

تمرین دوم درس برنامه‌سازی چابک

محمد طه جهانی نژاد

۹۸۱۰۱۳۶۳

معرفی کلی متدولوژی Crystal Clear

خانواده‌ی Crystal، خانواده‌ای از متدولوژی‌ها هستند که در سال ۱۹۹۸ توسط کوبرن معرفی شدند. اعضای مختلف این خانواده برای پروژه‌های مختلف، بر اساس Criticality و Size (تعداد افراد دخیل در پروژه)، تعریف شده‌اند. هر متدولوژی Crystal با یک رنگ مشخص می‌شود که هرچه این رنگ تیره‌تر باشد، به این معنی است که پیاده‌سازی این متدولوژی سنگین‌تر می‌باشد. متدولوژی Crystal Clear، همانطور که از نام آن پیداست، بی‌رنگ‌ترین و چابک‌ترین عضو این خانواده است و امروزه در عمل برای پروژه‌هایی با سایز کوچک (حدود ۶ نفره) استفاده می‌شود.

★ نکته: هرچه تعداد اعضای دخیل در یک پروژه بیشتر باشد، مدیریت پروژه دشوارتر است و نیاز به متدولوژی سنگین‌تری داریم. اما هرچه Criticality پروژه بیشتر شود، نیازی به عوض کردن متدولوژی نیست و تنها باید جزئیات و دقت را افزایش دهیم.

ویژگی‌های مشترک این دسته از متدولوژی‌ها عبارتند از:

- فرایند تکراری افزایشی دارند و طول هر دوره‌ی تکرار بیشتر از ۴ ماه نمی‌تواند باشد.
- پشتیبانی از تیم‌های توزیع شده ندارد. همه‌ی افراد باید در یک مکان حضور داشته باشند.
- وابستگی زیادی به جریان درست اطلاعات بین افراد مختلف (یا به اصطلاح کوبرن، اسمز اطلاعاتی) و ارتباط درست بین افراد داخل هر تیم با همدیگر و اعضای دیگر تیم‌ها دارند. همانطور که بالاتر اشاره شد، متدولوژی Clear برای تیم‌هایی با اندازه‌ی کوچک کاربرد دارد. در این متدولوژی فقط یک تیم ایجاد داریم. نرم افزار قابل استفاده هم حداقل باید هر سه ماه یکبار تحویل داده شود. (اگرچه زمان‌های کوتاه‌تری پیشنهاد می‌شود) دقت کنید که منظور از تحویل (Delivery) لزوماً Deployment نیست و این تحویل می‌تواند در محیط ایجاد رخ دهد.

مقایسه‌ی Work Unit ها

بررسی Crystal Clear

در این بخش توضیحات کلی مربوط به فعالیت‌های انجام شده در دو متدولوژی DSDM و Crystal Clear بررسی و مقایسه می‌شود.

در متدولوژی Clear، سه فاز کلی برای انجام یک Release داریم:

