



تمرین شماره سه
درس متدولوژی‌های ایجاد نرم‌افزار

استاد:

دکتر رامان رامسین

دانشجو:

حسین لطفی

زمستان ۹۹

فهرست مطالب

| | |
|----|---|
| ۵ | ۱- بررسی و مقایسه فرایند متدولوژی DSDM با فرایند متدولوژی Scrum |
| ۵ | ۱-۱- جامع، شفاف، منطقی، دقیق و سازگار بودن تعریف متدولوژی |
| ۵ | ۱-۱-۱- بررسی متدولوژی‌ها |
| ۶ | ۱-۲-۱- مقایسه متدولوژی‌ها |
| ۷ | ۲-۱- پوشش چرخه‌ی عمر عمومی ایجاد نرم‌افزار |
| ۷ | ۱-۲-۱- بررسی متدولوژی‌ها |
| ۹ | ۲-۲-۱- مقایسه متدولوژی‌ها |
| ۱۰ | ۳-۱- پوشش فعالیت‌های چتری |
| ۱۰ | ۱-۳-۱- بررسی متدولوژی‌ها |
| ۱۱ | ۲-۳-۱- مقایسه متدولوژی‌ها |
| ۱۳ | ۴-۱- بی‌درزی و همواری انتقال |
| ۱۳ | ۱-۴-۱- بررسی متدولوژی‌ها |
| ۱۳ | ۲-۴-۱- مقایسه متدولوژی‌ها |
| ۱۴ | ۵-۱- مبتنی بر نیازمندی‌های وظیفه‌ای و غیروظیفه‌ای بودن |
| ۱۴ | ۱-۵-۱- بررسی متدولوژی‌ها |
| ۱۴ | ۲-۵-۱- مقایسه متدولوژی‌ها |
| ۱۵ | ۶-۱- آزمون‌پذیری، ملموس بودن و قابلیت پیگیری تا نیازمندی |
| ۱۵ | ۱-۶-۱- بررسی متدولوژی‌ها |
| ۱۵ | ۲-۶-۱- مقایسه متدولوژی‌ها |
| ۱۷ | ۷-۱- تشویق به مشارکت فعالانه کاربر |
| ۱۷ | ۱-۷-۱- بررسی متدولوژی‌ها |
| ۱۷ | ۲-۷-۱- مقایسه متدولوژی‌ها |
| ۱۸ | ۸-۱- قابلیت اجرا و کارایی |
| ۱۸ | ۱-۸-۱- بررسی متدولوژی‌ها |

| | |
|----|--|
| ۱۸ | ۲-۸-۱- مقایسه متدولوژی‌ها |
| ۲۰ | ۹-۱- مدیریت پیچیدگی فرایند |
| ۲۰ | ۱-۹-۱- بررسی متدولوژی‌ها |
| ۲۰ | ۲-۹-۱- مقایسه متدولوژی‌ها |
| ۲۱ | ۱۰-۱- توسعه‌پذیری، مقیاس‌پذیری، قابلیت پیکربندی و انعطاف‌پذیری |
| ۲۱ | ۱-۱۰-۱- بررسی متدولوژی‌ها |
| ۲۱ | ۲-۱۰-۱- مقایسه متدولوژی‌ها |
| ۲۳ | ۱۱-۱- حوزه کاربرد |
| ۲۳ | ۱-۱۱-۱- بررسی متدولوژی‌ها |
| ۲۳ | ۲-۱۱-۱- مقایسه متدولوژی‌ها |
| ۲۴ | ۲- بررسی و مقایسه فرایند متدولوژی DAD با فرایند متدولوژی Crystal Clear |
| ۲۴ | ۱-۲- جامع، شفاف، منطقی، دقیق و سازگار بودن تعریف متدولوژی |
| ۲۴ | ۱-۱-۲- بررسی متدولوژی‌ها |
| ۲۵ | ۲-۱-۲- مقایسه متدولوژی‌ها |
| ۲۶ | ۲-۲- پوشش چرخه‌ی عمر عمومی ایجاد نرم‌افزار |
| ۲۶ | ۱-۲-۲- بررسی متدولوژی‌ها |
| ۲۸ | ۲-۲-۲- مقایسه متدولوژی‌ها |
| ۲۹ | ۳-۲- پوشش فعالیت‌های چتری |
| ۲۹ | ۱-۳-۲- بررسی متدولوژی‌ها |
| ۳۰ | ۲-۳-۲- مقایسه متدولوژی‌ها |
| ۳۲ | ۴-۲- بی‌درزی و همواری انتقال |
| ۳۲ | ۱-۴-۲- بررسی متدولوژی‌ها |
| ۳۲ | ۲-۴-۲- مقایسه متدولوژی‌ها |
| ۳۳ | ۵-۲- مبتنی بر نیازمندی‌های وظیفه‌ای و غیروظیفه‌ای بودن |
| ۳۳ | ۱-۵-۲- بررسی متدولوژی‌ها |
| ۳۳ | ۲-۵-۲- مقایسه متدولوژی‌ها |

- ۶-۲-۶- آزمون‌پذیری، ملموس بودن و قابلیت پیگیری تا نیازمندی ۳۴
- ۶-۲-۱- بررسی متدولوژی‌ها ۳۴
- ۶-۲-۲- مقایسه متدولوژی‌ها ۳۴
- ۶-۲-۷- تشویق به مشارکت فعالانه کاربر ۳۵
- ۶-۲-۷-۱- بررسی متدولوژی‌ها ۳۵
- ۶-۲-۷-۲- مقایسه متدولوژی‌ها ۳۵
- ۶-۲-۸- قابلیت اجرا و کارایی ۳۶
- ۶-۲-۸-۱- بررسی متدولوژی‌ها ۳۶
- ۶-۲-۸-۲- مقایسه متدولوژی‌ها ۳۶
- ۶-۲-۹- مدیریت پیچیدگی فرایند ۳۸
- ۶-۲-۹-۱- بررسی متدولوژی‌ها ۳۸
- ۶-۲-۹-۲- مقایسه متدولوژی‌ها ۳۸
- ۶-۲-۱۰- توسعه‌پذیری، مقیاس‌پذیری، قابلیت پیکربندی و انعطاف‌پذیری ۳۹
- ۶-۲-۱۰-۱- بررسی متدولوژی‌ها ۳۹
- ۶-۲-۱۰-۲- مقایسه متدولوژی‌ها ۴۰
- ۶-۲-۱۱- حوزه کاربرد ۴۲
- ۶-۲-۱۱-۱- بررسی متدولوژی‌ها ۴۲
- ۶-۲-۱۱-۲- مقایسه متدولوژی‌ها ۴۲

