coordinator
- la explicitación de los Términos de Referencia.

Cuadro comparativo entre DSDM y Scrum

La tabla que se presenta en la página siguiente constituye un Cuadro Comparativo entre DSDM y Scrum que pone de manifiesto las diferencias entre las mismas, deducidas de todo lo analizado en ambas metodologías.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| ITEMS | DSDM | SCRUM |
|--|--|--|
| Priorización de los requisitos | Priorización de los requisitos por valor para el cliente y coste de desarrollo en cada iteración según reglas de MoSCoW | Priorización de los requisitos por valor para el cliente y coste de desarrollo en cada iteración Sigue la ley de Pareto |
| Timeboxing | Timeboxing de las actividades del proyecto con subdivisión de cada uno en pequeños timeboxes , cada uno de los cuales pasa por tres fases: Investigación, Refinamiento y Consolidación , lo que facilita la flexibilidad de los requisitos | Timeboxing de las actividades del proyecto No los subdivide |
| Testeo | Se crea un sistema de buena calidad. Se aplican tests a lo largo de cada iteración. Es integrado a través del ciclo de vida. | Realiza un Agile BSC: controla calidad, costes, riesgos, la sostenibilidad de la velocidad. El BSC plantea una visión amplia y crítica. |
| Tailoring o Adaptación | Para cada tipo de proyecto DSDM facilita un marco de ejecución acomodado a la organización en cuestión. | N/A |
| Best Practice in the Marketplace | Sirve para medir el rendimiento de un componente. Por ejemplo, modelos de procesos de negocio, modelado orientado a objetos, modelos de navegación web, etc | N/A |
| Facilitated Workshops | Ayuda a las partes interesadas a discutir necesidades y funcionales y colabora con la comprensión mutua. | N/A |
| Prototipado | Un prototipo puede no ser completo ni probado respecto a todos sus requisitos funcionales y no funcionales afectados Considera cuatro categorías: -Negocios -Usabilidad -Rendimiento y Capacidad -Capacidad/Técnica | Sólo se presentan requisitos completados. El prototipo se presenta en la Demostración del Sprint Review. Se usa el Scrum Taskboard, que permite gestionar el <u>Product Backlog</u> Items comprobando las tareas completadas |
| Gestión de la Configuración del So-ftware – GCS | - Los miembros del equipo de DSDM deben poder tomar decisiones sin gastos administrativos innecesarios - Los métodos y herramientas de Configuration | N/A |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | |
|-------------------|--|--|
| | Managment no deben trabar el desarrollo - Todos los cambios que se produzcan en el desarrollo de un incremento son reversibles y toda liberación / prototipo se puede reproducir, al igual que los scripts de prueba y los resultados esperados, etc. | |
| Documentos | SDM Business Development Framework que completa DSDM Atern con sus Productos, creándolos: 1. Una única vez 2. Por períodos 3. Para cada realización, ya sea de uso externo o de uso interno 4. Si es necesario 5. Innecesarios | Scrum no provee una guía sobre documentación por lo que, en un proyecto, se le adjudican los documentos típicos de DSDM Atern Products. |

Tabla 13: Cuadro comparativo. Fuente: Elaboración propia

Consideraciones especiales sobre DSDM y Scrum

En el CAP.I se presentaron dos hipótesis, a saber:

1. DSDM es un enfoque que mantiene rigor metodológico y un control exhaustivo de un proyecto de software.
2. SCRUM obtiene resultados tempranos y a bajo coste, cumpliendo estándares de calidad.

Dimensiones de análisis

Ahora se tienen en cuenta dimensiones que contribuyan a profundizar en el proceso de análisis para luego extraer conclusiones provisionales.

- Registro exhaustivo.
- Control del proyecto.
- Calidad.
- Definición de roles.
- Coste.
- Gestión de riesgos.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Por lo expuesto, en el CAP.III, se puede inferir que:

El aporte de DSDM Atern en cuanto a resultados:

- a) DSDM aporta documentación, que permite el registro de la experiencia aplicable a futuros proyectos de la misma envergadura y que resulta una verdadera solución para empresas donde debe asegurarse firmemente el control de datos y riesgos, como por ejemplo bancos, servicios de inteligencia, organismos de gobierno, etc.

Además, desde el inicio de un proyecto, a través de los Productos de Factibilidad y la Fundación de Productos, para luego emplear Exploración y Productos de Ingeniería y finalmente Productos de Despliegue, DSDM mantiene el control del proyecto, aún después de la entrega en el post proyecto, ampliando con Productos del Post-Project para considerar las consecuencias en una Evaluación de los Beneficios, a posteriori de la implantación. (Fortuin Chris, DSDM Scrum Roadmap).

Por lo cual, se puede deducir que:

- b) DSDM abarca en su totalidad un proyecto, considerando entorno, tecnología, riesgos, calidad, de principio a fin.

Por otra parte, DSDM utiliza más roles que Scrum. Efectivamente, Scrum adolece del rol del Business Sponsor, y los roles de Business Visionary y Business Ambassador los desempeña en Scrum el Product Owner, que incluso se hace cargo de la responsabilidad del DSDM Project Manager. También el Scrum Master desempeña, conjuntamente, los roles de Project Manager, Workshop Facilitator y Atern Coach, propios de DSDM. Finalmente los roles de Team Leader, Business Advisor, Solution Developer y Solution Tester, de DSDM se reúnen en Scrum como Developer. (ver Fortuin Chris, DSDM Scrum Roadmap, en este CAP. IV). Esta diferencia incide obviamente en el coste.

La observación de Chris Fortuin es que al tener menos roles Scrum se facilita la comunicación (ver Fortuin Chris, DSDM Scrum Roadmap, en este CAP. IV), mientras que Matheu Caine (2012), considera que, por ejemplo, los Scrum de Scrums son más efectivos en base a los roles que incorpora DSDM Atern (ver CAP. II).

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Esto implica que:

- c) DSDM utiliza más roles que Scrum en la formación de su equipo, lo que incide en el aumento del coste, pero garantiza precisión en el desarrollo, implementación y mantenibilidad del proyecto.

Se agrega ahora que, la forma de trabajar de Scrum, con el uso de las Historias de usuario, los Burndown Chart, el Scrum Taskboard, los beneficios del Sprint Planning (mayor productividad, potenciación del compromiso asumido, aseguramiento de la confianza en una estimación del equipo en su conjunto) colaboran en la obtención temprana de resultados y a menor coste porque se desempeñan menos roles, como se dijo en el punto anterior.

Dice Pharro Richard, en el prefacio de *Agile Project Management* (2011:6) (ver CAP. II): “Scrum provee un excelente equipo con enfoque basado en permitir que el trabajo sea priorizado y entregado, usando el concepto de “backlog” de constante evolución para proporcionar la carga de trabajo del equipo. La fuerza de Scrum es su simplicidad, y puesto que es tan fácil de describir y empezar a usar, que esto ha impulsado su popularidad hasta la fecha”.

Pero Scrum padece de mayor riesgo en algunos proyectos, porque, tal como se vio en el CAP. II, como señala Wernham (2012), en DSDM la salida de cada iteración se llama la "solución emergente", mientras que en Scrum se denomina “incremento potencialmente liberable”.

En Scrum se habla de Product Backlog, en DSDM Prioritized Requirements List, pero cada elemento desarrollado, ya sea del Product Backlog o de la Prioritized Requirements List, requiere una definición de finalización cierta y precisa, para asegurar la calidad correcta.

En Scrum un incremento se establece como “Done” cuando se acuerda una “Definición de Hecho” para el equipo, que debe convenirse antes de que el equipo pueda planificar el desarrollo. Pero Scrum, en su Guía, no especifica cuándo se determina esta “Definición de Hecho”. Esto implica que los criterios de estricto sólo se deben acordar cuando agregan valor y no retrasan la solución. Para DSDM, en cambio, el trabajo de

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

definición se llevó a cabo durante la fase de Fundaciones del Proyecto. La guía de DSDM advierte que cualquier "definición de hecho" por adelantado al trabajo, durante la fase de Fundaciones, deberá revisarse periódicamente a lo largo del ciclo de vida del proyecto.

Por lo tanto:

- d) Scrum obtiene resultados tempranos a bajo coste, aunque con riesgo en algunos proyectos

Finalmente, se puede decir que:

- e) Scrum alcanza los estándares de calidad (ver Tabla 10 en este CAP. IV) porque:
 - Realiza mejora continua: en cada iteración el equipo realiza una retrospectiva
 - El equipo se comunica diariamente
 - El timeboxing colabora porque todas las actividades tienen la misma duración, por lo que la gente aprende a aprovechar ese tiempo: para organizarse, dar prioridades y asumir decisiones.
 - El equipo es multidisciplinar
 - En cada iteración el equipo estima el esfuerzo necesario para su concreción, asumiendo la responsabilidad que le cabe, en cuanto a tareas y completitud de requisitos, luego de una clara y definida autoasignación.
 - El equipo demuestra al cliente los resultados completados, y por ende operativos, al fin de cada iteración, manteniendo una velocidad de desarrollo sostenida, porque no revisa aspectos mal resueltos en iteraciones anteriores.

En definitiva, de a), b), c), d) y e) se pueden colegir las hipótesis 1. y 2., mencionadas en este capítulo (pág.141) y planteadas en el CAP.I.

Como conclusión provisional, se puede pensar que ambas metodologías pueden contribuir en forma conjunta al desarrollo de cualquier proyecto de software, inclinándose la decisión, en algunos casos, por el empleo de una sola de ellas.

Propuesta de Análisis Comparativo

Es ineludible destacar la simplicidad, ligereza de trabajo del enfoque Scrum y la prontitud en la entrega, que se denota todo tiempo en la forma de desarrollar la actividad del equipo, así como de las características de comunicación entre ellos y con el Scrum Master y el Product Owner. Por la autonomía del equipo, por la falta de burocracia, la sencillez y la continuidad de la comunicación, etc. Es ponderable también la realización de un BSC para medir el rendimiento estratégico.

Simultáneamente, es relevante la fortaleza de DSDM, la solidez de su labor, por los recaudos que toma realizando un Business Case, un Business Vission y un PAQ, para asegurarse, la organización, y asegurar al cliente de la conveniencia del desarrollo del proyecto. Es valorable la definición del Business Área Definition (BAD), del System Architecture Definition (SAD) y el Prototipar la Solución, elementos que no utiliza Scrum. Incluso son valiosos los roles que desempeña como es el caso del Ambassador User, también el Business Visionary y el Technical Coordinator del proyecto, mejorando la comunicación en los casos de Scrum de Scrums, y, sin duda, se destaca la explicitación de los Términos de Referencia.

Fueron analizadas las características esenciales de las dos metodologías, descubierto las coincidencias respecto de las ágiles, destacadas las que les son propias y marcadas las diferencias fundamentales en un cuadro comparativo.

Se tuvo en cuenta el aporte empírico de los consultores mencionados en el CAP. II, como también la colaboración de Chris Fortuin, en la toma de decisión a través del DSDM Scrum Roadmap (transcripto más arriba).

Y en este punto se agrega una opinión más:

“DSDM conserva muchas de las características de las metodologías tradicionales de gestión de proyectos, tales como los planes, los límites de la etapa, una tablero de proyectos y gerente de proyectos. Esto significa que las empresas que actualmente utilizan metodologías tales como PRINCE2 encontrarán en DSDM un reto menor de

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

adoptar. De hecho DSDM fue diseñado específicamente para que pueda ser utilizado junto con PRINCE2, si se desea.

Scrum, por otro lado, es un enfoque más ligero, que prescinde de gran parte de la documentación y la formalidad de las metodologías tradicionales de gestión de proyectos. Scrum ha demostrado ser un gran éxito en todo el mundo durante los últimos 20 años, pero puede ser más que un reto para las organizaciones tradicionales adoptarlo”. [] (2013, noviembre): *Agile...DSDM...SCRUM? Confused...Let us Help* <<http://www.quanta.co.uk/news/2013/11/agiledsdmscrum-confusedlet-us-help>>

Propuesta de Análisis Comparativo - Variables

Teniendo como meta resolver el problema planteado en el CAP.I consistente en determinar, entre DSDM y SCRUM, cuál metodología elegir para el desarrollo de un proyecto software, se considera lo siguiente:

Dados el requerimiento de un proyecto a desarrollar para una empresa específica, valorar la magnitud de la organización, para luego tener en cuenta la del proyecto en cuestión, y precisar, con alguna determinada calificación:

- Las características de la cultura organizacional, para recabar si habrá una buena disposición de colaboración por parte del cliente y/o usuarios, y
- los términos de referencia.