- **فاز Chartering:** این فاز اولین فعالیت متدولوژی می‌باشد. در این مرحله تیم ایجاد تشکیل می‌شود و امکان سنجی اولیه نیز به صورت کلی و سریع انجام می‌شود. اطلاعات اولیه برای شروع کدنویسی پروژه (مانند معماری و نیازمندی‌های پروژه) در همین مرحله استخراج می‌شود.
- **فاز Cyclic Delivery:** موتور اصلی تحویل که در طی Work Unit هایی نرم افزار ساخته می‌شود. هر دوره‌ی تحویل باید بین یک هفته تا سه ماه زمان ببرد. همچنین، چونکه این متدولوژی مبتنی بر یادگیری و بازخورد می‌باشد، باید هر Release حداقل دو فاز تحویل دوره‌ای داشته باشد تا بتوان در دوره‌ی دوم از بازخوردها و آموخته‌های دوره‌ی اول استفاده کرد.
- **فاز Wrap Up:** در آخرین فعالیت این متدولوژی نرم افزار در محیط کاربر Deploy شده و کارهای مربوط به Post mortem (مانند مرور و بازنگری این Release) در این فاز انجام می‌شود. در پایان این سه فاز یک Release از محصول اصلی ساخته می‌شود. اگر محصول چندین Release مختلف داشته باشد، به ازای هر کدام باید این سه فاز طی شوند. اکنون به طور دقیق‌تر Work Unit های هر Cyclic Delivery را بررسی می‌کنیم. به طور کلی، هر فاز Cyclic Delivery باید شامل حداقل دو Delivery باشد. هر Delivery نیز شامل چهار فعالیت است:
 1. **فعالیت Recalibration:** بر اساس Release Plan یی که داریم، کارهایی که می‌خواهیم در این دوره انجام شود را مشخص می‌کنیم. دو محصول Project Map و Release Plan ممکن است در این مرحله دچار تغییر و تحول شوند.
 2. **ایجاد نرم افزار در Iteration ها:** در هر iteration یک قطعه از نرم‌افزار تولید می‌شود. در طی این تولید، Ambassador به نرم افزار تولید شده نظارت دارد و بازخوردهایی را برای آن ثبت می‌کند. هر Delivery باید شامل حداقل یک Iteration باشد.
 3. **مرحله‌ی Delivery:** در پایان همه‌ی Iteration ها، یک یا چند نماینده‌ی مشتری حضور پیدا می‌کنند و دموی نرم‌افزار به آن‌ها ارائه می‌شود. بازخوردهای این افراد برای مراحل بعدی ایجاد جمع آوری می‌شوند. همانطور که بالاتر ذکر شد، این Delivery می‌تواند در محیط ایجاد انجام شود و لزوماً معادل با Deployment نیست.

4. مرحله‌ی Reflect: بر اساس بازخوردهای دریافت شده، سه P در پروژه، یعنی Process, Product و Plan طی یک Facilitated Workshop به نام Reflection بازنگری می‌شوند. مجری این جلسه معمولاً Lead Designer می‌باشد؛ اما اگر لازم باشد که جریان پروژه پس از این بازخوردها به طور کلی تغییر کند، تعمداً مجری این جلسه را Executive Sponsor انتخاب می‌کنند.

نکته‌ای که حائز اهمیت است، این است که هر Iteration در بین فعالیت‌های چهارگانه‌ی فوق، خود شامل چندین Work Unit کوچک‌تر می‌باشد. به طور دقیق‌تر، هر Iteration شامل سه مرحله‌ی زیر است:

1. برنامه‌ریزی: در اینجا به صورت Task محور، این Iteration را هدف گذاری می‌کنیم. تمرکز ما در این مرحله بر روی Task هایی است که می‌خواهیم در این Iteration انجام شوند.

2. فعالیت روزانه: هر Iteration شامل چند روز است. در هر روز، یک جلسه‌ی روزانه‌ی ایستاده داریم. پس از آن برنامه‌نویسان چندین مرحله Integration انجام می‌دهند. هر Integration خود از چندین اپیزود تشکیل شده. در این اپیزودها بخش‌های کوچکی از نرم‌افزار ساخته می‌شوند اما با Codebase ادغام نمی‌شوند. در انتهای همه‌ی اپیزودها (یعنی در انتهای هر Integration) تست نرم‌افزار انجام می‌شود و Designer Programmer ها عملیات Integration با Codebase را انجام می‌دهند. چونکه مراحل Integration بسیار کوتاه مدت هستند و در یک روز چندین Integration داریم، عملاً در این متدولوژی Continuous Integration انجام می‌شود.

3. پایان Iteration: در پایان هر Iteration یک مراسم اتمام برگزار می‌شود. هر سه P در طی یک Workshop مورد بازنگری واقع می‌شوند و یک جشن پیتزای کوچکی هم داریم. طبق توضیحات فوق، Work Unit های متدولوژی Crystal Clear به صورت زیر هستند:

Cyclic Delivery ⇒ Delivery ⇒ Iteration ⇒ Integration ⇒ Episode

در واقع کوچک‌ترین واحد کاری این متدولوژی همان اپیزودها هستند که در طی آن‌ها تسک‌ها انجام می‌شوند و قطعات خیلی کوچک نرم‌افزار در آن تولید می‌شوند. مزیت این متدولوژی در این است که Continuous Delivery و Testing به صورت مستمر روی Codebase در حال انجام است.