۱- بررسی و مقایسه فرایند متدولوژی DSDM با فرایند متدولوژی Scrum

۱-۱- جامع، شفاف، منطقی، دقیق و سازگار بودن تعریف متدولوژی

۱-۱-۱- بررسی متدولوژی‌ها

| | | |
|-------|--------------------------|--|
| DSDM | چرخه حیات و واحدهای کاری | چرخه حیات و واحدهای کاری به‌دقت، شفاف، بدون ابهام و با جزئیات تعریف شده‌اند. |
| | تولیدکننده‌ها | نقطه قوت اصلی DSDM مجموعه نقش‌های آن است. نقش‌ها به صورت جامع و کاملی توضیح داده شده‌اند. |
| | زبان مدل‌سازی | در این متدولوژی مدل‌سازی به صورت حداقلی و با صلاح دید تیم انجام می‌شود و هیچ نوع خاصی از مدل‌ها اجبار نشده است. ممکن است مدل‌سازی با هدف مدیریت پیچیدگی و انتقال اطلاعات بین اعضای تیم انجام گیرد. |
| | محصولات | محصولاتی که در این متدولوژی تولید می‌شوند به خوبی و بدون ابهام تعریف شده‌اند. |
| | تکنیک‌ها و قواعد | تکنیک‌ها و قواعد با جزئیات و دقیق بیان شده‌اند. |
| | فعالیت‌های چتری | فعالیت‌های چتری در این متدولوژی مورد توجه قرار گرفته‌اند و به خوبی تعریف شده‌اند. در ادامه و در زمان بررسی فعالیت‌های چتری بیشتر به آن‌ها می‌پردازیم. |
| Scrum | چرخه حیات و واحدهای کاری | این متدولوژی یکی از پرستفاده‌ترین متدولوژی‌ها است و کتاب‌ها و منابع زیادی در مورد آن وجود دارد. بنابراین توضیحات جامع و کاملی برای آن وجود دارد. |
| | تولیدکننده‌ها | نماینده مشتری، scrum master و تیم ایجاد از جمله نقش‌هایی هستند که در این متدولوژی تعریف شده‌اند. شرح وظایف هر یک از این نقش‌ها به خوبی بیان شده است. |
| | زبان مدل‌سازی | این متدولوژی اصطلاحاً یک متدولوژی model-phobic است و برای افزایش چابکی سعی می‌کند از مدل‌سازی دوری کند. با این که خود متدولوژی تمایلی به مدل‌سازی ندارد ولی در صورت نیاز می‌توان مدل‌سازی را به عنوان یک product backlog item اضافه کرد. |
| | محصولات | product backlog، sprint backlog و potentially shippable product increment از جمله محصولاتی هستند که در Scrum تولید می‌شوند. تعریف شفاف و کاملی از هر یک از این محصولات وجود دارد. |
| | تکنیک‌ها و قواعد | تکنیک‌ها و قواعد به خوبی بیان شده‌اند. |
| | فعالیت‌های چتری | این متدولوژی تمام فعالیت‌های چتری را پوشش می‌دهد. این فعالیت‌ها به صورت مطلوبی مشخص شده‌اند. |

شباهت‌ها و تفاوت‌ها

- هر دو متدولوژی با محوریت فرایند تعریف شده‌اند.
- مجموعه محصولات DSDM بسیار کامل‌تر از Scrum است.
- در هر دو متدولوژی نقش‌ها به خوبی تعریف شده‌اند.
- در هر دو متدولوژی محصولات با جزئیات و بدون ابهام تعریف شده‌اند.
- در DSDM مدل‌سازی تا حدی که به چابکی صدمه نزنند انجام می‌شود در حالی که Scrum به شدت از مدل‌سازی دوری می‌کند.
- در هر دو متدولوژی واحدهای کاری شفاف و کامل بیان شده‌اند.
- در هر دو متدولوژی تعریف فعالیت‌های چتری مطلوب است.

نتیجه‌گیری: از نظر جامع، شفاف، منطقی، دقیق و سازگار بودن تعریف متدولوژی می‌توان گفت دو متدولوژی در یک سطح قرار دارند.

۲-۱- پوشش چرخه‌ی عمر عمومی ایجاد نرم‌افزار

۱-۲-۱- بررسی متدولوژی‌ها

| تعریف | | DSDM |
|-------------------------------|---|------|
| اکتشاف و مدل‌سازی قلمرو مسئله | دو فاز اول فرایند این متدولوژی امکان‌سنجی و foundation هستند که به صورت ترتیبی اجرا می‌شوند. یکی از اهداف انجام این دو فاز تحلیل مقدماتی و رسیدن به چابکی سیستم است. قلمرو کسب‌وکار نیز مورد مطالعه قرار می‌گیرد. | |
| استخراج نیازمندی‌ها | در فاز دوم فرایند با نام foundation نیازمندی‌های سطح بالا استخراج می‌شوند. برای بیان نیازمندی‌ها از user story استفاده می‌شود. در بررسی‌های انجام شده مشخص شده است که در این فاز تنها قادر به استخراج ۲۰ درصد از نیازمندی‌ها هستیم. البته از دید مشتری این نیازمندی‌ها ۸۰ درصد از کل نیازمندی‌ها را تشکیل می‌دهند. سایر نیازمندی‌های ریزدانه‌تر در طول انجام پروژه پدیدار می‌شوند. در فاز evolutionary development جزئیات به نیازمندی‌ها اضافه می‌شوند. | |
| امکان‌سنجی | یکی از فازهای فرایند DSDM امکان‌سنجی نام دارد که در ابتدای اجرای فرایند انجام می‌شود. در این فاز باید نهایتاً در عرض یک هفته ریسک‌های عمده شناسایی شوند. باید از وجود نیرو، سواد و تکنولوژی لازم برای انجام پروژه اطمینان حاصل کنیم. همچنین مقرون به صرفه بودن انجام پروژه از دید کسب‌وکار نیز در این فاز بررسی می‌شود. در انتهای این فاز تصمیم‌گیری می‌شود که پروژه ادامه پیدا کند یا متوقف شود. | |
| ایجاد | | |
| طراحی معماری | در فاز foundation معماری مشخص می‌شود و با انجام prototyping سعی به هرچه دقیق‌تر کردن آن می‌شود. ممکن است در شرایط خاص پس از استقرار نیز معماری دستخوش تغییر شود که البته بسیار پرهزینه خواهد بود. | |
| طراحی تفصیلی | در فاز evolutionary development که به صورت iterative incremental انجام می‌شود طراحی تفصیلی انجام می‌شود. | |
| برنامه‌نویسی | در فاز evolutionary development که به صورت iterative incremental انجام می‌شود برنامه‌نویسی انجام می‌شود. در تیم ایجاد افرادی که نقش solution developer را دارند مسئول برنامه‌نویسی هستند. | |
| آزمون | در فاز evolutionary development آزمون‌های مختلف به صورت پیوسته انجام می‌شوند. در تیم ایجاد کار solution testerها فقط تست کردن است. از جمله تست‌هایی که انجام می‌دهند می‌توان به unit testing، integration testing، system testing و acceptance testing اشاره کرد. | |

| | | |
|-------------------------------|-------------------------|--|
| | استقرار | <p>یکی از فازهای فرایند استقرار است که به صورت دوره‌ای انجام می‌شود. فاز استقرار از سه فعالیت اصلی تشکیل شده است:</p> <p>۱- Assemble: در این قسمت incrementهای ایجاد شده assemble می‌شوند.</p> <p>۲- بازبینی: با انجام بازبینی‌های لازم از کیفیت رلیز اطمینان حاصل می‌شود.</p> <p>۳- استقرار: قرار دادن نرم‌افزار ایجاد شده در محیط کاربر</p> |
| نگهداری | | |
| نگهداری | فاقد مرحله نگهداری است. | |
| تعریف | | |
| اکتشاف و مدل‌سازی قلمرو مسئله | | <p>نماینده مشتری در قالب نقش product owner به صورت تمام وقت در کنار تیم ایجاد حضور دارد. نماینده مشتری دیدی کلی از آن چیزی که قرار است ساخته شود دارد. در جریان فرایندی با نام grooming این دید به اجزاء کوچکتری شکسته می‌شود و این اجزاء در product backlog list به ترتیب اولویت جمع‌آوری می‌شوند.</p> |
| استخراج نیازمندی‌ها | | <p>نیازمندی‌ها به صورت user story استخراج می‌شوند. این user storyها در جریان grooming مبنای ایجاد product backlog قرار می‌گیرند.</p> |
| امکان‌سنجی | | <p>Scrum فعالیت صریحی برای پشتیبانی از امکان‌سنجی انجام نمی‌دهد.</p> |
| ایجاد | | |
| طراحی معماری | | <p>در این متدولوژی صراحتاً تلاشی برای هیچ نوع از طراحی از جمله طراحی معماری نمی‌شود.</p> |
| طراحی تفصیلی | | <p>در این متدولوژی صراحتاً تلاشی برای هیچ نوع از طراحی از جمله طراحی تفصیلی نمی‌شود.</p> |
| برنامه‌نویسی | | <p>از جمله کارهایی که تیم ایجاد در زمان sprint execution به آن می‌پردازد، برنامه‌نویسی است.</p> |
| آزمون | | <p>به آزمون نیز در این متدولوژی توجه شده است. آزمون‌های لازم به صورت مجموعه‌ای از taskها در داخل sprint backlogها برنامه‌ریزی می‌شوند.</p> |
| استقرار | | <p>در انتهای هر اسپرینت نسخه‌ای از محصول که قابلیت استقرار دارد تولید می‌شود ولی لزومی به استقرار آن نیست. تصمیم برای استقرار یا عدم استقرار محصول یک تصمیم در سطح کسب‌وکار است و تحت تأثیر مواردی مثل موارد زیر گرفته می‌شود:</p> <ul style="list-style-type: none"> - آیا به اندازه کافی feature برای استقرار پیاده‌سازی شده است؟ - آیا مشتری تحمل تغییر در یک بازه زمانی کوتاه را دارد؟ |
| نگهداری | | |
| نگهداری | | <p>مانند اکثر متدولوژی‌های چابک فاقد مرحله نگهداری است و برای نگهداری باید از سایر روش‌ها در کنار آن استفاده کنیم.</p> |