De acuerdo a estas valoraciones y calificaciones se determinará, para la organización y el desarrollo del proyecto requerido, en base al DSDM Scrum Roadmap y el Cuadro comparativo (Tabla 13), la conveniencia de utilizar Scrum o DSDM Atern o bien Scrum y DSDM Atern.

Cada uno de las 4 variables mencionadas (la organización, el proyecto, la cultura organizacional y los términos de referencia) estará representada por un vector de n coeficientes a los se les podría adjudicar una escala de valores, en función de las cuales se tomaría la decisión de cuál metodología emplear, DSDM o Scrum o ambas, para el desarrollo del proyecto requerido. Se aclara que a su vez, cada coeficiente podría ser un nuevo vector, cuando ocurran distintas alternativas por ítem y también distintos valores por alternativa.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Valores pasibles de ser considerados

A priori, se sugiere considerar, por ejemplo, los siguientes niveles o escalas (a analizar):

- 1- Sobre la magnitud de la organización (vector de 3 coeficientes):
 - a. PyME o Gran Empresa
 - b. Gubernamental o no
 - c. Reglamentada o no
- 2- Sobre la magnitud del proyecto (ver Tabla 11: Tipos de Proyectos, Cap. IV) (vector de 4 coeficientes):
 - a. Pequeño
 - b. Grande
 - c. Híbrido
 - d. Proyecto de cambio de negocio
- 3- Sobre la cultura organizacional:

Podría usarse el PAQ de DSDM Atern de 21 preguntas, que proporciona *DSDM Atern Hand Book* (2007) (ver ANEXO I. El vector tendría 21 coeficientes, variando cada uno dentro de una escala de 5 valores) y/o basarse en una encuesta que refiera a los Requisitos para poder utilizar Scrum, descritos en el CAP. III, a saber (vector de 5 coeficientes):

 - a. Cultura de empresa basada en trabajo en equipo, con características como: delegación, creatividad y mejora continua
 - b. Compromiso del cliente en la gestión de los resultados del proyecto y del ROI y de la disponibilidad para colaborar con el equipo en forma directa durante todo el desarrollo del proyecto
 - c. Compromiso de la dirección de la organización para formar equipos autogestionados y multidisciplinarios, fomentando una cultura de gestión basada en la colaboración y en la facilitación a cargo de los líderes del equipo.
 - d. Compromiso conjunto y colaboración de los miembros del equipo.
 - e. Relación entre proveedor y cliente basada en ganar-ganar, colaboración y transparencia....

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- 4- Sobre los términos de Referencia, se puede ver cuáles considerar de los siguientes y determinar, sobre los mismos, una escala de valores (vector de a lo sumo 17 coeficientes en caso de considerarse todos los términos acá expuestos):
- a. el ámbito espacial,
 - b. el objetivo general,
 - c. los objetivos específicos,
 - d. el tiempo disponible,
 - e. los requisitos de los técnicos que van a participar (especialidades y tiempos mínimos de dedicación)
 - f. los resultados esperados,
 - g. contenidos de los informes,
 - h. visión,
 - i. componentes,
 - j. roles,
 - k. responsabilidades,
 - l. recursos,
 - m. finanzas,
 - n. plan de calidad,
 - i. desglose del trabajo,
 - ii. calendarización y
 - iii. factores de éxito/riesgos y restricciones.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

CAPITULO V - DISEÑO CUALITATIVO

En este capítulo se trabaja con un diseño metodológico de tipo cualitativo y participativo, que permite recolectar, analizar, construir y concluir sobre datos cualitativos, obtenidos a través de instrumentos específicamente elaborados por la tesis para el desarrollo de la presente tesis, los cuales constituyen el ANEXO II.

La metodología permite explicar las razones de los diferentes aspectos de tal comportamiento. En otras palabras, investiga el por qué y el cómo se tomó una decisión, en contraste con la investigación cuantitativa, que busca responder preguntas tales como cuál, dónde, cuándo, cuánto. La misma, se basa en pequeñas muestras o casos de estudio, para ver qué sucede en la observación de grupos de población reducidos, como por ejemplo en este caso comunidad, universidad, empresa, equipo de trabajo, entre otras.

El método de análisis comparativo constante de Grounded Theory resulta ser el más apropiado para este estudio, ya que se definen dos hipótesis, dimensiones de análisis, las cuales permiten comparar ambas metodologías ágiles DSDM vs. SCRUM, luego gracias al modelo de pensamiento hipotético deductivo, se construye teoría y conclusiones.

Dos temas a investigar

En esta oportunidad, hay dos temas a investigar, que surgen de los planteos de los consultores citados en el CAP.II:

- DSDM es una metodología con “mayor fortaleza” que Scrum para el desarrollo de proyectos software
- Scrum es una metodología “más ágil” que DSDM en el sentido de obtener resultados más tempranos y a menor coste, aunque con más riesgos, en algunos proyectos. Pero que alcanza también a conformar los estándares de calidad porque logra, con un equipo más reducido en número que el de DSDM, desempeñar todos los roles necesarios para completar un desarrollo exitoso.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Dados estos interrogantes se realizó una muy acotada investigación de tipo cualitativa participativa:

- Cualitativa porque se recolectaron datos tratando de comprender la realidad tal como la experimentan sus protagonistas, del por qué y cómo tomaron una decisión. La investigación cualitativa se basa en observar en forma directa a grupos reducidos.
- Y participativa porque se remitió a un determinado campo, elegido por la tesista, con la participación de los sujetos investigados, y la pretensión de obtener cambios en la población para mejorar su labor. En este caso, la población es la comprendida por profesionales abocados, directa o indirectamente, al desarrollo de proyectos software.

Con este fin se formularon tres cuestionarios, sitios en el ANEXO II.

Respuestas a los cuestionarios del ANEXO II

El ANEXO II se compone de tres cuestionarios, dirigidos a Profesionales Dedicados al Desarrollo de Proyectos Software, a Evaluadores de Calidad de Empresas Desarrolladoras de Software y a Ingenieros en Informática con Conocimiento, aunque teórico, en metodologías Scrum y DSDM

Se transcriben en la página siguiente las respuestas de los profesionales encuestados, habiéndose consultado a tres por cada área.

Las respuestas se vuelcan en tres tablas, una por cada cuestionario, donde los números indican la cantidad de respuestas afirmativas por cada respuesta opcional y en la columna Observaciones se describen las aclaraciones correspondientes.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Respuestas al Cuestionario I

| | Pregunta | Respuestas | Observaciones |
|---|---|--|--|
| 1 | ¿Considera usted que su experiencia es mayor en proyectos software para grandes empresas o en pymes? | <ul style="list-style-type: none"> Grandes empresas: Pymes: | <i>Los 5 profesionales respondieron que su experiencia es igual en grandes empresas que en PyMES</i> |
| 2 | ¿Diría usted que su experiencia con los métodos se centra principalmente en el desarrollo de proyectos o de productos software o ambos en partes iguales? | <ul style="list-style-type: none"> Proyectos: 2 Productos: 1 Ambos: | |
| 3 | ¿Diría usted que su experiencia con el desarrollo de proyectos y/o productos software viene principalmente a través de la aplicación práctica, los estudios académicos, la investigación o de todos los puntos considerados? | <ul style="list-style-type: none"> La aplicación práctica: Los estudios académicos: La investigación: Todos los puntos considerados: 3 | |
| 4 | Con respecto a DSDM ¿tiene usted alguna certificación formal? | <ul style="list-style-type: none"> Si: No: 3 | <ul style="list-style-type: none"> Agile Advance Practitioner: DSDM Advanced Practitioner: Agile Foundation: DSDM Coach: DSDM Foundation: DSDM Trainer: |
| 5 | Independientemente de la certificación, por favor, con el uso de DSDM, indique el número de años de experiencia (si los tiene contabilizados) en esta metodología, la cantidad de proyectos en los que intervino, la duración del proyecto de mayor envergadura y su rol en ese proyecto: | <ul style="list-style-type: none"> Número de años de experiencia: --- Cantidad de proyectos en los que intervino: --- | <ul style="list-style-type: none"> Duración del proyecto de mayor envergadura (se consideró el mayor entre todos): --- Rol desempeñado en ese proyecto (se consideró el rol del proyecto de punto anterior): --- |
| 6 | Con respecto a SCRUM, ¿tiene usted alguna certificación formal? | <ul style="list-style-type: none"> Si: 1 No: 2 | <ul style="list-style-type: none"> C. Scrum Master (CSM): C. Scrum Product Owner (CSPO): --- C. Scrum Developer (CSD): 1 C. Scrum Professional (CSP): C. Scrum Coach (CSC): |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | | |
|---|---|--|--|
| | | | <ul style="list-style-type: none"> • C. Scrum Trainer (CST): |
| 7 | Con respecto a SCRUM, por favor indique el número de años de experiencia (si los tiene contabilizados) en esta metodología, la cantidad de proyectos en los que intervino, la duración del proyecto de mayor envergadura y su rol en este proyecto: | <ul style="list-style-type: none"> • Años de experiencia (se consideró el nro. mayor): 3 • Cantidad de proyectos (se consideró el nro. mayor): 2 | <ul style="list-style-type: none"> • Duración del proyecto de mayor envergadura (se consideró el mayor entre todos): <i>1 año</i> • Rol desempeñado en ese proyecto (se consideró el rol del proyecto de punto anterior): <i>SCD</i> |
| 8 | Por favor explique acerca de cualquier otra gestión formal de proyecto o de calificaciones de desarrollo de proyecto software o experiencia que usted pueda tener: | <p>Lo que sigue es la respuesta de cada uno de los 3 desarrolladores de proyectos software:</p> <ol style="list-style-type: none"> 1. <i>Capacitación a empresas en TDD y Scrum</i> 2. <i>Empleo del Proceso Unificado de Desarrollo para grandes proyecto. Es recomendable usarlo porque establece claramente las fases, las actividades de cada fase, los roles para realizar cada actividad y los artefactos para documentar las tareas realizadas.</i> 3. <i>Uso de modelo en cascada para pequeños proyectos, principalmente para gestionar el mantenimiento de aplicaciones.</i> | |

Tabla 14: Respuestas al Cuestionario I para Desarrolladores de Proyectos Software