بررسی DSDM

در متدولوژی DSDM، چهار فاز کلی داریم:

1. **فاز Feasibility:** در این فاز، انجام پروژه از نظر Business Case و از نظر تکنولوژی امکان‌سنجی می‌شود. توجیه اقتصادی پروژه در اینجا مورد بررسی واقع می‌شود. امکان‌سنجی در حدی انجام می‌شود که بدانیم باید پروژه ادامه پیدا کند یا خیر. شش محصول در DSDM داریم که توصیف اولیه آن‌ها در این فاز مشخص می‌شود.
2. **فاز Foundation:** ورودی این فاز شش محصول فاز قبلی هستند. در این مرحله، می‌دانیم که پروژه Feasible است و اکنون باید شش محصول گذشته را به حدی از بلوغ برسانیم که امکان شروع ایجاد نرم‌افزار وجود داشته باشد. اهداف اصلی این فاز عبارتند از تکمیل توجیه و منطق بیزینسی، استخراج معماری سطح بالا و تعریف Solution، نحوه‌ی مدیریت، ایجاد و تحویل نرم‌افزار، و همچنین نحوه‌ی پیاده‌سازی DSDM (در واقع Tailoring آن) در این پروژه.
3. **فاز Evolutionary Development:** در این فاز، که در واقع Work Unit های اصلی DSDM در بستر آن تعریف می‌شوند، نرم‌افزار به صورت تدریجی ایجاد می‌شود. نرم‌افزار مورد نظر از تعدادی Increment تشکیل شده و هر Increment در طی چند Timebox که هرکدام به مدت ۲ الی ۴ هفته طول می‌کشند، تولید می‌شود. در انتهای هر Increment مرحله‌ی Deployment انجام می‌شود. جزئیات بیشتر هر Timebox در ادامه توضیح داده می‌شود.
4. **فاز Deployment:** در این مرحله Increment تولید شده که می‌تواند همان محصول نهایی و یا جزئی از آن باشد، وارد محیط کاربر می‌شود. فعالیت‌های دیگری مانند Assembling و Reviewing هم در این فاز انجام می‌شود. اکنون به فعالیت‌های دقیق‌تر هر Timebox که همان Work Unit های DSDM می‌باشند می‌پردازیم. در هر ۵، Timebox فعالیت اصلی انجام می‌شود:
1. **مرحله‌ی Kick Off:** در این مرحله کل تیم ایجاد (شامل نقش‌های سمت Business و نقش‌های Technical)، به همراه مدیر پروژه، Coordinator و Visionary در طی یک جلسه‌ی کوتاه یک الی سه ساعته، اهداف یک Timebox را از روی Delivery Plan مرور می‌کنند. لازم به ذکر است که در صورت لزوم، Delivery Plan هم دستخوش تغییراتی می‌شود. معیارهای پذیرش برای User Story ها و وابستگی‌های خارجی و داخلی هر کدام بررسی شده و در نهایت یک دید کلی نسبت به User Story ها و Timebox ایجاد می‌شود.
2. **مرحله‌ی Investigation:** نیازمندی‌هایی که در جلسه‌ی Kick Off هدف گرفته شده‌اند، به Task ها شکسته شوند و معیار پذیرش برای آن‌ها تعریف شود. محصول اصلی که در این بخش ساخته می‌شود، Timebox Plan می‌باشد که تمام تیم ایجاد باید روی آن توافق داشته باشند. در جریان تولید این نقشه، از افراد مختلف سمت Business بازخورد گرفته می‌شود.

3. **مرحله Refinement:** معنی لغوی آن، تدقیق و افزایش جزئیات می‌باشد. ۶۰ تا ۸۰ درصد نیروی کاری در این مرحله صرف می‌شود و عملاً کدهای مربوط به ایجاد نرم‌افزار در این مرحله نوشته می‌شوند. هدف اصلی این مرحله انجام هرچه بیشتر از Timebox Plan، طبق تعریف Definition Of Done می‌باشد. در این مرحله، Analyst و Visionary به نیازمندی‌های پروژه توجه زیادی دارند و افراد مختلف تیم را هدایت و راهنمایی می‌کنند. در انتهای این مرحله یک جلسه در قالب Workshop انجام می‌شود، یک دمو به ذینفعان ارائه شده و ایرادات آن استخراج می‌شود. همچنین User Story هایی که Done نشده‌اند لیست می‌شوند و به مرحله بعدی منتقل می‌شوند. مستنداتی مانند Timebox Review Record ها نیز در طی این جلسه تولید می‌شوند.