۱-۲-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- هر دو متدولوژی فاقد فاز نگهداری هستند.
- در Scrum تحلیل مقدماتی نداریم در حالی که در DSDM تحلیل مقدماتی انجام می‌شود.
- در DSDM طراحی معماری داریم در حالی که در Scrum خبری از آن نیست.
- در DSDM طراحی تفصیلی انجام می‌شود در حالی که Scrum نسبت به آن بی‌توجه است.
- در DSDM در فاز post-project فعالیت‌های پس‌امرگی انجام می‌شود در حالی که در Scrum نشانی از فعالیت‌های پس‌امرگی نیست.
- فرایند هر دو متدولوژی iterative incremental است.

نتیجه‌گیری: DSDM از نظر پوشش چرخه حیات عمومی ایجاد نرم‌افزار از Scrum کامل‌تر است و بخش‌های بیشتری از آن را پوشش می‌دهد.

۳-۱- پوشش فعالیت‌های چتری

۱-۳-۱- بررسی متدولوژی‌ها

| | | |
|---|---------------------|------|
| <p>از بسیاری از تکنیک‌های مدیریت ریسک در این متدولوژی استفاده شده است. به این تکنیک‌ها در زیر اشاره شده است:</p> <ul style="list-style-type: none"> - امکان‌سنجی اولیه - Prototyping - فرایند iterative incremental - مشارکت فعالانه کاربر - رلیز زودهنگام - اعتبارسنجی و صحت‌سنجی مداوم - بازبینی‌های دوره‌ای - یکپارچه سازی مداوم - استفاده از project approach questionnaire برای ارزیابی ریسک فرایند DSDM | <p>مدیریت ریسک</p> | DSDM |
| <p>در فاز foundation برنامه‌ریزی و زمان بندی اولیه انجام می‌شود. نقشی با عنوان project manager در این متدولوژی تعریف شده است که مدیریت کلی پروژه را بر عهده دارد.</p> <p>یکی از محصولاتی که در این متدولوژی ایجاد می‌شود management approach definition نام دارد که در آن روش مدیریت پروژه مشخص می‌شود. نحوه سازماندهی و برنامه‌ریزی پروژه نیز در آن ثبت می‌شوند. محصول دیگری که برای مدیریت پروژه ایجاد می‌شود delivery plan نام دارد که یک برنامه سطح بالاست.</p> <p>از timebox plan هم برای برنامه‌ریزی جزئی در سطح timebox استفاده می‌شود.</p> <p>از جمله practice‌هایی از DSDM که به مدیریت پروژه مربوط می‌شود می‌توان به موارد زیر اشاره کرد:</p> <ul style="list-style-type: none"> - Timeboxing: تعیین ضرب‌الاجل برای رسیدن به اهداف موردنظر - Project planning and control: برنامه‌ریزی سطح بالا و سطح پایین و هم چنین تعیین فعالیت‌های کنترلی | <p>مدیریت پروژه</p> | |
| <p>یکی از قواعد این متدولوژی این است که هرگز نباید بر سر کیفیت سازش کرد که این خود نشان دهنده اهمیتی است که این متدولوژی به کیفیت می‌دهد.</p> <p>وجود موارد زیر در DSDM به تضمین کیفیت کمک می‌کند:</p> <ul style="list-style-type: none"> - مشارکت فعالانه کاربر - قابلیت رهگیری تا نیازمندی‌ها - اعتبارسنجی و صحت‌سنجی مداوم | <p>تضمین کیفیت</p> | |

| | | |
|---|---------------------|-------|
| <p>- مرور و بازبینی فرایند به صورت دوره‌ای</p> | | |
| <p>این متدولوژی از نظر مدیریت ریسک بسیار قوی عمل می‌کند و اکثر تکنیک‌های مدیریت ریسک را پوشش می‌دهد. از جمله این تکنیک‌ها می‌توان به موارد زیر اشاره کرد:</p> <ul style="list-style-type: none"> - فرایند iterative incremental - مشارکت فعالانه کاربر به صورت دائمی نماینده مشتری در کنار تیم ایجاد و شرکت در sprint review - برنامه‌ریزی بر مبنای ریسک با اولویت دادن به کارهای با ریسک بیشتر در زمان اولویت بندی product backlog itemها - رلیز زود هنگام و مداوم - اعتبارسنجی و صحت‌سنجی مداوم با برگزاری sprint review | <p>مدیریت ریسک</p> | |
| <p>از نظر مدیریت پروژه نیز Scrum به خوبی عمل می‌کند. موارد زیر جهت مدیریت پروژه در این متدولوژی انجام می‌شود:</p> <ul style="list-style-type: none"> - انجام sprint planning در ابتدای هر اسپرینت - برگزاری جلسات متعدد از جمله daily Scrum, sprint review و sprint retrospective باعث همکاری و ارتباط هر چه بهتر بین اعضای تیم می‌شود. - اصلاح و بازبینی دائمی برنامه - زمان بندی با مشخص کردن time boxها | <p>مدیریت پروژه</p> | Scrum |
| <p>وجود موارد زیر در Scrum به تضمین کیفیت کمک می‌کند:</p> <ul style="list-style-type: none"> - قابلیت رهگیری تا نیازمندی‌ها - اعتبارسنجی و صحت‌سنجی مداوم - جلسات مرور و بازبینی به صورت دوره ای - مشارکت فعالانه کاربر - در این متدولوژی می‌توان definition of done را به گونه‌ای تعریف کرد که کیفیت مطلوب حاصل شود. | <p>تضمین کیفیت</p> | |

۱-۳-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- با این که هم در Scrum و هم در DSDM تضمین کیفیت مهم است ولی DSDM یک قدم فراتر از Scrum رفته و علاوه بر تایید مشتری به دنبال خوشحالی مشتری نیز هست.
- در DSDM نقش project manager برای مدیریت پروژه تعریف شده است در حالی که نقش مستقلی برای مدیریت پروژه در Scrum نداریم.
- هر دو متدولوژی به مدیریت ریسک توجه داشته و از تکنیک‌های مختلفی که به آن‌ها اشاره شد برای آن استفاده می‌کنند. با این حال DSDM به خاطر امکان‌سنجی اولیه و prototyping از نظر مدیریت ریسک بهتر از Scrum است.

- در هر دو متدولوژی پیش از آغاز پیاده‌سازی برنامه‌ریزی انجام می‌شود.

نتیجه‌گیری: در هر دو متدولوژی فعالیت‌های چتری انجام می‌شوند. ولی به خاطر توجه بیشتر DSDM به این فعالیت‌ها در این معیار DSDM بر Scrum برتری دارد.

۴-۱- بی‌درزی و همواری انتقال

۱-۴-۱- بررسی متدولوژی‌ها

| | | |
|--|---------------|-------|
| <p>در این متدولوژی برای مدل کردن نیازمندی‌ها از user story استفاده می‌شود. user storyها شیء‌گرا نیستند و با قرار دادن آن‌ها به عنوان مبنای ایجاد سایر مدل‌ها paradigm shift ایجاد می‌شود که باعث بی‌درز نبودن این متدولوژی می‌شود.</p> | بی‌درزی | DSDM |
| <p>در این متدولوژی مانند سایر متدولوژی‌های چابک به خاطر iterative بودن فرایند و چرخه‌های کوتاه ایجاد، انتقال به صورت هموار اتفاق می‌افتد.</p> | همواری انتقال | |
| <p>این متدولوژی بی‌درز نیست زیرا از تعریف user story به تعریف شیء‌گرا می‌پردازیم. اگر user storyها خیلی ریزدانه باشند کمتر مشکل ایجاد می‌شود.</p> | بی‌درزی | Scrum |
| <p>به دلیل iterative incremental بودن فرایند این متدولوژی و عدم ساخت مدل‌های کاملاً جدید در میانه انجام فرایند، انتقال در آن هموار است.</p> | همواری انتقال | |

۱-۴-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- هر دو متدولوژی به خاطر استفاده از user story که غیر شیء‌گرا است بی‌درز نیستند.
- در هر دو متدولوژی به خاطر iterative incremental بودن فرایند انتقال از فازی به فازی دیگر هموار است.