Fuente: Elaboración propia

Respuestas al Cuestionario II

| | Pregunta | Respuestas | Observaciones |
|---|--|--|--|
| 1 | ¿Considera usted que su experiencia como evaluador de calidad de empresas desarrolladoras de proyectos software es mayor en grandes empresas o en pymes? | <ul style="list-style-type: none"> • Grandes empresas: • Pymes: | <i>Los 3 profesionales respondieron que su experiencia es igual en grandes empresas que en PyMES</i> |
| 2 | ¿Dada su experiencia como evaluador de calidad de empresas desarrolladoras de proyectos software, diría usted que las que utilizan metodologías tradicionales se centran más en el proyecto o en el producto software o en ambos por partes iguales? | <ul style="list-style-type: none"> • Proyectos: • Productos: • Ambos: 3 | |
| 3 | ¿Dada su experiencia como evaluador de empresas desarrolladoras de proyectos software, diría usted que las que utilizan metodologías ágiles se centran más en el proyecto o en | <ul style="list-style-type: none"> • Proyectos: • Productos: 3 • Ambos: | |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | | |
|----|---|---|---|
| | el producto software o en ambos por partes iguales? | | |
| 4 | ¿Diría usted que, de acuerdo a su experiencia como evaluador, en Argentina es mayor la cantidad de empresas desarrolladoras que utilizan metodologías tradicionales que las que utilizan metodologías ágiles? | <ul style="list-style-type: none"> • Tradicionales: • Agiles: 3 | |
| 5 | Aproximadamente, del total de empresas desarrolladoras de proyectos software que usted haya evaluado en Argentina en los dos últimos años, en qué porcentaje utilizan metodología ágil? | <ul style="list-style-type: none"> • En más de un 20%: • En más de un 50%: 3 • El 100 %: | |
| 6 | Aproximadamente, del total de empresas desarrolladoras de proyectos software que usted haya evaluado en países extranjeros en los dos últimos años, en qué porcentaje utilizan metodología ágil? | <ul style="list-style-type: none"> • En más de un 20%: • En más de un 50%: 3 • El 100 %: | Países : <ul style="list-style-type: none"> - Chile - Bolivia |
| 7 | De las empresas desarrolladoras de proyectos software que utilizan Metodologías Ágiles y que usted haya evaluado, ¿es mayor la cantidad que emplea Scrum que DSDM? | <ul style="list-style-type: none"> • Si: 3 • No: | Porcentaje estimativo (el mayor de los porcentaje que resultó de las respuestas afirmativas):100% |
| 8 | Con respecto a DSDM, ¿ha encontrado alguna “No conformidad “al evaluar en calidad una empresa desarrolladora que utilice esta metodología? | <ul style="list-style-type: none"> • Si: --- • No: --- | “No conformidad “ (se enumeran los profesionales): <ol style="list-style-type: none"> 1. --- 2. --- 3. --- |
| 9 | Con respecto a Scrum ¿ha encontrado alguna “No conformidad “ al evaluar en calidad una empresa desarrolladora que utilice esta metodología? | <ul style="list-style-type: none"> • Si: 1 • No:2 | “No conformidad “ (se enumeran los profesionales): <ol style="list-style-type: none"> 1. <i>Generalmente utilizan SCRUM no completamente como plantea el método sino con algunas características propias como por ejemplo que el usuario no participa en su rol de Product Owner.</i> 2. --- 3. --- |
| 10 | Si ha evaluado alguna empresa desarrolladora de proyectos software que utilice la metodología DSDM y de considerarlo apropiado, indique, por favor, las apreciaciones que estime convenientes sobre la utilización de esta metodología en comparación con las tradicionales en cuanto a documentación, roles, actividades, costes y plazo de entrega (se enumeran los profesionales): | | |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | |
|-----------|--|
| | <ol style="list-style-type: none"> 1. --- 2. --- 3. --- <p>y en comparación con Scrum (se enumeran los profesionales):</p> <ol style="list-style-type: none"> 1. --- 2. --- 3. --- |
| 11 | <p>Si ha evaluado alguna empresa desarrolladora de proyectos software que utilice la metodología Scrum y de considerarlo apropiado, indique, por favor, las apreciaciones que estime convenientes sobre la utilización de esta metodología en comparación con las tradicionales (se enumeran los profesionales):</p> <ol style="list-style-type: none"> 1. <i>Produce artefactos ejecutables de software orientados a la solución más rápidamente utilizables por el usuario.</i> 2. .. 3. ... <p>y en comparación con DSDM (se enumeran los profesionales):</p> <ol style="list-style-type: none"> 1. <i>Este método (DSDM) no es utilizado en el mercado latinoamericano</i> 2. (Idem) 3. (Idem) |

Tabla 15: Respuestas al Cuestionario II para Evaluadores de Calidad de Empresas Desarrolladoras de Proyectos Software. Fuente: Elaboración propia

Respuestas al Cuestionario III

| | Pregunta | Respuestas | Observaciones |
|----------|---|--|---------------|
| 1 | ¿Considera ud. que DSDM aporta a la empresa suficiente documentación como para aplicarla en proyectos software futuros de la misma envergadura? [marque con x la que corresponda]: | <ul style="list-style-type: none"> • Si: 3 • No: • No Sabe: | |
| 2 | ¿Considera ud. que DSDM aporta la documentación necesaria para empresas donde debe asegurarse firmemente el control de datos y riesgos, como por ejemplo bancos, servicios de inteligencia, organismos de gobierno, etc.? | <ul style="list-style-type: none"> • Si: 3 • No: • No Sabe: | |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | | |
|----|--|--|--|
| 3 | <p>¿Piensa ud. que la cantidad de documentación que genera DSDM es excesiva?</p> <p>Si la respuesta es afirmativa: ¿cree ud. que esta situación aleja tal vez a DSDM del concepto “ágil” de las Metodologías Agiles?</p> | <ul style="list-style-type: none"> • Si: 1 • No: 2 • No Sabe: <ul style="list-style-type: none"> • Si: 1 • No: 2 • No Sabe: | |
| 4 | ¿Considera ud. que Scrum aporta a la empresa suficiente documentación como para aplicarla en proyectos futuros de la misma envergadura? | <ul style="list-style-type: none"> • Si: 2 • No: 1 • No Sabe: | |
| 5 | ¿Considera ud. que Scrum aporta la documentación necesaria para empresas donde debe asegurarse firmemente el control de datos y riesgos, como por ejemplo bancos, servicios de inteligencia, organismos de gobierno, etc.? | <ul style="list-style-type: none"> • Si: 1 • No: 2 • No Sabe: | |
| 6 | <p>¿Piensa ud. que la cantidad de documentación que genera Scrum es excesiva?</p> <p>Si la respuesta es afirmativa: ¿cree ud. que esta situación aleja tal vez a Scrum del concepto “ágil” de las Metodologías Agiles?</p> | <ul style="list-style-type: none"> • Si: • No: 3 • No Sabe: <ul style="list-style-type: none"> • Si: • No: • No Sabe: | |
| 7 | ¿Considera ud. necesario realizar un Business Case para asegurar que el empleo de una u otra metodología o ambas es lo apropiado para un proyecto dado? | <ul style="list-style-type: none"> • Si: 3 • No: • No Sabe: | |
| 8 | ¿Considera ud. necesario realizar un Business Vision para asegurar que el empleo de una u otra metodología o ambas es lo apropiado para un proyecto dado? | <ul style="list-style-type: none"> • Si: 3 • No: • No Sabe: | |
| 9 | Considera ud. que la cantidad de roles que se desempeñan en DSDM es excesiva? | <ul style="list-style-type: none"> • Si: 1 • No: 2 • No Sabe: | |
| 10 | Considera ud. que la cantidad de roles que se desempeñan en Scrum es escasa? | <ul style="list-style-type: none"> • Si: • No: 3 • No Sabe: | |
| 11 | <p>¿Considera ud. que DSDM aventaja a Scrum como metodología ágil?</p> <p>Por favor, cualquiera haya sido su respuesta, explique en pocas palabras el por qué:</p> | <ul style="list-style-type: none"> • Si: • No: 2 • No Sabe: 1 | <p>Si porque (se enumeran los profesionales):</p> <p>1-</p> <p>2-</p> <p>3-</p> |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | | |
|----|---|--|---|
| | | | <p>No porque (se enumeran los profesionales):</p> <p>1-<i>DSDM fue precursora de las metodologías ágiles y es más sólida que Scrum pero Scrum acorta plazos y requiere menos costes</i></p> <p>2- <i>Las 2 se complementan</i></p> <p>3-</p> |
| 12 | <p>¿Considera ud. que Scrum aventaja a DSDM como metodología ágil?</p> <p>Por favor, cualquiera haya sido su respuesta, explique en pocas palabras el por qué:</p> | <ul style="list-style-type: none"> • Si: • No: 3 • No Sabe: | <p>Si porque (se enumeran los profesionales):</p> <p>1-</p> <p>2-</p> <p>3-</p> <p>No porque (se enumeran los profesionales):</p> <p>1- <i>Las 2 se complementan. No creo que pueda considerarse una mejor que la otra</i></p> <p>2- <i>Scrum es aparentemente más ágil pero DSDM se funda de una manera más sólida.</i></p> <p>3-<i>En cuanto a costes y tiempo ambas pueden adecuarse a las necesidades de la empresa.</i></p> |
| 13 | <p>¿Le parece conveniente la instalación de una Sede de DSDM Consortium en Argentina?</p> <p>Por favor, cualquiera haya sido su respuesta, explique en pocas palabras el por qué:</p> | <ul style="list-style-type: none"> • Si: 3 • No: • No Sabe: | <p>Si porque (se enumeran los profesionales):</p> <p>1- <i>Resultaría bastante costoso lograrlo pero facilitaría a posteriori la asistencia a workshops y conferencias que podrían realizarse en Argentina.</i></p> <p>2- <i>Ayudaría a la capacitación en DSDM, la que podría usarse en proyectos de gran envergadura o para organizaciones como bancos, prestadoras de servicios de la salud, etc.</i></p> <p>3- <i>Son interesantes los roles en DSDM y a veces muy</i></p> |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | | | |
|----|---|--|---|
| | | | <p><i>necesarios. DSDM tiene un concepto más amplio y abarcativo en cuanto a negocios.</i></p> <p>No porque (se enumeran los profesionales):</p> <p>1-</p> <p>2-</p> <p>3-</p> |
| 14 | Si es que le parece conveniente la instalación de una Sede de DSDM Consortium en Argentina, ¿le resultaría interesante participar de las actividades que en la misma se desarrollen? | <ul style="list-style-type: none"> • Si: 3 • No: • No Sabe: | |
| 15 | <p>¿Le interesaría obtener alguna certificación para implementar la metodología DSDM?</p> <p>Si su respuesta fue afirmativa explique, por favor, en pocas palabras qué nivel de certificación quisiera obtener:</p> | <ul style="list-style-type: none"> • Si: • No: • No Sabe: 3 | <p>Nivel de certificación a obtener (se enumeran los profesionales):</p> <p>1-</p> <p>2-</p> <p>3-</p> |

Tabla 16: Respuestas al Cuestionario III para Ingenieros en Informática con conocimiento teórico de las Metodologías DSDM y Scrum. Fuente: Elaboración propia

En Argentina y América en general, excepto EEUU, donde recién se está iniciando su utilización, no se emplea DSDM como metodología de desarrollo de proyectos software.

Teniendo conocimiento previo de esta situación, por información recibida de DSDM Consortium y comentario general de profesionales locales, dedicados al desarrollo de proyectos de software, se realizó el Cuestionario III, para tener algunas consideraciones, aunque basados en conocimientos teóricos, de destacados profesionales en informática de la Argentina.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Algunas conclusiones extractadas de las respuestas

- Los desarrolladores de proyectos de software y los evaluadores de calidad en empresas desarrolladoras de proyectos software tienen igual experiencia en grandes empresas que en PyMES.
- Ninguno de ellos, desarrolladores y evaluadores, tiene experiencia en aplicación de la Metodología DSDM
- Según los evaluadores más de un 50% de las empresas desarrolladoras de proyectos de software utiliza metodologías ágiles. Agregan que lo mismo ocurre en países como Chile y Bolivia.
- Los ingenieros en informática con conocimiento teórico de DSDM coincidieron en la realización de un Business Case y un Business Vision para asegurar el empleo de una u otra metodología o ambas para un proyecto de software.
- Al compararlas, en alguna medida acuerdan que ninguna es mejor que la otra. Consideran: que se complementan, que Scrum es más ágil pero DSDM más sólida y, en cuanto a costes, que ambas pueden adecuarse.

CAPITULO VI - CONCLUSIONES Y FUTUROS TRABAJOS

En este capítulo se plasman las conclusiones a las que se pudo arribar por este trabajo y finalmente se proponen investigaciones a futuro, sobre los valores que podrían eventualmente tomar las variables definidas en la Propuesta de Análisis Comparativo, expresado en el CAP.IV, como también investigar sobre las diferencias señaladas en el CAP.V, entre DSDM y Scrum, en cuanto a “mayor fortaleza de DSDM respecto de Scrum” y “mayor agilidad” de Scrum respecto de DSDM”, ampliando el campo de estudio.

Conclusiones

A lo largo de la presente tesis, se recrearon las características básicas de las metodologías ágiles de desarrollo de software y se las comparó con las tradicionales, sosteniendo los beneficios y ventajas de aplicación de las primeras.

En el CAP.II se hizo un breve reseña histórica para señalar el surgimiento de las Metodologías Ágiles, se preentó el Manifiesto Ágil y los 12 principios en los que se basan. A través de un cuadro comparativo se las confrontó con las tradicionales para, luego, destacar sus características primordiales y los beneficios que aportan. Entonces se introdujeron las metodologías Scrum y DSDM, haciendo un relato en cuanto a antecedentes, creadores, publicaciones, certificadores, proyectos en que fueron empleadas. Respecto de DSDM se mencionaron sus versiones pasadas, actuales y en proceso, destacándose DSDM Atern, brevemente descrita en el ANEXO I, sobre la que se basó DSDM Agile Project Framework, creado para completar otros enfoques.