4. **مرحله Consolidation:** باقی‌مانده‌ی کارها که در مرحله‌ی قبل مشخص شدند در اینجا انجام می‌شوند. در واقع این مرحله، یک جمع‌بندی نهایی است که در طی آن یک Solution Increment ریزدانه تولید می‌شود. در انتهای این مرحله محقق شدن اهداف Timebox و همچنین رفع ایرادات لیست شده در مرحله‌ی قبل هم بررسی می‌شوند. User Story هایی که محقق نشده‌اند هم به صورت Open در Requirements List باقی می‌مانند.

5. **مرحله Close Out:** یک جلسه‌ی کوتاه یک الی سه ساعته برای پایان یک Timebox. بازنگری‌های کلی Timebox گذشته و برنامه‌ریزی کلی برای Timebox آینده در این جلسه انجام می‌شود. دو نقش Visionary و Coordinator همه‌ی User Story ها را بررسی و Done بودن یا نبودن آن‌ها را تعیین می‌کنند. در کنار این موضوع، Visionary برای تسک‌ها و User Story های Done نشده تصمیم‌گیری می‌کند. (که به Timebox بعدی یا ریلیز منتقل شوند و یا کلاً از پروژه حذف شوند) یک Workshop با حضور تیم توسعه هم برگزار می‌شود و طی آن فرایندها بازنگری و در صورت نیاز مورد تغییر واقع می‌شوند.

مقایسه دو متدولوژی

به صورت کلی، قالب Process در دو متدولوژی شباهت‌های فراوانی دارد. در هر کدام از متدولوژی‌ها، فعالیت‌ها از Pre-Project آغاز شده و طی آن‌ها Feasibility بررسی می‌شود و یک معماری مختصر استخراج می‌شود. همانطور که مشاهده شد، در Clear به این فعالیت Chartering و در DSDM به آن Feasibility & Foundation Phase می‌گوییم. در واقع در متدولوژی Crystal Clear این دو فاز DSDM در طی یک فعالیت انجام می‌شوند که طبیعتاً علت این موضوع، سبک‌تر بودن متدولوژی Clear می‌باشد.

در ادامه، هر کدام از متدولوژی‌ها به صورت Iterative Incremental نرم‌افزار را ایجاد می‌کنند. تفاوت مهم این دو، در طول Work Unit ها می‌باشد. در متدولوژی Clear به Continuous Integration توجه بیشتری شده و به همین منظور، Work Unit ها را به بخش‌های ریزتری شکسته به حدی که، همانطور که بالاتر به آن اشاره شد، در طول هر روز ما چند Episode داریم که طی آن بخش‌های کوچک نرم‌افزار تولید می‌شوند. در نقطه‌ی مقابل، در متدولوژی DSDM، واحدهای کاری همان Timebox ها هستند که بین ۲ الی ۴ هفته طول می‌کشند و انجام Deployment در انتهای این Timebox انجام می‌شود. از این منظر، متدولوژی Clear شاید برای برخی پروژه‌های سبک‌تر بهتر عمل کند؛ زیرا بازخورد مداوم نماینده‌ی مشتری روی دموی محصول و Solution Increment ها وجود دارد و هدایت تیم راحت‌تر انجام می‌شود. البته باید در نظر داشت که متدولوژی DSDM برای پروژه‌هایی با اندازه‌ی بزرگتر تعریف شده و امکان انجام Delivery و دریافت بازخورد مداوم از نماینده‌ی مشتری به سادگی وجود ندارد. در این بخش هم، می‌توان گفت که متدولوژی Clear دو فاز Evolutionary Development و Deployment از DSDM را در یک فعالیت Cyclic Delivery خلاصه کرده است. در انتها هم، هر دو متدولوژی فعالیت‌های مربوط به Post Mortem را انجام می‌دهند و تجربیات کسب شده در طول این پروژه و آموخته‌های خود را برای استفاده در پروژه‌های بعدی در قالب مستندات ذخیره می‌کنند.