نتیجه‌گیری: از نظر بی‌درزی و همواری انتقال هر دو تقریباً در یک سطح قرار دارند.

۵-۱- مبتهی بر نیازمندی‌های وظیفه‌ای و غیروظیفه‌ای بودن

۱-۵-۱- بررسی متدولوژی‌ها

| | | |
|---|---------------------------------|--------------|
| <p>نیازمندی‌ها در قالب user story به صورت سطح بالا در ابتدای پروژه استخراج می‌شوند و به تدریج در حین انجام پروژه به آن‌ها جزئیات افزوده می‌شود.</p> <p>این متدولوژی مبتنی بر نیازمندی است و نیازمندی‌ها مبنای انجام سایر قسمت‌های پروژه قرار می‌گیرند.</p> <p>یکی از محصولات این متدولوژی تولید می‌شود prioritized requirements list نام دارد که در آن توصیف سطح بالایی از نیازمندی‌ها نوشته می‌شود و نیازمندی‌ها در آن بر اساس ارزش در کسب‌وکار اولویت بندی می‌شوند.</p> | <p>نیازمندی‌های وظیفه‌ای</p> | <p>DSDM</p> |
| <p>نیازمندی‌های غیروظیفه‌ای همانند نیازمندی‌های وظیفه‌ای در این متدولوژی مورد توجه قرار گرفته‌اند.</p> | <p>نیازمندی‌های غیروظیفه‌ای</p> | |
| <p>در این متدولوژی همه چیز بر مبنای product backlog item انجام می‌شود. خود product backlog item‌ها در ابتدا بیان‌کننده نیازمندی‌ها از دید کاربر هستند. پس می‌توان گفت این متدولوژی مبتنی بر نیازمندی‌ها است.</p> | <p>نیازمندی‌های وظیفه‌ای</p> | <p>Scrum</p> |
| <p>تیم ایجاد برای مشخص کردن آماده بودن potentially shippable product increment در انتهای هر اسپرینت، بر روی یک سری معیارهای تمام شدن (definition of done) توافق می‌کند. تا زمانی که این معیارها برآورده نشده باشند کار تیم در آن اسپرینت تمام نشده است. به صورت رایج نیازمندی‌های غیروظیفه‌ای در معیارهای تمام شدن قرار می‌گیرند.</p> | <p>نیازمندی‌های غیروظیفه‌ای</p> | |

۱-۵-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- هر دو متدولوژی مبتنی بر نیازمندی‌ها هستند.
- در هر دو متدولوژی از **user story** برای استخراج نیازمندی‌ها استفاده می‌شود.
- نیازمندی‌های غیروظیفه‌ای در هر دو متدولوژی مورد توجه قرار گرفته‌اند.
- هم در **DSDM** و هم در **Scrum** نیازمندی‌ها اولویت بندی می‌شوند و نیازمندی‌های با اولویت بالاتر زودتر پیاده‌سازی می‌شوند.

نتیجه‌گیری: با این معیار این دو متدولوژی در یک سطح از کیفیت قرار دارند.

۱-۶-۱- آزمون‌پذیری، ملموس بودن و قابلیت پیگیری تا نیازمندی

۱-۶-۱-۱- بررسی متدولوژی‌ها

| | | |
|-------|---------------------------|---|
| DSDM | آزمون‌پذیری | تعداد محصولات در این متدولوژی نسبتاً زیاد است که این امر به آزمون‌پذیری آسیب وارد می‌کند. از سوی دیگر محصولات ساده و قابل فهم هستند و یکدیگر را کامل می‌کنند و به یکدیگر جزئیات زائد اضافه نمی‌کنند که باعث افزایش آزمون‌پذیری می‌شود. |
| | ملموس بودن | محصولات هم برای کاربران و هم برای ایجادکنندگان ملموس و قابل درک هستند. |
| | قابلیت پیگیری تا نیازمندی | این متدولوژی requirements driven است برای همین قابلیت پیگیری تا نیازمندی در آن در سطح بسیار مطلوبی وجود دارد. |
| Scrum | آزمون‌پذیری | محصولاتی که در این متدولوژی تولید می‌شوند بسیار کم هستند (sprint backlog و product backlog). هر یک از این محصولات خود ساده و قابل فهم نیز هستند و یکدیگر را تکمیل می‌کنند. از آن جایی که در متدولوژی‌های چابک به شدت از کارهای اضافی پرهیز می‌کنیم جزئیات زائد هم در این محصولات مشاهده نمی‌شود. علاوه بر موارد ذکر شده آزمون‌های زیادی که در طی اجرای فرایند انجام می‌شوند نیز نشان دهنده آزمون‌پذیری بالای محصولات این متدولوژی هستند. |
| | ملموس بودن | در این متدولوژی به صراحت گفته شده است که کاری که در هر اسپرینت انجام می‌شود باید منجر به ایجاد چیزی با ارزش و ملموس از دید مشتری شود. از دید ایجاد کنندگان نیز محصولات بسیار ملموس هستند زیرا هم تعداد آنها کم است هم بسیار ساده هستند و پیچیدگی ندارند و هم کاربرد آنها در طول انجام پروژه به وضوح قابل مشاهده است. |
| | قابلیت پیگیری تا نیازمندی | این متدولوژی یک متدولوژی requirements driven است. از طریق product backlog قابلیت پیگیری تا نیازمندی در آن محقق شده است زیرا همه چیز بر اساس product backlog itemها انجام می‌شود. خود product backlog itemها در ابتدا بیان‌کننده نیازمندی‌ها از دید کاربر هستند. پس می‌توان گفت پیگیری تا نیازمندی به خوبی در این متدولوژی قابل انجام است. |

۱-۶-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- آزمون‌پذیری محصولات به علت تعداد کم‌تر آنها در Scrum بیشتر از DSDM است.

- محصولات هم در Scrum و هم در DSDM هم برای کاربر و هم برای تیم ایجاد ملموس هستند.
- در هر دو متدولوژی قابلیت پیگیری تا نیازمندی وجود دارد.

نتیجه گیری: از نظر آزمون پذیری محصولات Scrum بهتر از DSDM است ولی از نظر ملموس بودن محصولات و قابلیت پیگیری تا نیازمندی هر دو متدولوژی تقریباً در یک سطح قرار دارند.