Luego se resaltó la diferencia entre Proyecto y Proceso de Software para después señalar los puntos imprescindibles para el desarrollo de un proyecto de software, destacando beneficiosas características de DSDM para esta tarea. Y a continuación, se hizo referencia a comentarios de distintos consultores internacionales sobre la aplicación de uno u otro o ambos enfoques, en función de resultados con relación a escalabilidad del producto, controles sobre la realización de “Scrum de Scrums”, ante

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

grandes proyectos, las respuestas correctas a “qué” y “cuándo” (se conseguirá) y “cuánto” (va a costar).

En el CAP.III se describieron las características básicas de estas dos metodologías, fundamentos, roles, artefactos, técnicas, métricas, beneficios, restricciones, riesgos y se completó presentando las bases de DSDM Atern, sus 8 principios y sus productos.

En el CAP.IV se planteó la Propuesta de Análisis Comparativo. Primero se presentaron los aspectos fundamentales de Scrum, roles y skills necesarios y potenciación del equipo, la planificación del proyecto y de la iteración, las métricas y el uso del cuadro de mandos integral, BSC, y los beneficios de Scrum.

Luego, se mostraron las características destacadas de DSDM, la flexibilidad de sus requisitos, la gente y el equipo, los workshops y sus beneficios, se enumeraron las técnicas básicas y se detallaron los elementos de que dispone para decidir cuándo usar DSDM.

De The DSDM Agile Project Framework for Scrum, formulado a partir de The DSDM Agile Project Framework y creado desde DSDM Atern para complementar otros enfoques ágiles, en particular Scrum, se resumieron: el lifecycle, los roles, los artefactos y las técnicas

Se presentó el llamado DSDM Roadmap, definido por Fortuin, y se adicionaron las aclaraciones pertinentes.

Haciendo un resumen de las características de ambas metodologías se destacó el cumplimiento de las dos sobre los principios ágiles y se enumeraron las particulares de cada una de ellas.

Se expuso entonces, basándose en todo lo analizado en el presente trabajo, un cuadro comparativo que pone de manifiesto las diferencias entre las mismas, que las resume, pero que no las enfrenta, las distingue, y en ciertos casos las complementa.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

A partir de realizar un análisis profundo por medio de las dimensiones definidas: registro exhaustivo, control del proyecto, calidad, definición de roles, coste y gestión de riesgos; se pudo inferir que se han comprobado las hipótesis propuestas al inicio de la presente tesis de maestría:

1. DSDM es un enfoque que mantiene rigor metodológico y un control exhaustivo de un proyecto de software.
2. SCRUM obtiene resultados tempranos y a bajo coste, cumpliendo estándares de calidad.

Pudo apreciarse como muy clara la rigurosidad metodológica del enfoque DSDM como aparentemente “más ágil” y “divertida” la manera de trabajar de Scrum.

Se atendió a los comentarios de los consultores internacionales, que han experimentado con cada una de ellas y emiten sus opiniones en base a resultados obtenidos y estudios realizados. Estos, en su mayoría, destacan la necesidad de la aplicación de DSDM, por la calidad, por los riesgos, por la mantenibilidad, por la escalabilidad, porque permite obtener respuestas a qué, cuándo y cuánto, reconociendo, algunos de ellos, la prontitud en la entrega y el menor coste de Scrum respecto de DSDM.

En base a todos los datos recabados, se presentó una Propuesta de Análisis Comparativo, con cuatro variables a considerar: la organización, el proyecto, la cultura organizacional y los términos de referencia, para solucionar la disquisición sobre qué metodología elegir, entre DSDM y Scrum, ante un proyecto software para una determinada organización. Y se ejemplificaron valores posibles a se tomados por las variables.

Finalmente, se encuestaron a profesionales dedicados directa o indirectamente al desarrollo de proyectos software, aunque no se logró contactar desarrolladores de proyectos que utilizaran DSDM. La pregunta esencial encerrada en estos cuestionarios es: Como metodologías: DSDM, es de “mayor fortaleza” que Scrum? Scrum, es “más ágil” que DSDM?. Abriéndose un posible tema de investigación.

Futuros trabajos

De la presente tesis se desprende la necesidad de seguir investigando en esta línea de trabajo sobre aplicación de metodologías ágiles:

- Análisis e investigación de posibles valores a tomar por las cuatro variables determinadas en la Propuesta de Análisis Comparativo: la organización, el proyecto, la cultura organizacional y los términos de referencia. De tal forma y, una vez determinada cada variable como un vector de n coeficientes, estudiar los valores o escalas sobre los cuales se desplazarían estos coeficientes, deduciendo o definiendo resultados en cada caso, en el sentido de cuál metodología emplear, ya sea Scrum, DSDM o ambas.
- Analizar sistemáticamente, si DSDM es una metodología acorde a desarrollo de software para grandes empresas vs. SCRUM para PyMES.

CAPITULO VII - BIBLIOGRAFIA

- Albaladejo Xavier (2009, diciembre): *Características de un Equipo Agile*
<<http://www.proyectosagiles.org/skills-equipo-agil>> [Consulta: febrero de 2014]
- Arce Rioboó José Carlos: *La Diferencia entre un Proyecto y un Proceso*,
<<http://www.keisen.com/article62.html>> [Consulta: febrero de 2014]
- Baetjer, Jr. H. (1998): *Software as Capital*, Washington, D.C: IEEE Computer Society Press. Pág. 85.
- Caine Matheu (2011, agosto): *What is DSDM Atern?*
<<http://www.mcpa.biz/2011/08/what-is-dsdm-atern/>>
[Consulta: diciembre de 2013]
- Caine Matheu (2011, setiembre): *Agile Project Management*
<<http://www.mcpa.biz/2011/09/agile-project-management/>>
[Consulta: diciembre de 2013]
- Caine Matheu (2012, marzo): *Scaling Agile Project.*
<<http://www.mcpa.biz/2012/03/scaling-agile-projects/>>
[Consulta: diciembre de 2013]
- Caine Matheu (2012, mayo): *Mapping documents to DSDM Atern*
<<http://www.mcpa.biz/2012/05/mapping-documents-to-dsdm-atern/>>
[Consulta: diciembre de 2013]
- Caine Matheu (2013, marzo): *Scaling Agile Projects.*
<<http://www.mcpa.biz/2012/03/scalingagile-projects/>>
[Consulta: diciembre de 2013]
- Canós José H., Letelier Patricio y Penadés M^a Carmen (2003): *Metodologías Ágiles en el Desarrollo de Software – DSIC - Universidad Politécnica de Valencia*,
<<http://issi.dsic.upv.es/archives/f1069167248521/actas.pdf>>
[Consulta: diciembre de 2013]
- Carnegie Dale (1998, octubre): *How To Win Friends And Influence People*. Nueva York: Pocket Books- Simon & Schuster

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Craddock Andrew, Roberts Barbara, Richards Keith, Godwin Julia y Tudor Dorothy, bajo la revisión de Casal-Giménez José, Clare Nick, Ramseyer Mischa (2007): *The DSDM Agile Project Framework for Scrum*.
<<http://www.dsdm.org/sites/default/files/The%20DSDM%20Agile%20Project%20Framework%20v1%2011.pdf>> [Consulta: octubre de 2013]
- Foote Brian, Harrison Neil y Rohnert Hans (1999, diciembre): *PLoPD 4*. (Software Patterns Series). Londres: Addison Wesley Publishing Company
- Fortuin Chris (2010): *DSDM Scrum Roadmap*.
<<http://orangefortune.com/DSDMScrumRoadmap.html>>
[Consulta: diciembre de 2013]
- Fowler Martin (2003, abril): *La nueva metodología*
<http://www.willydev.net/Willydev_old/Root/descargas/prev/LasNuevasMetologiasPDF> [Consulta: diciembre de 2013]
- Greaves Karen (2010, 4 de febrero): *Coaching Teams to do Better Scrum*
<<http://scrumcoaching.wordpress.com/2010/02/04/undone-work/>>
[Consulta: marzo de 2014]
- Hill Nicola Hildred (2012, mayo): *Agile Business Continuity – What is it?*
<<http://abithon.wordpress.com/author/nicolahildredhill/>>
[Consulta: diciembre de 2013]
- Jacobson Ivar, Booch Grady, Rumbaugh James (1999): *The Unified Software Development Process*. Reading, Massachusetts: Addison-Wesley Professional
- Kerth Norman L (2001, febrero) *Project Retrospectives: A Handbook for Team Reviews* Londres: Dorset House
- (c) Kerth Norman L. (2001): *An Approach to Postmorta, Postparta & Post Project Reviews* <<http://c2.com/doc/ppm.pdf>> [Consulta: diciembre de 2013]
- Mills Amy (2013, noviembre): *Agile...DSDM...SCRUM? Confused...Let us Help*
<<http://www.quanta.co.uk/news/2013/11/agiledsdmscrum-confusedlet-us-help>>
[Consulta: mayo de 2014]
- Palacio Juan (2007, octubre): *Flexibilidad con Scrum*
<http://www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf>
[Consulta: mayo de 2014]
- Pressman Roger S. (2006): *Ingeniería de Software*. México, D.F.: Mc Graw-Hill Interamericana. Pág. 80

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Rasmusson Jonathan (2012): *The Agile Samurai: How Agile Masters Deliver Great Software*. Colección: Pragmatic Programmers. Londres: Pragmatic Bookshelf, Susannah Davidson Pfalser
- Robinson Richard (2010, mayo): *Why prefer DSDM to Scrum?*
<<http://hamesrobinson.blogspot.com.ar/2010/05/why-prefer-dsdm-to-scrum.html>> [Consulta: diciembre de 2013]
- Schwaber, K., & Sutherland, J. (1996). *Controlled Chaos: Living on the Edge*. San José, California: OOPSLA.
- Schwaber Ken y Beedle Mike (2001, 21 de octubre): *Agil Software Development with Scrum*, (Series en Agil Software Development). Londres: Prentice Hall - Pearson Education
- Schwaber Ken, Sutherland Jeff (2012, abril): *Software in 30 days*. Hoboken, NJ: Wiley John & Sons
- Sols Rodríguez-Candela, Alberto; Fernández Fernández, Isabel; Romero Yacobi, Javier (2013): *Gestión integral de proyectos*. Madrid: Universidad Pontificia Comillas
- Sommerville Ian (2005): *Ingeniería de Software*. Madrid: Pearson Education
- Stapleton J. (1997): *DSDM Dynamic Systems Development Method: The Method in Practice*. Londres: Addison-Wesley.
- Strauss, A. (1987): *Qualitative analysis for social scientists*. Cambridge, England: Cambridge University Press.
- Sutherland, Jeff (2001): *Agile Can Scale: Inventing and Reinventing SCRUM in Five Companies*. <<http://www.controlchaos.com/storage/scrum-articles/Sutherland%20200111%20proof.pdf>> [Consulta: diciembre de 2013]
- Sutherland, Jeff, Schwaber Ken (2012, 23 de marzo): *Software de Treinta Días*. East Anglia: John Wiley & Sons, Inc.
- Sutherland, Jeff (2013): *Scrum of Scrums*.
<<http://guide.agilealliance.org/guide/scrumofscrums.html>,>
[Consulta: diciembre de 2013]
- Taylor, S. J. y Bodgdan, R. (1980): *Introducción a los métodos cualitativos de investigación*. Barcelona: Paidós.
- Wernham Brian (2012, 31 de Julio): *Agile Project Management for Government*. London: Maitland and Strong
- [] (2001): *Manifiesto por el Desarrollo Ágil de Software*. <<http://agilemanifesto.org/iso/es/>>
[Consulta: diciembre de 2013]