مقایسه‌ی Product ها

بررسی Crystal Clear

محصولاتی که در متدولوژی Clear وجود دارند به شرح زیر هستند:

- قراردادهای ذکر شده برای پیاده‌سازی متدولوژی (همان Convention ها)، که در فاز Chartering تمامی اعضا روی آن توافق می‌کنند.
- نقشه‌ی پروژه، که شامل وظایف ایجاد (Development Tasks) و وابستگی‌های بین آن‌ها می‌باشد. این نقشه در ابتدای هر Cyclic Delivery بازبینی می‌شود و در صورت نیاز تغییر می‌کند. در نتیجه این محصول یک محصول تکاملی است.
- نقشه‌ی Release، که زمان‌های برنامه‌ریزی شده‌ی دوره‌های تحویل و Iteration ها در آن مشخص شده است. در ابتدای هر Cyclic Delivery، مانند Project Map، این نقشه هم می‌تواند دستخوش تغییر شود.
- نقشه‌ی هر Iteration که توسط Coordinator تولید و نگهداری می‌شود.

- لیست نیازمندی‌ها، که تمامی نیازمندی‌هایی که نرم‌افزار نهایی باید آن‌ها را برطرف کند در آن ذکر شده است.
 - توصیف معماری (Architecture Description) که توسط Lead Designer و در اوایل پروژه تولید می‌شود و یک توصیف کلی از معماری نرم‌افزار نهایی می‌باشد.
 - کدها و تست‌های نوشته شده (Source Code and Tests) که توسط تیم توسعه یا همان Designer Programmer ها، در جریان تولید نرم‌افزار و به مرور تولید و تکمیل می‌شود.
- توجه کنید که محصولات فوق، استخراج شده از بین صحبت‌ها و اسلایدهای درس هستند. با مراجعه به کتاب کوبرن در مورد متدولوژی Crystal Clear، تعداد بیشتری از محصولات را مشاهده می‌کنیم که به گفته‌ی خود نویسنده، وجود همه‌ی آن‌ها برای همه‌ی پروژه‌ها نه اجباری و نه اختیاری هستند. بنابراین لیستی که در بخش بالا نوشته شده، تقریباً لیست بخشی از محصولات است که در جریان Process با آن درگیر هستیم و سر و کار داریم.

بررسی DSDM

در DSDM چهارده محصول مختلف داریم:

1. محصول Terms Of Reference که در آن تعریف کلی محصول ذکر شده است. این محصول یک محصول سمت Business است و برای کارهای Governance استفاده می‌شود.
2. محصول Business Case که توجیه اقتصادی پروژه در آن بررسی شده. این محصول یک محصول تکاملی است و طی فعالیت‌های مختلف DSDM به مرور ساخته می‌شود.
3. لیست نیازمندی‌های اولویت بندی شده یکی دیگر از محصولاتی است که در سمت Business قرار دارد و تیم ایجاد هم از آن استفاده می‌کنند. این لیست در جریان هر Increment و Iteration دچار تغییر و تکامل می‌شود.
4. تعریف معماری Solution محصولی است که در آن معماری سطح بالای نرم‌افزار هم از نظر محصولی و هم از نظر بیزینسی در آن تعریف شده است.
5. تعریف Development Approach، مشابه محصول قبل یک محصول در راستای Solution و یک محصول تکاملی است با این تفاوت که در آن تکنولوژی‌ها و ابزارهای مورد استفاده برای ایجاد نرم‌افزار مشخص شده است.
6. نقشه‌ی Delivery که نشان‌دهنده‌ی زمان‌بندی سطح بالای ایجاد Increment ها و Timebox های مرتبط با آن می‌باشد.