۷-۱- تشویق به مشارکت فعالانه کاربر

۱-۷-۱- بررسی متدولوژی‌ها

| | | |
|---|----------------------|-------|
| <p>مشارکت فعالانه کاربر در قالب نقش‌های متعدد محقق می‌شود. پنج تا از این نقش‌ها در ارتباط با کسب‌وکار و به تبع آن با مشتری هستند که در زیر به بیان هر یک از آن‌ها می‌پردازیم:</p> <ul style="list-style-type: none"> - business ambassador که وظیفه آن انجام کارهای مربوط به سازمان و دادن بازخورد به صورت مداوم است و در کنار تیم ایجاد قرار دارد. - business analyst که وظیفه آن منتقل کردن نیازمندی‌ها و ویژگی‌های سازمانی به تیم است. - business advisor که حوزه کسب‌وکار را خوب می‌شناسد تیم ایجاد را پشتیبانی می‌کند. - business visionary که ویژگی‌های سیستم را از دید منافع سازمانی مشخص می‌کند و در سطح مدیریت پروژه فعالیت می‌کند. - business sponsor که تامین کننده منابع مالی پروژه است. دو نقش اول مشارکت فعالانه‌تری نسبت به بقیه نقش‌ها دارند. همچنین از طریق prototyping از کاربر مرتباً بازخورد گرفته می‌شود. | مشارکت فعالانه کاربر | DSDM |
| <p>از این نظر Scrum فوق العاده است. یکی از نقش‌ها در product Scrum owner است که نماینده مشتری است و تمام وقت در کنار تیم ایجاد حضور دارد. در جریان sprint review کارهایی که در طی اسپرینت انجام شده‌اند به نماینده‌های مشتری (غیر از نماینده ثابت مشتری) نشان داده می‌شوند و از آن‌ها بازخورد گرفته می‌شود. در product backlog grooming نیز با نظارت و تایید نماینده مشتری ایجاد، شکستن و افزودن جزئیات، اولویت بندی کردن و تخمین زدن product backlog itemها انجام می‌شود.</p> | مشارکت فعالانه کاربر | Scrum |

۱-۷-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- در هر دو متدولوژی مشتری در قالب یک نقش در کنار تیم ایجاد حضور دارد.
- در هر دو متدولوژی پیوسته از مشتری بازخورد گرفته می‌شود.
- نقش‌هایی که برای مشتری در نظر گرفته شده است در **DSDM** بیشتر از **Scrum** است. فعالیت‌های مربوط به مشتری در بین این نقش‌ها تقسیم شده است. در **DSDM** توجه ویژه‌ای به فعالیت‌های سطح سازمان شده است.

نتیجه‌گیری: مشارکت کاربر در **DSDM** بیشتر از **Scrum** است.

۸-۱- قابلیت اجرا و کارایی

۱-۸-۱- بررسی متدولوژی‌ها

| | |
|---|--|
| <p>با استفاده از suitability filter‌هایی که برای آن تعریف شده است می‌توان اطمینان حاصل کرد که این متدولوژی برای یک پروژه مناسب هست یا نه. فرایند این متدولوژی پیچیده نیست. همین مسئله باعث افزایش قابلیت اجرای آن می‌شود.</p> | <p>قابلیت اجرا</p> |
| <p>از مواردی که به کارایی این متدولوژی کمک می‌کنند می‌توان به موارد زیر اشاره کرد:</p> <ul style="list-style-type: none"> ▪ پیچیدگی کم واحدهای فرایند ▪ وابسته نبودن به ابزار خاص ▪ عدم توجه ایجادکنندگان به جزئیاتی که آن‌ها را از فعالیت‌های اصلی دور می‌کند. ▪ مکانیزم‌های قوی و متعدد مدیریت پروژه | <p>کارایی</p> <p style="text-align: center;">DSDM</p> |
| <p>فرایند این متدولوژی بسیار ساده است و با جلوگیری از انجام کارهای زائد مشکل پیچیدگی ندارد. امروزه از Scrum در بسیاری از شرکت‌ها استفاده می‌شود که این خود دلیل دیگری برای قابلیت اجرای بالای آن است.</p> | <p>قابلیت اجرا</p> |
| <p>از مواردی که به کارایی این متدولوژی کمک می‌کنند می‌توان به موارد زیر اشاره کرد:</p> <ul style="list-style-type: none"> ▪ پیچیدگی کم واحدهای فرایند ▪ وابسته نبودن به ابزار خاص ▪ عدم توجه ایجادکنندگان به جزئیاتی که آن‌ها را از فعالیت‌های اصلی دور می‌کند. ▪ مکانیزم‌های قوی و متعدد مدیریت پروژه ▪ مبتنی بر نیازمندی‌ها <p>مواردی که باعث آسیب زده به کارایی متدولوژی می‌شوند:</p> <ul style="list-style-type: none"> ▪ تکیه بر ارتباط چهره به چهره بین اعضای تیم برای انتقال اطلاعات که امروزه میدانیم این روش کفایت نمی‌کند. ▪ عدم مدل‌سازی و طراحی می‌تواند مشکل ساز شود. | <p>کارایی</p> <p style="text-align: center;">Scrum</p> |

۱-۸-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- واحدهای فرایندی در هر دو متدولوژی پیچیده نیستند.
- هیچ کدام از متدولوژی‌ها به ابزار خاصی وابسته نیستند.

- هر دو متدولوژی از مکانیزم‌های قوی و متعدد مدیریت پروژه بهره می‌برند.
- مبتنی بودن بر نیازمندی‌ها به کارایی هر دو متدولوژی کمک می‌کند.
- عدم مدل‌سازی در Scrum کارایی آن را کاهش می‌دهد در حالی که در DSDM به مدل‌سازی توجه شده است.

نتیجه‌گیری: Scrum از DSDM ساده‌تر است در نتیجه قابلیت اجرای آن بالاتر است ولی از نظر کارایی به علت مدل‌سازی بهتر DSDM بهتر است.

۹-۱- مدیریت پیچیدگی فرایند

۱-۹-۱- بررسی متدولوژی‌ها

| | | |
|---|-----------------------|--------------|
| <p>فازهای این متدولوژی به صورت زیر است (در صورت لزوم هر یک از این فازها در بخش‌های قبلی توضیح داد شده‌اند):</p> <p>۱- Pre-project ۲- Project-prope ۱- فازهای ترتیبی ۱- امکان‌سنجی ۲- Foundations ۲- فازهای تکراری ۱- evolutionary development ۲- استقرار ۳- post-project</p> <p>با شکستن فرایند به چندین مرحله پیچیدگی مدیریت شده است.</p> | <p>مدیریت پیچیدگی</p> | <p>DSDM</p> |
| <p>در Scrum پروژه به چرخه‌هایی با نام اسپرینت که حداکثر یک ماه طول می‌کشند شکسته می‌شود. در ابتدای هر اسپرینت در جریان sprint می‌کشند تعدادی از کارها که به شکل product backlog item تعریف شده‌اند برای انجام در آن اسپرینت انتخاب می‌شوند و به افراد سپرده می‌شوند. سپس هر product backlog item به تعدادی task شکسته می‌شود. این taskها بسیار ریزدانه هستند. همان طور که مشاهده می‌شود با شکستن کارهای درشت دانه به taskهای ریزدانه پیچیدگی تا حد مطلوبی کنترل می‌شود.</p> | <p>مدیریت پیچیدگی</p> | <p>Scrum</p> |

۱-۹-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- هر دو متدولوژی با شکستن فرایند به بخش‌های کوچک‌تر سعی کرده‌اند پیچیدگی واحدهای فرایندی را کاهش دهند.
- DSDM با تقسیم فرایند به چند iteration و Scrum با شکستن فرایند به چند اسپرینت سعی داشته‌اند پیچیدگی فرایند را مدیریت کنند.

نتیجه‌گیری: فرایند Scrum از DSDM پیچیدگی کم‌تری دارد. زیرا بخش‌های کم‌تری از چرخه عمومی ایجاد نرم‌افزار را پوشش می‌دهد.