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- [] (2002): *Why DSDM?* <www.dsdm.org> [Consulta: diciembre de 2003]
- [] (2003): *Información General sobre Escalabilidad*
<[http://msdn.microsoft.com/es-es/library/aa292203\(v=vs.71\).aspx](http://msdn.microsoft.com/es-es/library/aa292203(v=vs.71).aspx)>
[Consulta: diciembre de 2013]
- []: *DSDM Tour*. <www.dsdm.org> [Consulta: 2003]
- [] (2007): *DSDM Atern* <<http://dsdm.org/dig-deeper/book/dsdm-atern-handbook>> [Consulta: enero de 2014]
- [] (2009, marzo): *Ingeniería del software: metodologías y ciclos de vida*. Madrid: Laboratorio Nacional de Calidad del Software de INTECO
<https://www.inteco.es/file/N85W1ZWfHifRgUc_oY8_Xg ingeniería del software metodologías y ciclos de vida > [Consulta: diciembre de 2013]
- [] (2010): *Lista de objetivos / requisitos priorizada (Product Backlog)*
<<http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog>>
[Consulta: febrero de 2014]
- [] (2010): *Gráficos de trabajo pendiente (Burndown charts)*
<<http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts>>
[Consulta: febrero de 2014]
- [] (2010): *Planificación de la iteración (Sprint Planning)*
<<http://www.proyectosagiles.org/planificacion-iteracion-sprint-planning>>
[Consulta: febrero de 2014]
- [] (2010): *Ejemplo de uso del tablero o pizarra de tareas (Scrum Taskboard)*
<<http://www.proyectosagiles.org/ejemplo-tablero-pizarra-tareas-scrum-taskboard>> [Consulta: febrero de 2014]
- [] (2010): *Objetivos como historias de usuario*
<<http://www.proyectosagiles.org/introduccion-estimacion-planificacion-agil#historias-usuario>> [Consulta: febrero de 2014]
- [] (2010): *Beneficios de Scrum* <<http://www.proyectosagiles.org/beneficios-de-scrum#expectativas>> [Consulta: febrero de 2014]
- [] (2011, febrero): *Agile Project Management*. (white paper) APMG-International, DSDM Consortium <<http://www.apmg-international.com/>>
[Consulta: noviembre de 2013]
- [] (2011, enero): *DSDM is Agile's Best Kept Secret?*.
<<http://agilescout.com/dsdm-is-agiles-best-kept-secret/>>
[Consulta: diciembre de 2013]

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- [] (2012, 15 de mayo): *Scrum & DSDM Atern: AND, not XOR*
<<http://abithon.wordpress.com/2012/05/15/scrum-dsdm-atern-and-not-xor>>
[Consulta: diciembre 2013]
- [] (2012, julio): *Metodologías para el desarrollo de software*, Unidad VI. Asignatura: 071-4323. Núcleo de Monagas Maturín, Monagas, Venezuela: Universidad de Oriente<http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_para_el_desarrollo_de_software> [Consulta: enero de 2014]
- [] (2012): *Agile Project Management*
<<http://www.pmpprogress.com/certificaciones/agile/>>
[Consulta: noviembre de 2013]
- [] (2012): *Qué es Scrum* <<http://www.proyectosagiles.org/que-es-scrum>>
[Consulta: diciembre de 2013].
- [] (2013): *El Ciclo de Vida de los Requisitos en un Proyecto Ágil* <www.dsdm.org>
[Consulta: abril de 2013]
- [] (2013, noviembre): *Agile...DSDM...SCRUM? Confused...Let us He*
<<http://www.quanta.co.uk/news/2013/11/agiledsdmscrum-confusedlet-us-help>>
[Consulta: mayo de 2014]
- [] (2014, febrero): *Agile Business Analysts, AgileBA*, <www.dsdm.org>
[Consulta: abril de 2014]
- []: *Historia de Scrum*. <<http://www.proyectosagiles.org/historia-de-scrum>>
[Consulta: diciembre de 2013].
- []: *Dynamic System Development Method (DSDM)* <<http://www.philadelphia.edu.jo/>>
[Consulta: abril de 2014]
- []: *1471-2000 - IEEE Recommended Practice for Architectural Description for Software-Intensive Systems* [Consulta: abril de 2014]

Enlaces de interés

- DSDM Consortium <www.dsdm.org>
- Agile Alliance <www.agilealliance.org>
- Scrum.org <www.scrum.org>
- Instituto Nacional de Tecnología Industrial - INTI
- <www.inti.gob.ar/capacitacion/software/scrum.htm>
- Cámara de Empresas de Software y Servicios Informáticos de la República

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- Argentina - CESSI <www.cessi.org.ar>
- Instituto Argentino de Normalización y Certificación - IRAM <www.iram.org.ar>
- Proyectos ágiles.org <www.proyectosagiles.org>
- APMG-International <www.apmg-international.com>
- Project Management Institute - PMI <www.pmi.org>
- Yellow House <www.yellowhouse.net>

ANEXO I - DSDM ATERN

En este Anexo se presentan algunas características de DSDM Atern

Flexibilidad de DSDM Atern

Atern se puede usar como suplemento de otras disciplinas de gestión de proyectos, como por ejemplo PRINCE2™, eXtreme Programming y SCRUM, facilitando el marco de gestión ágil necesario que favorezca la entrega comprobada de proyectos ágiles.

Casi todos los proyectos tiene 4 parámetros: tiempo, costo, características y calidad.

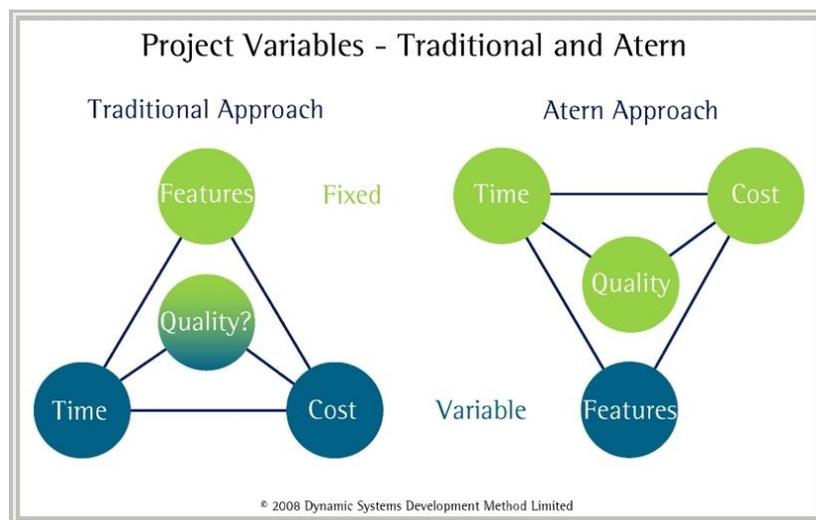


Fig. 34: Variables de un proyecto

Fuente: *What is DSDM Atern* <<http://www.mcpa.biz/2011/08/what-is-dsdm-atern/>>/

En el diagrama de la izquierda, enfoque tradicional, se ve que tiempo y costo se consideran variable. Por lo tanto, si hay desvío del proyecto puede que se agreguen más recursos o se postergue la fecha de entrega, lo que dañaría la calidad y la confianza.

Pero fijando tiempo, costo y calidad, en la fase de Fundaciones, los imprevistos se controlan a través de las funciones a ser entregadas: las de menor prioridad se quitan o posponen, de acuerdo a las reglas de MoSCoW. La calidad se fija porque se acuerdan criterios mínimos de aceptación antes de iniciar el desarrollo, el que se desenvolverá de manera controlada y con claros objetivos.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

En tanto se respeten las reglas de MoSCoW y Timeboxing, un subconjunto mínimo de funciones será entregado a tiempo y dentro del budget.

Los requisitos de alto nivel están bien definidos, por lo tanto se puede detectar la diferencia entre la necesidad de lograr detalles adicionales (relativos, por ejemplo, a la comprensión de las necesidades) y adiciones a la amplitud del proyecto (scope creep: corrupción del alcance)

Apropiados niveles de rigor

Dado un proyecto particular, Atern tiene que determinar el nivel de rigor a utilizar. Cada proyecto es distinto. Mucha formalidad podría atrasar el progreso e incluso paralizarlo. Pero tampoco es buena muy poca formalidad. Es importante lograr un equilibrio que permita gestionar los proyectos Atern.

Para determinar el nivel de rigor que debe aplicarse en todo el proyecto, desde el inicio se hace una evaluación del riesgo. El concepto es tener una formalidad considerable: se eliminan desechos y todas las actividades, en todo nivel incremental, añaden valor, de forma tal de ser formal y riguroso sin estorbar la evolución.

Ciclo de Vida del Proyecto

Atern abarca todo el ciclo de vida del proyecto, lo que lo distingue de los métodos ágiles más comunes, abarca mucho más que el desarrollo de software.

Utiliza disciplinas de gestión de proyectos y mecanismos que podrán asegurar la claridad de los beneficios del proyecto, que la solución planteada es viable y las bases lo suficientemente sólidas. Todo proyecto Atern consta de siete fases.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

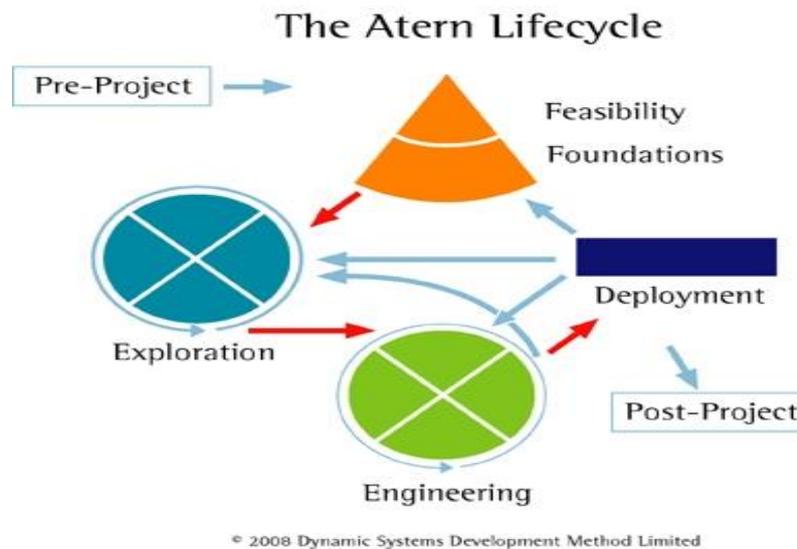


Fig. 35: Ciclo de vida del proyecto

Fuente: *DSDM Atern Project Structure* <<http://www.mcpa.biz/2011/10/dsdm-atern-project-structure-overview/>>

Fases del Lifecycle Atern

| Fase | Descripción |
|---------------|--|
| Pre-proyecto | Se acuerdan los términos de referencia para el trabajo |
| Factibilidad | Se evalúa brevemente la viabilidad del proyecto y se esboza el Business Case (Caso de Negocio) con justificación |
| Cimientos | Fase clave que asegura que el proyecto es bien entendido y bastante bien definido antes de comenzar la actividad |
| Exploración | Fase de desarrollo iterativo durante la cual el equipo se expande sobre los requisitos de alto nivel para demostrar la funcionalidad |
| Ingeniería | Fase de desarrollo Iterativo donde la solución se ha diseñado ingenierilmente para poder desplegarse para el lanzamiento |
| Despliegue | Por cada incremento (conjunto de timeboxes) del proyecto, la solución esté disponible. |
| Post-proyecto | Evalúa los beneficios acumulados |

Tabla 17: Fases de un proyecto

Fuente: *DSDM Atern Stucture* <<http://www.mcpa.biz/2011/10/dsdm-atern-project-structure-overview/>>

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

En general, las fases de exploración e ingeniería se fusionan para una mayor adaptación a la situación.