7. تعریف Management Approach، در این محصول نحوه‌ی کلی مدیریت پروژه و نحوه‌ی گزارش پیشرفت و ... مشخص می‌شود.
8. محصول Feasibility Assessment، محصولی که در انتهای فاز feasibility آماده می‌شود و نشان‌دهنده‌ی نتیجه‌ی آن فاز می‌باشد.
9. محصول Foundation Summary، همانند محصول قبلی، وضعیت Solution و Business را در انتهای فاز Foundation نشان می‌دهد. این دو محصول پایه‌های مهمی برای انجام فازهای بعدی هستند.
10. خود محصول در حال ساخت، یا Evolving Solution به عنوان مهم‌ترین محصول خروجی DSDM در دسته‌ی محصولات قرار دارد و به مرور ساخته می‌شود.
11. نقشه‌ی هر Timebox که در آن برنامه‌ریزی برای آن Timebox مشخص شده است. این محصول یک محصول تکاملی نیست ولی در ابتدای هر Timebox و در جریان Timebox planning هر بار ایجاد می‌شود.
12. محصول Timebox Review Record که شامل بازخوردهای ثبت شده در انتهای هر Timebox است. همانند محصول پیشین، این محصول هم هر بار در انتهای تایم‌باکس‌ها ایجاد می‌شود.
13. گزارش بازنگری پروژه، در این محصول بازخوردهای مربوط به هر Increment نوشته شده است. در انتهای هر Increment بدون تغییر محتوای سابق آن، بخش‌های جدیدی به آن افزوده می‌شود.
14. محصول Benefits Assessment، که در آن گزارش بررسی سوددهی پروژه نوشته می‌شود.

مقایسه دو متدولوژی

همانند بخش‌هایی از Work Unit که هر دو متدولوژی در آن مشترک بودند، در بخش Product ها هم، اشتراکاتی دارند که مربوط به همان Work Unit ها می‌باشد. محصولاتی مانند Release، Project Map، Plan و Timebox Plan و همچنین لیست نیازمندی‌ها با تعاریف متفاوتی در هر دو متدولوژی ذکر شده است. بدیهتاً این محصولات جزو پایه‌ی برنامه‌ریزی برای توسعه محصول می‌باشند و این اشتراک بین متدولوژی‌ها طبیعی است.

یکی از مهم‌ترین تفاوت‌هایی که در این قسمت مشاهده می‌شود، تاکید بیشتر DSDM بر روی انجام Documentation مداوم است؛ به طوری که در انتهای هر Timebox بازخوردهای ثبت شده در یک Timebox Review Record ثبت می‌شوند و همچنین در انتهای هر Increment گزارش مشابهی نیز نوشته و ذخیره می‌شود. در طرف دیگر، متدولوژی Clear به علت سبک‌تر بودن، این موارد را به صورت

اجباری مطرح نکرده است. در کتاب آقای کوپرن، یک نقش و یک محصول دیگر هم در انتهای تمامی محصولات ذکر شده است. نقش Writer به عنوان نگارنده‌ی Document ها در این متدولوژی وجود دارد اما تمرکز این شخص بیشتر بر روی Document های سمت کاربر و User Manual می‌باشد. بنابراین می‌توان گفت که در متدولوژی Clear با توجه به اینکه اندازه‌ی تیم شرکت‌کننده در ایجاد پروژه کوچک و حدود ۶ نفر می‌باشد، نیازی به این مستندات احساس نشده و بنابراین هزینه‌ی نوشتن آن‌ها را در این متدولوژی متحمل نمی‌شویم. اما در متدولوژی دیگر، یعنی DSDM، چون تیم سمت Business و تیم تکنیکال هر دو از اندازه‌ی نسبتاً بزرگتری برخوردار هستند، این مستندات برای اشتراک اطلاعات و تثبیت و عدم تغییر آن‌ها الزامی می‌باشد.

مقایسه Role ها

بررسی Crystal Clear

در اسلایدهای درس و در منابع قرار داده شده، صراحتاً صحبتی از نقش‌های مختلف در متدولوژی Clear نشده است. اما با مراجعه به کتاب آقای کوپرن متوجه می‌شویم که نقش‌های زیر به صورت کلی در این متدولوژی تعریف شده‌اند:

1. اسپانسر اجرایی (Executive Sponsor) که هدایت کننده‌ی پروژه می‌باشد. همانطور که در بخش Work Units ذکر شد، اگر در جلسات Reflection نیاز به تغییر مسیر پروژه وجود داشته باشد، این شخص به عنوان برگزار کننده‌ی جلسه انتخاب می‌شود.
2. نقش Coordinator که در کنار تیم ایجاد به ساخت نقشه راه‌های مختلف (نقشه راه پروژه، نقشه‌ی هر Release و ...) کمک می‌کند. در کل این شخص به طور جزئی‌تری نسبت به Sponsor نسبت به پیشرفت و هدایت پروژه مسئول و ناظر است.
3. نقش‌های Ambassador و Business Expert که مسئولیت آن‌ها ثبت بازخورد در مورد دموهای ارائه شده و همچنین ایجاد اولیه‌ی لیست نیازمندی‌ها می‌باشد.
4. نقش Lead Designer که تعریف معماری اولیه را تولید می‌کند. همچنین برگزاری جلسات Reflection با این شخص می‌باشد.
5. تیم ایجاد، یا همان Designer-Programmer ها که مالک و مسئول کدهای نوشته شده و همچنین تست‌ها و طرح‌های اولیه طراحی نرم‌افزار هستند.
6. دو نقش Tester و Writer هم در این متدولوژی تعریف شده که به ترتیب مسئول گزارش ایرادات نرم‌افزار و نوشتن User Manual هستند.

همانطور که قبلا هم اشاره شد، تمامی نقش‌های ذکر شده در بالا برای همه‌ی پروژه‌ها اجباری نیستند و بنا به نیاز پروژه و اعضا، این متدولوژی می‌تواند با انجام Tailoring متناسباً تغییر کند.

بررسی DSDM

در DSDM سیزده نقش مختلف دخیل هستند. از آنجا که این متدولوژی برای پروژه‌های با اندازه‌های مختلف استفاده می‌شود، در صورت نیاز بعضی از این نقش‌ها می‌توانند به صورت هرمی و با سلسله مراتب مختلف باز تعریف و استفاده شوند. نقش‌ها به صورت خلاصه به شرح زیر هستند:

1. نقش Business Sponsor، که در سطح پروژه بالاترین اختیارات را دارد. مسئولیت بررسی توجیه اقتصادی پروژه و اطمینان از حرکت آن روی همین مسیر به این شخص وابسته است.
2. نقش Business Visionary، که به عنوان دست راست اسپانسر عمل می‌کند. این شخص یک دید کلی از پروژه دارد و به صورت دقیق‌تری نسبت به اسپانسر در جریان حرکت پروژه بر روی مسیر محقق شدن آن دید کلی می‌باشد.
3. نقش Technical Coordinator، که به عنوان دست چپ اسپانسر عمل می‌کند. از هماهنگ بودن اعضای تیم تکنیکال اطمینان حاصل می‌کند و پیشرفت پروژه از نظر فنی تحت نظارت این شخص می‌باشد. این نقش و نقش قبلی شباهت بسیاری به یکدیگر دارند اما حوزه‌ی کاری آن‌ها کاملاً متفاوت است.
4. نقش Project Manager، که وظیفه‌ی مدیریت و پیاده‌سازی متدولوژی را در تیم ایجاد دارد. نسبت به Coordinator ارتباط بیشتری با تیم ایجاد دارد و ناظر به پیشرفت پروژه می‌باشد.
5. نقش Business Analyst، که در دو وجه عمل می‌کند. هم وظایف سمت Business دارد و هم سمت Technical. این فرد به ارتباط بهتر دو وجه پروژه کمک می‌کند.
6. نقش Team Leader، که نظارت مستقیم به تیم ایجاد دارد و نسبت به همکاری آن‌ها و برنامه‌ریزی‌های Delivery مسئول است.
7. نقش Business Ambassador، در واقع نماینده‌ی Business در بین تیم ایجاد است. در استخراج نیازمندی‌های اولیه‌ی پروژه و همچنین ثبت بازخوردها عامل موثری می‌باشد.
8. نقش Solution Developer، که همان برنامه‌نویس می‌باشد و در ایجاد نرم‌افزار مستقیماً نقش دارد. این افراد باید با Technical Coordinator ارتباط خوبی داشته باشند.
9. نقش Solution Tester، وظیفه‌ی آزمون نرم‌افزار و استخراج خطاها و Bug ها را به عهده دارد. این فرد باید Team Lead را نسبت به این خطاها آگاه کند.