۱-۱۰-۱- توسعه پذیری، مقیاس پذیری، قابلیت پیکربندی و انعطاف پذیری

۱-۱۰-۱-۱- بررسی متدولوژی‌ها

| | | |
|--|-----------------|-------|
| هیچ صحبتی از هسته قابل توسعه و نقاط توسعه نشده است در نتیجه می‌توان گفت توسعه پذیر نیست. | توسعه‌پذیری | DSDM |
| چون از فرمالیسم پشتیبانی نمی‌کند برای پروژه‌های life critical مناسب نیست اما چون تست‌های گوناگونی در طی اجرای آن انجام می‌شوند برای پروژه‌هایی در سایر سطوح حساسیت قابل استفاده است. این متدولوژی در پروژه‌هایی با تعداد افراد زیاد نیز قابل استفاده است. وجود مدل‌سازی در DSDM تاثیر مثبتی بر قابلیت مقیاس‌پذیری دارد. در کل می‌توان گفت مقیاس‌پذیری یکی از نقاط قوت این متدولوژی محسوب می‌شود. | مقیاس‌پذیری | |
| با استفاده از project approach questionnaire می‌توان با توجه به نیاز متدولوژی را پیکربندی کرد. | قابلیت پیکربندی | |
| در جریان بازبینی در فاز استقرار با برگزاری جلسات retrospective فرایند DSDM مورد بازبینی قرار می‌گیرد و در صورت نیاز تغییرات لازم در آن اعمال می‌شود. | انعطاف‌پذیری | |
| برای Scrum کتاب‌ها و منابع زیادی وجود دارد. در این کتاب‌ها روش‌هایی برای توسعه آن بیان شده است. پس می‌توان گفت این متدولوژی توسعه‌پذیر است. | توسعه‌پذیری | Scrum |
| چون از فرمالیسم پشتیبانی نمی‌کند برای پروژه‌های life critical مناسب نیست اما چون تست‌های گوناگونی در طی اجرای آن انجام می‌شوند برای پروژه‌هایی در سایر سطوح حساسیت قابل استفاده است. به دلیل نبود model-phobic بودن و اکتفا به ارتباط چهره به چهره برای ارتباط میان اعضای تیم مناسب پروژه‌هایی با تعداد افراد زیاد نیست. در کل می‌توان گفت Scrum از نظر مقیاس‌پذیری ضعیف است. | مقیاس‌پذیری | |
| Scrum یک چهارچوب فرد محور بر اساس مجموعه‌ای از ارزش‌ها، اصول و روش‌ها است که ساختاری را فراهم می‌کند که با استفاده از آن یک سازمان می‌تواند با توجه به نیازهای خود آن را پیاده‌سازی کند. پس می‌توان گفت قابل پیکربندی است. | قابلیت پیکربندی | |
| در انتهای هر اسپرینت جلسه‌ای با نام sprint retrospective برگزار می‌شود که در آن اعضای تیم ایجاد، product owner و scrum master حضور دارند. آن‌ها فرایند Scrum را بازبینی می‌کنند و در صورت نیاز تغییرات لازم را در آن اعمال می‌کنند. | انعطاف‌پذیری | |

۱-۱۰-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- Scrum توسعه‌پذیر است در حالی که DSDM این گونه نیست.
- هر دو متدولوژی در آغاز فرایند قابلیت پیکربندی و سفارشی سازی را دارند.
- برگزاری جلسات retrospective در هر دو متدولوژی به انعطاف‌پذیری آن‌ها کمک می‌کند.
- DSDM به علت توجه به دغدغه‌های سازمانی و مدل‌سازی بهتر نسبت به Scrum مقیاس‌پذیری بیشتری دارد.
- DSDM برای پروژه‌هایی با تعداد افراد بیشتر مناسب‌تر است.
- هر دو متدولوژی مناسب پروژه‌های life critical نیستند.

نتیجه‌گیری: از نظر قابلیت پیکربندی و انعطاف‌پذیری این دو متدولوژی تقریباً در یک سطح قرار دارند. از نظر توسعه‌پذیری Scrum و از نظر مقیاس‌پذیری DSDM بهتر است.

۱-۱-۱- حوزه کاربرد

۱-۱۱-۱- بررسی متدولوژی‌ها

| | | |
|---|--------------------|--------------|
| <p>چون فاقد فرمالیسم است مناسب پروژه‌های life critical نیست. ولی برای سایر سطوح حساسیت مناسب است.</p> <p>بدلیل این که فاقد مکانیزم نگهداری است مناسب پروژه‌هایی که نگهداری در آن‌ها مهم است نیست. مگر این که از مکانیزم‌های رایج نگهداری در کنار آن استفاده شود.</p> <p>چون سازگاری بالایی دارد برای پروژه‌هایی با نیازمندی‌های متغیر مناسب است.</p> <p>در کل می‌توان گفت حوزه کاربرد گسترده‌ای دارد.</p> | <p>حوزه کاربرد</p> | <p>DSDM</p> |
| <p>چون فاقد فرمالیسم و مدل‌سازی است مناسب پروژه‌های life critical نیست.</p> <p>بدلیل این که فاقد مکانیزم نگهداری است مناسب پروژه‌هایی که نگهداری در آن‌ها مهم است نیست. مگر این که از مکانیزم‌های رایج نگهداری در کنار آن استفاده شود.</p> <p>در این متدولوژی از تغییر استقبال می‌شود بنابراین برای پروژه‌هایی که احتمال تغییر نیازمندی‌ها در میانه انجام پروژه زیاد است مناسب است.</p> <p>بدلیل عدم مدل‌سازی و اتکا به ارتباط چهره به چهره برای پروژه‌هایی با تعداد افراد زیاد مناسب نیست.</p> <p>در سایر حوزه‌ها بدلیل سادگی و وجود منابع متعدد برای آن قابل استفاده و بسیار محبوب است.</p> | <p>حوزه کاربرد</p> | <p>Scrum</p> |

۱-۱۱-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- مجموعه محصولات DSDM بسیار کامل‌تر از Scrum است. در نتیجه DSDM نسبت به Scrum در طیف وسیع‌تری از پروژه‌ها قابل استفاده است.
- هیچ کدام از این دو متدولوژی به علت عدم پوشش فرمالیسم مناسب پروژه‌هایی با سطح بحرانی life critical نیستند.
- هر دو متدولوژی مناسب پروژه‌هایی با سطوح بحرانی C، D و E هستند.
- در DSDM نقش‌هایی برای رسیدگی به دغدغه‌های سازمانی تعریف شده است. این نشان می‌دهد که این متدولوژی بر خلاف Scrum در سطح سازمانی نیز کاربرد دارد.

نتیجه‌گیری: DSDM حوزه کاربرد گسترده‌تری نسبت به Scrum دارد.

۲- بررسی و مقایسه فرایند متدولوژی DAD با فرایند متدولوژی Crystal Clear

۲-۱-۱-۲-۱- جامع، شفاف، منطقی، دقیق و سازگار بودن تعریف متدولوژی

۲-۱-۱-۲-۱-۱-۲- بررسی متدولوژی‌ها

| | | |
|---|---------------------------------|---------------|
| <p>چرخه حیات و واحدهای کاری جامع، شفاف و به خوبی بیان شده‌اند. در ادامه برحسب نیاز به بررسی جزئیات آن خواهیم پرداخت.</p> | <p>چرخه حیات و واحدهای کاری</p> | |
| <p>در این متدولوژی نقش‌ها در دو دسته تقسیم بندی شده‌اند: ۱- نقش‌های اصلی: team lead, product owner, team member, architecture owner و stakeholder ۲- نقش‌های ثانویه: specialist, independent tester, domain expert, technical expert و integrator تعریف جامع و کاملی از هر یک از این نقش‌ها وجود دارد.</p> | <p>تولیدکننده‌ها</p> | |
| <p>غالباً از زبان UML استفاده می‌شود.</p> | <p>زبان مدل‌سازی</p> | |
| <p>محصولات در این متدولوژی به خوبی و با جزئیات تعریف شده‌اند.</p> | <p>محصولات</p> | |
| <p>چون این متدولوژی بیشتر یک چهارچوب است تکنیک‌ها و قواعد به صورت سطح بالا تعریف شده‌اند و جزئیات در حد ضرورت بیان شده‌اند.</p> | <p>تکنیک‌ها و قواعد</p> | DAD |
| <p>فعالیت‌های چتری در این متدولوژی با نام ongoing goals بیان شده‌اند که شامل موارد زیر هستند:</p> <ul style="list-style-type: none"> - انجام ماموریت پروژه - افزایش مهارت اعضای تیم - بهبود زیرساخت‌های موجود - بهبود فرایند و محیط تیم (کارهای مدیریت پروژه) - بکارگیری زیرساخت‌های موجود - مدیریت ریسک <p>علاوه بر این‌ها در این متدولوژی هدف از تضمین کیفیت فراتر است و به دنبال خوشحالی مشتری نیز هستیم. مشاهده می‌شود که فعالیت‌های چتری در این متدولوژی به خوبی مورد توجه قرار گرفته‌اند.</p> | <p>فعالیت‌های چتری</p> | |
| <p>چرخه حیات و واحدهای کاری جامع، شفاف و بدون ابهام تعریف شده‌اند. در بخش‌های بعدی به بررسی جزئی‌تر آن‌ها می‌پردازیم.</p> | <p>چرخه حیات و واحدهای کاری</p> | |
| <p>در این متدولوژی چند نقش اصلی داریم که در هر تیمی باید حضور داشته باشند. executive sponsor, lead designer, ambassador user و از جمله این نقش‌ها هستند. چند نقش دیگر نیز تعریف شده‌اند که در صورت نیاز و با صلاح دید تیم، به تیم اضافه می‌شوند. این نقش‌ها عبارت‌اند از system analyst - designer, programmer, business expert, testers, text-write و coordinator تعریف جامع و کاملی از هر یک از این نقش‌ها وجود دارد.</p> | <p>تولیدکننده‌ها</p> | Crystal Clear |