Ocho principios de Atern

Atern se basa en ocho principios, originados en las Best Practices.

| Icono | Fase | Descripción |
|---|--|---|
|  | Centrarse en las necesidades del negocio | Entregar lo que necesita el negocio. Las verdaderas prioridades de la empresa deben ser entendidas con un Business Case sólido. |
|  | Estar al día | Los timeboxes se planifican con antelación dentro del plazo establecido. Las fechas no se modifican, las características varían en función de las prioridades del negocio, con el fin de alcanzar la fecha límite. |
|  | Colaborar | Los equipos trabajan con espíritu de cooperación y compromiso activo. La colaboración promueve la comprensión, la velocidad y la propiedad compartida. Los equipos deben estar suficientemente autorizados e incluyen los representantes de las empresas. |
|  | Nunca renunciar a la calidad | El nivel de calidad se establece desde el principio. Los proyectos deben probarse tempranamente y en forma continua y revisados constantemente. |
|  | Construir de forma incremental | Los incrementos permiten a la empresa aprovechar el trabajo antes de que el producto final se haya completado, fomentar la confianza de las partes interesadas y el feedback. Esto se basa en hacer el suficiente el análisis inicial para continuar y aceptar cada detalle que pueda surgir más tarde. |
|  | Desarrollar Iterativamente | Aceptar que el trabajo no es siempre el correcto al principio. Utilizar timeboxes para permitir el cambio pero confirmar continuamente que la solución es la correcta. |
|  | Comunicar continuamente y en forma clara | Utilizar talleres apoyados por facilitadores, standups diarios, modelos, prototipos, presentaciones y fomentar la comunicación cara a cara. |
|  | Demostrar control | El equipo necesita ser proactivo en el seguimiento y control de los avances en línea con la Fase de Fundamentos. Tienen que evaluar constantemente la viabilidad del proyecto en base a los objetivos de negocio. |

Tabla 18: Principios de Atern

Fuente: *DSDM Atern Principals* <<http://www.mcpa.biz/2011/10/dsdm-atern-principals-overview/>>

Estos principios determinan la forma de trabajar. El marco completo se deduce de los mismos. El incumplimiento de cualquiera de ellos puede llevar indefectiblemente al fracaso.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Roles y responsabilidades de Atern

El siguiente diagrama, denominado "Baby Alien", es el diagrama standart Atern DSDM que ilustra los tres tipos de roles: del proyecto, del desarrollo de la solución y otros.

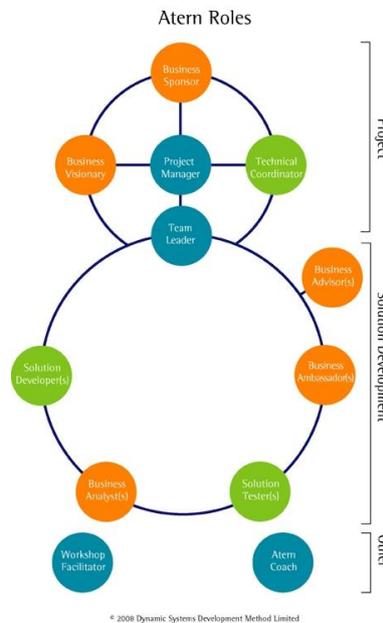


Fig. 36: Roles y responsabilidades de Atern. Fuente: *DSDM Atern Project Structure*
 <<http://www.mcpa.biz/2011/10/dsdm-atern-project-structure-overview/>>

Roles del proyecto

| Rol | Principales responsabilidades |
|------------------------------|--|
| Business Sponsor | Posee el Business Case. Garantiza la financiación y la dotación de recursos. Garantiza una eficaz toma de decisiones y se ocupa de escaladas rápidamente. |
| Project Manager | Planificación de alto nivel. Supervisa el progreso, la disponibilidad de recursos, la configuración del proyecto, gestiona asuntos de riesgo y el escalamiento. |
| Business Visionary | Posee la visión empresarial y el impacto de los cambios comerciales más amplios. Monitorea el progreso. Contribuye para requisitos clave, sesiones de diseño y revisión. |
| Technical Coordinator | Acuerda y controla la arquitectura técnica. Asesora y coordina equipos. Identifica y gestiona el riesgo técnico. Se asegura de que se cumplan los requisitos no funcionales (fiabilidad, tiempo de respuesta, capacidad de almacenamiento, restricciones del sistema). |

Tabla 19: Roles del Proyecto

Fuente: *DSDM Atern Roles and Responsibilities* <<http://www.mcpa.biz/2011/10/dsdm-atern-roles-and-responsibilities-an-overview/>>

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Roles de desarrollo de soluciones

| Rol | Principales responsabilidades |
|----------------------------|--|
| Team Leader | Centra al equipo en la entrega a tiempo. Alienta la participación del equipo completo. Administra actividades de timeboxes detallados y las del día a día. Asegura que las actividades de prueba y revisión estén programadas y completadas. |
| Business Ambassador | Contribuye a todas las sesiones de requisitos, diseño y revisión. Proporciona el punto de vista de negocios para todas las tomas de decisiones del día a día. Describe escenarios de negocio para ayudar a diseñar y probar la solución. Proporciona seguridad de que la solución es correcta. Coordina la aceptación comercial. |
| Solution Developer | Crea la solución y participa plenamente en todas las actividades de control de calidad adecuadas. |
| Tester Solution | Trabaja con los roles de negocio para definir escenarios de prueba para la solución. Realiza todos los informes técnicos de los resultados de pruebas al Jefe de Equipo y al Coordinador Técnico. |
| Business Analyst | Apoya la comunicación entre empresa y técnicos integrantes del equipo. Gestiona todos los productos necesarios relacionados con los requerimientos del negocio. Asegura que las consecuencias empresariales de decisiones del día a día son correctamente pensadas. |
| Business Advisor | Proporciona aporte de especialistas, por ejemplo, un contador o un asesor de impuestos. Por lo general, un usuario previsto de la solución |

Tabla 20: Roles del desarrollo de soluciones

Fuente: *DSDM Atern Roles and Responsibilities* <<http://www.mcpa.biz/2011/10/dsdm-atern-roles-and-responsibilities-an-overview/>>

Otros roles

| Rol | Principales responsabilidades |
|-----------------------------|--|
| Atern Coach | Ayuda a los equipos nuevos para equipos Atern a sacar el máximo provecho de Atern. Adapta Atern para las necesidades del proyecto. |
| Workshop Facilitator | Gestiona y organiza talleres. Responsable por el contexto, no por el contenido. Independiente. |
| Other Specialists | Los expertos necesarios en un corto plazo, posiblemente técnico, por ejemplo especialistas en la carga de prueba, etc. |

Tabla 21: Otros roles

Fuente: *DSDM Atern Roles and Responsibilities* <<http://www.mcpa.biz/2011/10/dsdm-atern-roles-and-responsibilities-an-overview/>>

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Atern Products o Productos de Atern

Los Atern Products son las entregas y están asociadas con cada fase del ciclo de vida.

Tal como se dijo en el Cap. III, con mayor detalle, no todos los productos son necesarios en cada proyecto, además varía su formalidad, en función del proyecto y la organización.

No pueden eludirse factores contractuales y standards corporativos.

Se clasifican, de acuerdo al momento de su creación, en:

Por una única vez:

- Business Case en detalle
- Testing Strategy o Estrategia de prueba

Antes y después de la Arquitectura de Alto Nivel

- Por períodos:
 - Delivery Plan o Plan de Entrega, que incluye el alcance de alto nivel
 - Financial Reports o Informes Financieros
- Para cada realización externa:
 - Guía del usuario
 - Guías de Instalación y Actualización
 - Notas de la versión
 - Guía de funcionamiento
- Para cada realización interna:
 - Datos de Prueba
 - Casos de Prueba
 - Automatización de Pruebas (Unit & Negocios)
 - Resultados de la prueba
- Si es necesario:
 - Pantalla y el informe de diseño Diseños
- Innecesarios

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Project Approach Questionnaire - PAQ

Es un cuestionario con escala likert, que se utiliza para gestionar los eventuales riesgos que deban abordarse.

Generalmente se completa durante la fase de Feasibility y se reevalúa en las Foundations.

Consiste de 21 preguntas dirigidas a participantes clave del proyecto y stakeholders, cuyas respuestas revisan el Project Manager y el Business Sponsor.

Constituye el Appendix B del DSDM Atern (2007:139), que publica, como e-book, DSDM Consortium en <www.dsdm.org>.

Las preguntas son las siguientes:

1. “El business driver está claramente detrás y es visible para todos los miembros del equipo del proyecto”
2. “El Business Sponsor y/o Business Visionary demuestran claramente la propiedad del proyecto”
3. “El Business Sponsor, el Business Visionary y el Technical Co-ordinator todos entienden y aceptan la filosofía de Atern”
4. “El desarrollo cuenta con un calendario claramente definido”
5. “Los requisitos pueden ser priorizados y hay flexibilidad para aceptar que no todos los requisitos son ‘Must Have’ requirements”
6. “Los requisitos han sido definidos en un alto nivel en el inicio del proyecto y se reconoce que los cambios son probables durante el desarrollo de los detalles”.
7. “Se acepta que el detalle tanto de los requisitos como de la solución surgirá como el proyecto avance”
8. “El Business Sponsor y el Business Visionary son conscientes de la importancia de la **participación activa implicación empresarial** y tienen la **voluntad y la autoridad para comprometer recursos empresariales adecuados para el proyecto según sea necesario**”

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

9. “Los Business Ambassadors están suficientemente facultados para guiar el día a día de la evolución de la solución”
10. “Los Solution Developers están suficientemente facultados para proporcionar la mejor solución que se pueda de una perspectiva de negocio dentro de las limitaciones de arquitectura pre-acordados”.
11. “Los recursos para el desarrollo de la solución son asignados a un nivel adecuado y el equipo será en gran parte estable durante todo el proyecto (o, al menos, a lo largo de cada incremento)”
12. “El Equipo del Proyecto y el Equipo de Desarrollo de la Solución está capacitados para adoptar los roles y responsabilidades dentro de Atern”
13. “Será posible para los Desarrolladores de la Solución tener fácil acceso a los Business Ambassadors y a los Business Advisors en todo el proyecto”
14. “El equipo de Desarrollo de la Solución (incluyendo recursos de negocio y de desarrollo de la solución) tendrá el conocimiento colectivo apropiado y / o habilidades técnicas para entregar la solución”
15. “El equipo de Desarrollo de la Solución (incluyendo recursos de negocio y de desarrollo de la solución) tendrá las habilidades sociales adecuadas (comunicación, negociación, etc) para trabajar de manera efectiva unos con otros y aquellos a su alrededor”
16. “Estrategias para la comunicación continua y las prácticas trabajo colaborativo son suficientes para apoyar claramente el desarrollo iterativo de la solución”
17. “Las herramientas y técnicas de desarrollo tecnológico apoyan un enfoque iterativo para el desarrollo de soluciones”
18. “No existen limitaciones técnicas, contractuales o de índole para impedir que la solución esté siendo dividida en incrementos para el desarrollo y la entrega (incluso si los productos de tales incrementos no se implementan de inmediato)”
19. “Todos los participantes en el proyecto comprenden y aceptan que la entrega a tiempo de una solución aceptable es la principal medida de éxito para el proyecto”
20. “Todas las partes aceptan que la evaluación continua de la idoneidad para el propósito de todas las entregas durante el desarrollo es esencial”
21. “No hay normas obligatorias o prácticas vigentes que funcionen en contra de la evolución de una solución a partir de un conjunto en línea con la base de los requisitos de alto nivel”.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

ANEXO II – CUESTIONARIOS I, II y III

Cuestionario I: Dirigido a un Desarrollador de Proyectos Software

Muchas gracias por su importante colaboración y la tan amable disposición al responder este breve cuestionario. Su trayectoria y experiencia profesional, sin duda enriquecerá mi tesis de maestría en informática.