10. نقش business Advisor، فردی که همواره به حضور او نیاز نیست. یک متخصص در یک زمینه business یی خاص که در شرایط مورد نیاز از او کمک گرفته می‌شود.
11. نقش Technical Advisor، معادل همان نقش قبلی در حوزه‌ی فنی. شخصی که در یک زمینه‌ی فنی تخصص دارد و در مواقع نیاز به تیم فنی مشورت می‌دهد.
12. و 13. دو نقش Workshop Facilitator و DSDM Coach هم در این متدولوژی تعریف شده‌اند. این نقش‌ها افراد مجزایی نیستند. نقش اول در هنگام برگزاری Workshop های مختلف به تیم‌ها کمک می‌کند و می‌تواند هر کدام از نقش‌های فوق باشد. نقش دوم هم به پیاده‌سازی DSDM در بین تیم پروژه کمک می‌کند.

مقایسه دو متدولوژی

همانطور که مشاهده شد، متدولوژی DSDM نقش‌های بیشتر و ریزتری تعریف کرده است. این نقش‌ها به طور عامده‌ای یک ساختار هرمی برای مدیریت اجزای مختلف پروژه در حوزه‌های بیزینسی و فنی را تشکیل می‌دهند. با وجود این ساختار هرمی، مدیریت تعداد اعضای بالای پروژه ممکن می‌شود. در طرف دیگر، در متدولوژی Clear، به علت سبک بودن و کوچک‌تر بودن اندازه‌ی تیم، نقش‌های بسیار متفاوتی تعریف نشده؛ و چه بسا که در یک تیم کوچک‌تر یک نفر می‌تواند چند نقش مختلف را (مانند Programmer و Tester) با هم داشته باشد.

نکته‌ای که در اینجا حائز اهمیت است این است که نقش‌های اصلی برای ایجاد خود Solution (یعنی Developer ها و Tester ها) و همچنین نقش‌های مربوط به مدیریت تیم و هدایت مسیر پروژه (مانند Sponsor و Coordinator) در هر دو متدولوژی وجود دارند. از این موضوع می‌توان فهمید که این دو دسته نقش حداقل‌های مورد نیاز برای انجام یک پروژه هستند و در متدولوژی DSDM لایه‌های مدیریتی بیشتری تعریف شده تا فرایند نظارت بر افراد تسهیل گردد.

به عنوان آخرین نکته، نقش Ambassador نیز در هر دو متدولوژی دیده می‌شود که نشان‌دهنده‌ی اهمیت بالای ثبت بازخورد می‌باشد. وجود چنین فردی به عنوان نماینده‌ی مشتری به کنترل راحت‌تر مسیر پیشرفت پروژه منطبق با خواسته‌های مشتریان کمک شایانی می‌کند.

جمع‌بندی نهایی

در این مستند هر دو متدولوژی با توجه نقش‌ها، محصولات و فعالیت‌های مختلفشان به تفصیل توضیح داده شده و با یکدیگر مقایسه شدند. این دو متدولوژی هر کدام مزایا و معایبی دارند و هر کدام برای دسته‌ی مشخصی از پروژه‌های ایجاد نرم‌افزار استفاده می‌شوند.

به طور خلاصه، متدولوژی DSDM به علت پیاده‌سازی جزئیات بیشتر و تاکید بر مستندسازی و تثبیت و انتشار اطلاعات، برای پروژه‌هایی با سطح اهمیت و اندازه‌ی بالاتر مفیدتر می‌باشد و این در حالی است که متدولوژی Clear به علت سبک‌تر بودن و عدم تاکید بر روی انجام فعالیت‌های زمان‌بر مانند مستندسازی، طبق گفته‌ی آقای کوبرن، برای پروژه‌های کوچک‌تر با اندازه‌ی حداکثر ۶ نفره قابل پیاده‌سازی است.

وجود این تفاوت‌ها میان این دو متدولوژی لزوماً نشانه‌ای برای برتری یکی بر دیگری نیست و هر تیم با توجه به پروژه‌ای که تعریف کرده و ویژگی‌هایی که از متدولوژی نیاز دارد، می‌تواند هر کدام را انتخاب و پیاده‌سازی کند.