| | |
|------------------|---|
| زبان مدل سازی | استفاده از یک زبان مدل سازی خاص را اجبار نکرده است. ولی غالباً از UML استفاده می شود. |
| محصولات | محصولات به صورت دقیق و شفاف مشخص شده اند. |
| تکنیک ها و قواعد | تکنیک ها و قواعد به خوبی در این متدولوژی تعریف شده اند. |
| فعالیت های چتری | فعالیت های چتری از قبیل مدیریت ریسک، مدیریت پروژه و تضمین کیفیت در این متدولوژی در نظر گرفته شده اند و به شکل مطلوبی بیان شده اند. در بخش های بعدی بیشتر به فعالیت های چتری که در این متدولوژی انجام می شوند خواهیم پرداخت. |

۲-۱-۲- مقایسه متدولوژی ها

شباهت ها و تفاوت ها

- هر دو متدولوژی با محوریت فرایند تعریف شده اند.
- در هر دو متدولوژی نقش ها به خوبی تریف شده اند.
- در هر دو متدولوژی محصولات با جزئیات و بدون ابهام تعریف شده اند.
- در هر دو متدولوژی غالباً از UML برای مدل سازی استفاده می شود.
- در هر دو متدولوژی واحدهای کاری شفاف و کامل بیان شده اند.

نتیجه گیری: با این معیار دو متدولوژی در یک سطح قرار دارند.

۲-۲- پوشش چرخه‌ی عمر عمومی ایجاد نرم‌افزار

۱-۲-۲- بررسی متدولوژی‌ها

| تعریف | | DAD |
|-------------------------------|---|-----|
| اکتشاف و مدل‌سازی قلمرو مسئله | در فاز inception به اکتشاف و مدل‌سازی قلمرو مسئله می‌پردازیم. از جمله کارهایی که در این فاز انجام می‌شوند می‌توان تشکیل تیم، استخراج نیازمندی‌های اولیه، تهیه معماری اولیه، امکان‌سنجی اولیه، همسویی با استراتژی سازمان، دست‌یابی به چشم‌انداز مشترک با ذی‌نفعان و راه‌اندازی محیط را نام برد. مدل‌سازی قلمرو مسئله نیز در صورت احساس نیاز انجام می‌شود. | |
| استخراج نیازمندی‌ها | یکی از کارهایی که در فاز inception و در مرحله collaborate انجام می‌شود استخراج نیازمندی‌های اولیه است. سایر نیازمندی‌ها در فاز construction و در مرحله collaborate استخراج می‌شوند و به آن‌ها جزئیات افزوده می‌شود و اولویت بندی می‌شوند. نیازمندی‌ها در این فاز ممکن است دستخوش تغییر شوند. | |
| امکان‌سنجی | در فاز inception در مرحله collaborate امکان‌سنجی اولیه انجام می‌شود. | |
| ایجاد | | |
| طراحی معماری | در فاز inception و در مرحله collaborate یک معماری اولیه مطرح می‌شود. در فاز construction بر روی معماری اولیه بررسی‌های بیشتری انجام می‌شود. معماری در صورت نیاز تغییر می‌کند و یا در صورتی که مناسب باشد تثبیت می‌شود. | |
| طراحی تفصیلی | در فاز construction در مرحله collaborate طراحی انجام می‌شود. البته باید توجه داشت که طراحی به شدت متدولوژی‌های سنگین وزن انجام نمی‌شود تا چابکی متدولوژی حفظ شود. | |
| برنامه‌نویسی | در فاز construction به صورت iterative incremental برنامه‌نویسی انجام می‌شود. | |
| آزمون | در فاز construction و هم در فاز transition تست‌هایی انجام می‌شوند. از جمله فعالیت‌هایی که در فاز construction انجام می‌شوند می‌توان موارد زیر را نام برد: <ul style="list-style-type: none"> - Regression testing - Test driven development - Acceptance test driven development - Parallel independent testing در فاز transition نیز end-of-lifecycle testing انجام می‌شود. | |
| استقرار | در فاز transition پس از اطمینان از آماده بودن محصول، محصول در محیط کاربر مستقر می‌شود. سایر کارهایی که در این فاز انجام می‌شوند عبارت‌اند از: <ul style="list-style-type: none"> - برنامه‌ریزی انتقال | |

| | |
|-------------------------------|---|
| | <ul style="list-style-type: none"> - انجام تست‌های نهایی - انتقال اطلاعات - نهایی کردن مستندات - آموزش ذی‌نفعان - آماده سازی محیط پشتیبانی |
| نگهداری | |
| نگهداری | <p>در این متدولوژی نگهداری به صورت یک فاز جداگانه تعریف نشده است بلکه گفته شده است پس از انجام اولین رلیز در تکرارهای بعدی فرایند به موازات ساخت رلیزهای بعدی باید کار نگهداری از رلیزهای قبلی نیز انجام شود.</p> |
| تعریف | |
| اکتشاف و مدل‌سازی قلمرو مسئله | <p>در فاز chartering با انجام کاوش ۳۶۰ درجه دید جامع، سطح بالا و با عمق کم از کل سیستم بدست می‌آید. یکی از کارهایی که در کاوش ۳۶۰ درجه انجام می‌شود ایجاد نمودارهای کلاس قلمرو مسئله است.</p> |
| استخراج نیازمندی‌ها | <p>نیازمندی‌های سطح بالا در فاز chartering و در زمان انجام کاوش ۳۶۰ درجه استخراج می‌شوند و معمولاً در قالب نمودار usecase ثبت می‌شوند.</p> |
| امکان‌سنجی | <p>در فاز chartering یکی از کارهایی که انجام می‌شود امکان‌سنجی اولیه است.</p> |
| ایجاد | |
| طراحی معماری | <p>در فاز chartering در مرحله کاوش ۳۶۰ درجه با بررسی تکنولوژی‌ها و معماری‌های مختلف معماری مناسب پروژه انتخاب می‌شود.</p> |
| طراحی تفصیلی | <p>در این متدولوژی فاز cyclic delivery از چند delivery تشکیل شده است. هر delivery از چند iteration تشکیل شده است. هر iteration در چند روز و در هر روز چند integration انجام می‌شود. هر integration نیز از چند episode تشکیل شده است. در هر episode طراحی، برنامه‌نویسی و آزمون واحد انجام می‌شود.</p> |
| برنامه‌نویسی | <p>همان‌طور که گفته شد در هر episode برنامه‌نویسی انجام می‌شود.</p> |
| آزمون | <p>همان‌طور که گفته شد در هر episode آزمون واحد انجام می‌شود. همچنین در انتهای هر integration آزمون یکپارچه سازی انجام می‌شود. در فاز wrap-up نیز acceptance testing انجام می‌شود.</p> |
| استقرار | <p>در فاز انتهایی یعنی فاز wrap-up محصول نهایی پس از acceptance testing در محیط کاربر مستقر می‌شود.</p> |
| نگهداری | |
| نگهداری | <p>فاقد مرحله نگهداری است.</p> |

Crystal Clear

۲-۲-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- فرایند هر دو متدولوژی iterative incremental است.
 - در Crystal Clear در فاز wrap-up فعالیت‌های پسامرگی انجام می‌شود در حالی که در DAD خبری از فعالیت‌های پسامرگی نیست.
 - در Crystal Clear خبری از فاز نگهداری نیست در حالی که در DAD به آن توجه شده است.
 - DAD بر خلاف Crystal Clear چرخه حیات عمومی ایجاد نرم‌افزار را به طور کامل پوشش می‌دهد.
 - در DAD نسبت به Crystal Clear توجه بیشتری به آزمون و استقرار شده است.
 - در Crystal Clear از usecase برای استخراج نیازمندی‌ها استفاده می‌شود در حالی که در DAD هم از usecase و هم از user story می‌توان برای استخراج نیازمندی‌ها استفاده کرد.
- نتیجه‌گیری:** با توجه به نکاتی که بیان شد می‌توان گفت با این معیار DAD از Crystal Clear بهتر است.