Lic. Hebe Carlota Anadón

Cuestionario Metodologías Ágiles [DSDM / SCRUM] – Abril de 2014-

1. ¿Considera usted que su experiencia es mayor en proyectos software para grandes empresas o en pymes? [marque con x la que corresponda]:

| | |
|------------------|--|
| Grandes empresas | |
| Pymes | |

2. ¿Diría usted que su experiencia con los métodos se centra principalmente en el desarrollo de proyectos o de productos software o de ambos en partes iguales? [marque con x la que corresponda]:

| | |
|--------------------|--|
| Proyectos Software | |
| Productos Software | |
| Ambos | |

3. ¿Diría usted que su experiencia con el desarrollo de proyectos y/o productos software viene principalmente a través de la aplicación práctica, los estudios académicos, la investigación o de todos los puntos considerados? [marque con x la que corresponda]:

| | |
|----------------------------------|--|
| La aplicación práctica | |
| Los estudios académicos | |
| La investigación | |
| De todos los puntos considerados | |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

4. Con respecto a DSDM ¿tiene usted alguna certificación formal? [marque con x la que corresponda]:

| | |
|----|--|
| Si | |
| No | |

En caso afirmativo indique por favor si usted es [marque con x la que corresponda]:

| | |
|----------------------------------|--|
| Profesional certificado DSDM | |
| Certificado DSDM Project Manager | |
| Certificado DSCM Trainer | |
| Consultor DSDM certificado | |
| Full Certificate | |

5. Independientemente de la certificación, por favor, con el uso de DSDM, indique el número de años de experiencia (si los tiene contabilizados) en esta metodología, la cantidad de proyectos en los que intervino, la duración del proyecto de mayor envergadura y su rol en ese proyecto:

| | |
|--|--|
| Número de años de experiencia | |
| Cantidad de proyectos en los que intervino | |
| Duración del proyecto de mayor envergadura | |
| Su rol en ese proyecto | |

6. Con respecto a SCRUM, ¿tiene usted alguna certificación formal? [marque con x la que corresponda]:

| | |
|----|--|
| Si | |
| No | |

En caso afirmativo indique por favor si usted es [marque con x la que corresponda]:

| | |
|--|--|
| Certified Scrum Master –CSM | |
| Certified Scrum Product Owner- CSPO | |
| Certified Scrum Developer-CSD | |
| Certified Scrum Professional-CSP | |
| Certified Scrum Coach-CSC | |
| Certified Scrum Trainer-CST | |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

7. Con respecto a SCRUM, por favor indique el número de años de experiencia (si los tiene contabilizados) en esta metodología, la cantidad de proyectos en los que intervino, la duración del proyecto de mayor envergadura y su rol en este proyecto:

| | |
|--|--|
| Número de años de experiencia | |
| Cantidad de proyectos en los que intervino | |
| Duración del proyecto de mayor envergadura | |
| Su rol en ese proyecto | |

8. Por favor explique acerca de cualquier otra gestión formal de proyecto o de calificaciones de desarrollo de proyecto software o experiencia que usted pueda tener.

.....

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Cuestionario II: Dirigido a un Evaluador de Proyectos Software

Muchas gracias por su importante colaboración y la tan amable disposición al responder este breve cuestionario. Su trayectoria y experiencia profesional, sin duda enriquecerá mi tesis de maestría en informática.

Lic. Hebe Carlota Anadón

Cuestionario Metodologías Ágiles [DSDM / SCRUM] - Abril de 2014

1. ¿Considera usted que su experiencia como evaluador de calidad de empresas desarrolladoras de proyectos software es mayor en grandes empresas o en pymes? [marque con x la que corresponda]:

| | |
|------------------|--------------------------|
| Grandes empresas | <input type="checkbox"/> |
| Pymes | <input type="checkbox"/> |

2. ¿Dada su experiencia como evaluador de calidad de empresas desarrolladoras de proyectos software, diría usted que las que utilizan metodologías tradicionales se centran más en el proyecto o en el producto software o en ambos por partes iguales? [marque con x la que corresponda]:

| | |
|-------------------|--------------------------|
| Proyecto Software | <input type="checkbox"/> |
| Producto Software | <input type="checkbox"/> |
| Ambos | <input type="checkbox"/> |

3. ¿Dada su experiencia como evaluador de empresas desarrolladoras de proyectos software, diría usted que las que utilizan metodologías ágiles se centran más en el proyecto o en el producto software o en ambos por partes iguales? [marque con x la que corresponda]:

| | |
|-------------------|--------------------------|
| Proyecto Software | <input type="checkbox"/> |
| Producto Software | <input type="checkbox"/> |
| Ambos | <input type="checkbox"/> |

4. ¿Diría usted que, de acuerdo a su experiencia como evaluador, en Argentina es mayor la cantidad de empresas desarrolladoras que utilizan metodologías tradicionales que las que utilizan metodologías ágiles? [marque con x la que corresponda]:

| | |
|---------------|--------------------------|
| Tradicionales | <input type="checkbox"/> |
| Agiles | <input type="checkbox"/> |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

5. Aproximadamente, del total de empresas desarrolladoras de proyectos software que usted haya evaluado en Argentina en los dos últimos años, en qué porcentaje utilizan metodología ágil?[marque con x la que corresponda]:

| | |
|------------------|--------------------------|
| En más de un 20% | <input type="checkbox"/> |
| En más de un 50% | <input type="checkbox"/> |
| El 100 % | <input type="checkbox"/> |

6. Aproximadamente, del total de empresas desarrolladoras de proyectos software que usted haya evaluado en países extranjeros en los dos últimos años, en qué porcentaje utilizan metodología ágil?[indique, por favor, cuáles son los países referidos y marque con x la que corresponda]:

Países:.....

| | |
|------------------|--------------------------|
| En más de un 20% | <input type="checkbox"/> |
| En más de un 50% | <input type="checkbox"/> |
| El 100 % | <input type="checkbox"/> |

7. De las empresas desarrolladoras de proyectos software que utilizan Metodologías Agiles y que usted haya evaluado, ¿es mayor la cantidad que emplea Scrum que DSDM?[marque con x la que corresponda]:

| | |
|----|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |

Si su respuesta es afirmativa, por favor, indique en qué porcentaje estimativo:
.....

8. Con respecto a DSDM, ¿ha encontrado alguna observación al evaluar en calidad una empresa desarrolladora que utilice esta metodología?[marque con x la que corresponda]:

| | |
|----|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |

En caso afirmativo, por favor, indique cuál/es:
.....
.....

9. Con respecto a Scrum ¿ha encontrado alguna observación al evaluar en calidad una empresa desarrolladora que utilice esta metodología?[marque con x la que corresponda]:

| | |
|----|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

En caso afirmativo, por favor, indique cuál/es:

.....
.....

10. Si ha evaluado alguna empresa desarrolladora de proyectos software que utilice la metodología DSDM y de considerarlo apropiado, indique, por favor, las apreciaciones que estime convenientes sobre la utilización de esta metodología en comparación con las tradicionales en cuanto a documentación, roles, actividades, costes y plazo de entrega:

.....
.....

y en comparación con Scrum:

.....
.....

11. Si ha evaluado alguna empresa desarrolladora de proyectos software que utilice la metodología Scrum y de considerarlo apropiado, indique, por favor, las apreciaciones que estime convenientes sobre la utilización de esta metodología en comparación con las tradicionales en cuanto a documentación, roles, actividades, costes y plazo de entrega:

.....

y en comparación con DSDM:

.....
.....

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Cuestionario III: Dirigido a un Ingeniero Informático con conocimiento teórico de las Metodologías DSDM y Scrum

Muchas gracias por su importante colaboración y la tan amable disposición al responder este breve cuestionario. Su trayectoria y experiencia profesional, sin duda enriquecerá mi tesis de maestría en informática.

Lic. Hebe Carlota Anadón

Cuestionario Metodologías Ágiles [DSDM / SCRUM] - Mayo de 2014

Entendiendo que ud conoce al menos en teoría las metodologías DSDM y Scrum se le solicita responda a las siguientes preguntas:

1. ¿Considera ud. que DSDM aporta a la empresa suficiente documentación como para aplicarla en proyectos software futuros de la misma envergadura? [marque con x la que corresponda]:

| | |
|----------|--|
| Si | |
| No | |
| No Sabe: | |

2. ¿Considera ud. que DSDM aporta la documentación necesaria para empresas donde debe asegurarse firmemente el control de datos y riesgos, como por ejemplo bancos, servicios de inteligencia, organismos de gobierno, etc.? [marque con x la que corresponda]:

| | |
|----------|--|
| Si | |
| No | |
| No Sabe: | |

3. ¿Piensa ud. que la cantidad de documentación que genera DSDM es excesiva? [marque con x la que corresponda]:

| | |
|----------|--|
| Si | |
| No | |
| No Sabe: | |

Si la respuesta es afirmativa: ¿cree ud. que esta situación aleja tal vez a DSDM del concepto “ágil” de las Metodologías Ágiles? [marque con x la que corresponda]:

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| | |
|----------|--|
| Si | |
| No | |
| No Sabe: | |

4. ¿Considera ud. que Scrum aporta a la empresa suficiente documentación como para aplicarla en proyectos futuros de la misma envergadura?[marque con x la que corresponda]:

| | |
|----------|--|
| Si | |
| No | |
| No Sabe: | |

5. ¿Considera ud. que Scrum aporta la documentación necesaria para empresas donde debe asegurarse firmemente el control de datos y riesgos, como por ejemplo bancos, servicios de inteligencia, organismos de gobierno, etc.?[marque con x la que corresponda]:

| | |
|----------|--|
| Si | |
| No | |
| No Sabe: | |

6. ¿Piensa ud. que la cantidad de documentación que genera Scrum es excesiva? [marque con x la que corresponda]:

| | |
|----------|--|
| Si | |
| No | |
| No Sabe: | |

Si la respuesta es afirmativa: ¿cree ud. que esta situación aleja tal vez a Scrum del concepto “ágil” de las Metodologías Ágiles? [marque con x la que corresponda]:

| | |
|----------|--|
| Si | |
| No | |
| No Sabe: | |

7. ¿Considera ud. necesario realizar un Business Case para asegurar que el empleo de una u otra metodología o ambas es lo apropiado para un proyecto dado?[marque con x la que corresponda]:

| | |
|----------|--|
| Si | |
| No | |
| No Sabe: | |

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

8. ¿Considera ud. necesario realizar un Business Vision para asegurar que el empleo de una u otra metodología o ambas es lo apropiado para un proyecto dado? [marque con x la que corresponda]:

| | |
|----------|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |
| No Sabe: | <input type="checkbox"/> |

9. ¿Considera ud. que la cantidad de roles que se desempeñan en DSDM es excesiva?[marque con x la que corresponda]:

| | |
|----------|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |
| No Sabe: | <input type="checkbox"/> |

10. ¿Considera ud. que la cantidad de roles que se desempeñan en Scrum es escasa? [marque con x la que corresponda]:

| | |
|----------|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |
| No Sabe: | <input type="checkbox"/> |

11. ¿Considera ud. que DSDM aventaja a Scrum como metodología ágil? [marque con x la que corresponda]:

| | |
|----------|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |
| No Sabe: | <input type="checkbox"/> |

Por favor, cualquiera haya sido su respuesta, explique en pocas palabras el por qué:

.....
.....

12. ¿Considera ud. que Scrum aventaja a DSDM como metodología ágil? [marque con x la que corresponda]:

| | |
|----------|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |
| No Sabe: | <input type="checkbox"/> |

Por favor, cualquiera haya sido su respuesta, explique en pocas palabras el por qué:

.....
.....

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

13. ¿Le parece conveniente la instalación de una Sede de DSDM Consortium en Argentina? [marque con x la que corresponda]:

| | |
|----------|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |
| No Sabe: | <input type="checkbox"/> |

Por favor, cualquiera haya sido su respuesta, explique en pocas palabras el por qué:

.....
.....

14. Si es que le parece conveniente la instalación de una Sede de DSDM Consortium en Argentina, ¿ le resultaría interesante participar de las actividades que en la misma se desarrollen? [marque con x la que corresponda]:

| | |
|----------|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |
| No Sabe: | <input type="checkbox"/> |

15. ¿Le interesaría obtener alguna certificación para implementar la metodología DSDM? [marque con x la que corresponda]:

| | |
|----------|--------------------------|
| Si | <input type="checkbox"/> |
| No | <input type="checkbox"/> |
| No Sabe: | <input type="checkbox"/> |

Si su respuesta fue afirmativa explique, por favor, en pocas palabras qué nivel de certificación quisiera obtener:

.....
.....

ANEXO III - EMPRESA DESARROLLADORA DE SOFTWARE (Confidencial)

Introducción

Este documento explica y detalla el procedimiento de desarrollo de proyectos basado en metodologías ágiles. Para este tipo de proyectos la Empresa se basará en Scrum que es un método de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Objetivos / Campo de aplicación

- Definir la metodología de desarrollo.
- Definir los mecanismos de control y seguimiento.

Precondiciones / Frecuencia

- Requerimiento de desarrollo como salida del procedimiento de evaluación.
- Incidencia no resuelta por el proceso de soporte.

Definiciones

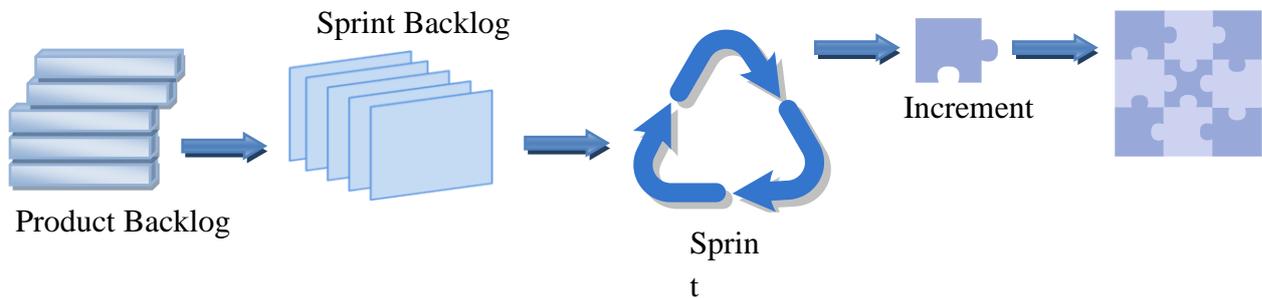
Scrum

Es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado en entornos de desarrollo ágil de software.

- Permite la auto-organización del equipo.
- Maximiza la capacidad del equipo para concretar entregas rápidas y responder a requisitos no planificados.
- Utiliza time-boxing que es una técnica de administración de tiempo que consiste en dividir el trabajo en tareas con una asignación de tiempo limitado y corto para su ejecución.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

El proceso y sus elementos



Cada sprint está constituido por un conjunto de elementos que vienen del *Product Backlog*. Durante la reunión de *Sprint Planning* se determinan los elementos del *Product Backlog* que forman parte del sprint. El equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el desarrollo del sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint. Durante el período de desarrollo de cada *sprint*, el equipo crea un incremento de software *entregable*.

- **Roles:** Propietario del producto, Gestor o Manager del Scrum, Equipo e Interesados.
- **Componentes del proceso:** Pila del producto (Product Backlog), Pila del sprint (Sprint Backlog), Incremento.
- **Reuniones:** Planificación del sprint, Revisión diaria, Revisión del sprint.

Sprint

El Sprint es el período en el cual se lleva a cabo el trabajo en sí. Es recomendado que la duración de los sprints sea constante y definida por el equipo en base a su propia experiencia. Al final de cada sprint, el equipo deberá presentar los avances logrados, y deberían entregar un producto con características de utilizable por el cliente. Asimismo se recomienda no cambiar los objetivos del sprint o sprint backlog a menos que la falta de estos cambios amenacen al éxito del proyecto.

Roles

Product Owner: representa la voz del cliente, a los stakeholders (interesados externos o internos). Se asegura de que el equipo Scrum trabaja de forma adecuada desde la

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

perspectiva del negocio. El Product Owner define las características del producto, las prioriza, las coloca en el Product Backlog y acepta o rechaza resultados del trabajo.

ScrumMaster (o Facilitador): El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El Scrum Master no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El Scrum Master se asegura de que el proceso Scrum se utiliza como es debido. El Scrum Master es el que hace que las reglas se cumplan. Trabaja de forma similar al director de proyecto. Gestiona y facilita la ejecución del proyecto.

Scrum Team o Equipo: El equipo tiene la responsabilidad de construir y entregar el producto, debe contar con las habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc).

Reuniones en Scrum

Daily Scrum: cada día de un sprint, se realiza la reunión sobre el estado de un proyecto.

- La reunión comienza puntualmente a su hora.
- La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.
- La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.
- Durante la reunión, cada miembro del equipo contesta a tres preguntas:
 - ¿Qué has hecho desde ayer?
 - ¿Qué es lo que estás planeando hacer hoy?
 - ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

Scrum de Scrum: estas reuniones permiten a los grupos de equipos discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración.

- Cada día normalmente después del “Daily Scrum”.
- Asiste una persona asignada por cada equipo.
- La agenda será la misma que la del Daily Scrum, añadiendo además las siguientes cuatro preguntas:
 - ¿Qué ha hecho tu equipo desde nuestra última reunión?

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

- ¿Qué hará tu equipo antes que nos volvamos a reunir?
- ¿Hay algo que demora o estorba a tu equipo?
- ¿Estás a punto de poner algo en el camino del otro equipo?

Reunión de Planificación del Sprint (Sprint Planning Meeting): al inicio del ciclo Sprint una “Reunión de Planificación del Sprint” se lleva a cabo.

- Seleccionar qué trabajo se hará.
- Preparar, con el equipo completo, el Sprint Backlog detalla el tiempo que tomará hacer el trabajo.
- Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint.

Reunión de Revisión del Sprint (Sprint Review Meeting):

- Revisar el trabajo que fue completado y no completado
- Presentar el trabajo completado a los interesados (alias “demo”)
- El trabajo incompleto no puede ser demostrado
- Cuatro horas como límite

Retrospectiva del Sprint (Sprint Retrospective): después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

Documentos

Product backlog: es una lista de requisitos de alto nivel priorizados que definen el trabajo a realizar. Contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables, etc. priorizadas. Es el qué va a ser construido. Es abierto y cualquiera puede modificarlo. Contiene estimaciones a grosso modo del esfuerzo de desarrollo requerido. Esta estimación ayuda ajustar la prioridad de las diferentes tareas.

Sprint backlog: es una lista detallada donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horas.

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

Burn down: es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, se visualiza el progreso del proyecto. Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el Backlog). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.

Participantes

Líder de Proyecto (Scrum Master)

Analista Funcional (Scrum Team)

Desarrollador (Scrum Team)

Maquetador (Scrum Team)

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

ANEXO IV - TABLA DE TESIS SOBRE METODOLOGIAS AGILES

Para un proyecto software: Metodología Ágil: DSDM vs. SCRUM

| Universidad | Tesista | Título de la Tesis | Año | Abstract |
|--|------------------------------|--|------------|--|
| Fac. informática – Univ.Politécnica de Madrid-España | Pilar Rodríguez González | Tesis de Maestría: Estudio de la Aplicación de metodologías Agiles para la Evolucion de Productos Software | 2008 | El objeto de esta investigación es estudiar la evolución de un producto software concreto utilizando las directrices marcadas por metodologías ágiles, en concreto por la metodología SCRUM. Se presentan los resultados obtenidos en aspectos tales como las características del producto a lo largo de la evolución, incluyendo estimaciones de la calidad del producto obtenido, la agilidad en el desarrollo, y evaluando el esfuerzo dedicado a adoptar la metodología. Además, dado que el factor humano es fundamental en este tipo de metodologías, se presenta un análisis cualitativo del desarrollo del proyecto. |
| Fac. informática – Univ.Politécnica de Madrid-España | José Germán Nuñez Mori | Tesis de Maestría: Usabilidad en Metodologías Agiles | 2010 | ...el presente trabajo tiene como objetivo principal, la inclusión de los principios de usabilidad, en proyectos Software, que se desarrollen en base a metodologías ágiles. |
| Fac. Cs.Físicas y Matemáticas – Depto. Cs. De la Computación - Univ. Chile-Chile | Agustín Antonio Villena Moya | Tesis de Maestría: Un Modelo Empírico de Enseñanza de las metodologías Agiles: El Caso del Curso CC62V – Ytaller de metodologías Agiles de Desarrollo de Software | 2008 | El objetivo de esta investigación por ende es comprobar mediante un análisis acucioso de un semestre del curso CC62V estos buenos resultados observados de manera informal, buscando además determinar las claves que explican los resultados del modelo pedagógico aplicado en el curso, y el real impacto logrado por éste en los aprendizajes de los alumnos que cursan dicho ramo. |
| UPC – Barcelona – España | Jose Carlos Carvajal Riola | Tesis de Maestría: Metodologías Agiles: Herramientas y Modelo de desarrollo para Aplicaciones Java EE como Metodología Empresarial | 2008 | |

Para un proyecto software: Metodología Ágil DSDM vs. SCRUM

| Universidad | Tesista | Título de la Tesis | Año | Abstract |
|--|-------------------------------|--|------|---|
| FRBA-UTN-Argentina | Mitaritonna, Alejandro Daniel | Tesis de Maestría: CAPA-3: Una innovadora metodología para el desarrollo de software en ambientes de trabajo virtuales | 2010 | El foco de esta investigación se centra en diseñar un eje regulador entre las tareas de control y gestión de los proyectos, las metodologías de desarrollo de software adecuadas y por último los tipos de herramientas que darán soporte a toda la metodología descripta en ambientes distribuidos. Por lo tanto se busca la relación en tres capas entre gestión, metodología y herramientas como un proceso completo cuyo objetivo es simplificar todas las tareas que forman parte de proceso de desarrollo de software y en donde los recursos humanos juegan un papel relevante en todo este esquema. Palabras claves: Metodología de desarrollo de software, equipos de desarrollo de software dispersos geográficamente, herramientas para el soporte de desarrollo de software, características de equipos dispersos geográficamente, control y gestión de proyectos de desarrollo de software distribuidos |
| Univ.de San Andrés-Buenos Aires –Argentina | Lic. Juan P. Grassi | Tesis de Maestría: Estudio sobre metodologías para la gestión de proyectos de software. Propuesta y aplicaciones | 2012 | En el campo de gestión del tiempo en proyectos de desarrollo de software, las metodologías ágiles pueden ser combinadas con metodologías basadas en ciclo de vida, y en esta combinación se logra un aporte de valor al proyecto que no puede ser conseguido utilizando estas herramientas por separado |
| PUCP – Perú | Jaime H. Samamé Silva | Tesis de Maestría: Aplicación de una metodología ágil en el desarrollo de un sistema de información | 2013 | El presente proyecto aplica una de estas metodologías ágiles, Programación Extrema (Extreme Programming), en un pequeño proyecto de software, utilizando herramientas de software libre como Java, y como repositorio de datos el estándar XML. El resultado de esta investigación aporta una guía del uso de la metodología ágil en un pequeño proyecto de software que tiene aplicabilidad dentro del ciclo de inteligencia |

Para un proyecto software: Metodología Ágil DSDM vs. SCRUM

| Universidad | Tesista | Titulo de la Tesis | Año | Abstract |
|---|--------------------------|--|------|---|
| Universidad de Castilla-La Mancha- España | D. Daniel Cabrero Moreno | Tesis Doctoral: Construcción y Evolución del Software Basados en Valor | 2009 | <p>La Ingeniería del Software Basada en Valor (ISBV) es una rama de investigación muy joven, bautizada con ese nombre en el año 2000 por Barry Boehm, que trata de predecir en qué tareas de Ingeniería del Software (IS) debe ser invertido el esfuerzo para que el software aporte más valor a las organizaciones.</p> <p>La presente Tesis Doctoral proporciona una revisión de todos los trabajos relacionados con la ISBV presentados en los foros de investigación desde el año 1997, cuando fue presentada la primera propuesta en este sentido. La revisión efectuada desvela una falta de consenso en el significado del término “valor”, y una serie de oportunidades de mejora muy importantes en esta rama de investigación.</p> <p>La alineación de las tareas de IS con los objetivos y prioridades de las compañías es un problema de plena actualidad en las organizaciones grandes y pequeñas, ya que sus objetivos son altamente dependientes del software, que constituye cada vez más una herramienta básica en cualquier actividad económica. Paradójicamente, la ISBV es al mismo tiempo un área poco madura, donde se necesita todavía mucho trabajo de investigación.</p> <p>Las aportaciones más relevantes realizadas a lo largo de este trabajo de investigación son la definición y ordenación de los conceptos de ISBV, la propuesta de una técnica genérica basada en dichos conceptos que permite combinar todas las propuestas de valor y aplicar metodológicamente el concepto de valor, y la introducción del concepto de valor en mantenimiento, diseño y desarrollo.</p> <p>Las propuestas realizadas están apoyadas en una serie de herramientas desarrolladas a medida para la validación y prueba de conceptos en la Dirección General de Tráfico.</p> |

Para un proyecto software: Metodología Ágil DSDM vs. SCRUM

| Universidad | Tesista | Título de la Tesis | Año | Abstract |
|------------------------------------|-----------------------------|---|------------|-----------------|
| Univ.Complutense de Madrid. España | Iván García-Magariño García | Tesis Doctoral: Un marco para la definición y transformación de modelos en los sistemas multiagentes | 2010 | |