۳-۲- پوشش فعالیت‌های چتری

۱-۳-۲- بررسی متدولوژی‌ها

| | | |
|--|---------------------|------------|
| <p>این متدولوژی risk driven است که نشان دهنده توجه فوق العاده این متدولوژی به مدیریت ریسک‌ها است.</p> <p>DAD توصیه می‌کند که با کوچک نگه داشتن تیم و نزدیک نگه داشتن اعضای تیم از نظر مکانی به یکدیگر ریسک انجام پروژه کاهش یابد.</p> <p>در فاز inception شناسایی ریسک و امکان‌سنجی اولیه انجام می‌شود. موارد زیر نیز به مدیریت ریسک در این متدولوژی کمک می‌کند:</p> <ul style="list-style-type: none"> - فرایند iterative incremental prototyping - مشارکت فعالانه کاربر - رلیز اولیه زودهنگام و انجام سایر رلیزها در بازه‌های زمانی کوتاه - اعتبارسنجی و صحت‌سنجی مداوم - مرور و بازبینی دوره‌ای پروژه در reflection workshopها - یکپارچه سازی مداوم | <p>مدیریت ریسک</p> | |
| <p>در این متدولوژی برنامه‌ریزی و زمان بندی به شدت مورد توجه است. look-ahead planning, release planning, iteration planning و phase planning و transition planning از جمله سطوح مختلف برنامه‌ریزی هستند که در این متدولوژی انجام می‌شوند. همچنین اصراری بر ثابت بودن برنامه‌ها نیست و برنامه‌ها مرتباً بازنگری شده و به‌روز می‌شوند.</p> <p>برگزاری جلسات daily stand up و تاکید بر لزوم نزدیک بودن اعضای تیم از نظر مکانی باعث ارتباط هر چه بهتر اعضای تیم می‌شود. سیاست people first که این متدولوژی پیش گرفته است نیز به مدیریت پروژه کمک می‌کند.</p> | <p>مدیریت پروژه</p> | <p>DAD</p> |
| <p>وجود موارد زیر در DAD به تضمین کیفیت کمک می‌کند:</p> <ul style="list-style-type: none"> - اعتبارسنجی و صحت‌سنجی مداوم - مشارکت فعالانه کاربر و گرفتن بازخورد از مشتری به صورت مداوم - برگزاری جلسات daily stand up و retrospective و workshop به صورت دوره ای - توسعه مبتنی بر تست - انجام refactoring برای افزایش کیفیت کد - توجه به افزایش مهارت اعضای تیم - توجه به خوشحالی مشتری در کنار تایید مشتری - قابلیت پیگیری تا نیازمندی | <p>تضمین کیفیت</p> | |

| | | |
|---|--------------|---------------|
| <p>موارد زیر به مدیریت ریسک در این متدولوژی کمک می‌کند:</p> <ul style="list-style-type: none"> - فرایند iterative incremental - مشارکت فعالانه کاربر - امکان‌سنجی اولیه - prototyping - رلیز اولیه زود هنگام و انجام سایر رلیزها در بازه‌های زمانی کوتاه - اعتبارسنجی و صحت‌سنجی مداوم - یکپارچه سازی مداوم - بازبینی دوره‌ای در قالب برگزاری reflection workshopها | مدیریت ریسک | Crystal Clear |
| <p>موارد زیر جهت مدیریت پروژه در این متدولوژی انجام می‌شود:</p> <ul style="list-style-type: none"> - برنامه‌ریزی و زمان بندی اولیه در فاز chartering در قالب project map، release map و blitz plan - در ابتدای هر cyclic delivery، release plan به‌روزرسانی و اصلاح می‌شود. - در انتهای هر cyclic delivery برنامه مجدداً بررسی می‌شود. - برنامه‌ریزی ریزدانه در ابتدای هر iteration که تسک‌هایی که در آن iteration قرار است انجام شوند را مشخص می‌کند. - برگزاری جلسات روزانه و لزوم حضور اعضای تیم در نزدیکی یکدیگر باعث همکاری و ارتباط هر چه بهتر بین اعضای تیم می‌شود. | مدیریت پروژه | |
| <p>وجود موارد زیر در Crystal Clear به تضمین کیفیت کمک می‌کند:</p> <ul style="list-style-type: none"> - اعتبارسنجی و صحت‌سنجی مداوم - جلسات مرور و بازبینی به صورت دوره ای - مشارکت فعالانه کاربر و گرفتن بازخورد از مشتری به صورت مداوم - در انتهای هر cyclic delivery در reflection workshop تیم کیفیت محصول تحویل داده شده را ارزیابی می‌کند. - توسعه مبتنی بر تست <p>این متدولوژی قابلیت رهگیری تا نیازمندی به طور کامل را ندارد. این مسئله به تضمین کیفیت ضربه وارد می‌کند.</p> | تضمین کیفیت | |

۲-۳-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- با این که هم در DAD و هم در Crystal Clear تضمین کیفیت مهم است ولی DAD یک قدم فراتر از Crystal Clear رفته و علاوه بر تایید مشتری به دنبال خوشحالی مشتری نیز هست.
- DAD یک متدولوژی risk driven است یعنی از نظر مدیریت ریسک یک سطح بالاتر از Crystal Clear قرار دارد.

- در هر دو متدولوژی reflection workshop برگزار می‌شود.
- در هر دو متدولوژی توسعه مبتنی بر تست انجام می‌شود.
- هم در DAD و هم در Crystal Clear برنامه‌ریزی هم به صورت درشت دانه و هم به صورت ریزدانه انجام می‌شود.
- ارتباط بین تیم‌ها در DAD در نظر گرفته شده است در صورتی که در Crystal Clear این گونه نیست.
- نتیجه‌گیری: هر دو متدولوژی توجه لازم را به فعالیت‌های چتری داشته‌اند ولی این توجه در DAD بیشتر از Crystal Clear بوده است.

۲-۴- بی‌درزی و همواری انتقال

۲-۴-۱- بررسی متدولوژی‌ها

| | | |
|--|---------------|---------------|
| <p>در این متدولوژی برای استخراج نیازمندی‌ها می‌توان از usecase و یا user story استفاده می‌شود. هر دو این موارد شیء‌گرا نیستند و اگر بلافاصله به سراغ پیاده‌سازی برویم با درز مواجه می‌شویم. البته روش‌هایی برای کم کردن این درز وجود دارد. اگر از user story استفاده می‌شود می‌توان با ریزدانه کردن آن‌ها و اگر از usecase استفاده می‌شود می‌توان با ایجاد نمودار توالی این درز را کاهش داد.</p> | بی‌درزی | DAD |
| <p>انتقال در این متدولوژی از فازی به فاز دیگر و از مرحله‌ای به مرحله‌ای دیگر هموار است. زیرا فرایند این متدولوژی iterative incremental است.</p> | همواری انتقال | |
| <p>این متدولوژی بی‌درز نیست. زیرا از نمودار usecase که غیر شیء‌گرا است به تسک‌ها و پس از مشخص شدن تسک‌ها به سراغ پیاده‌سازی می‌رویم.</p> | بی‌درزی | Crystal Clear |
| <p>در این متدولوژی مانند سایر متدولوژی‌های چابک به خاطر iterative بودن فرایند و چرخه‌های کوتاه ایجاد، انتقال به صورت هموار اتفاق می‌افتد.</p> | همواری انتقال | |

۲-۴-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- در هر دو متدولوژی درز دیده می‌شود.
- هر دو متدولوژی به علت iterative incremental بودن فرایندشان از انتقال هموار بهره می‌برند.

نتیجه‌گیری: با این معیار هر دو متدولوژی در یک سطح قرار دارند.

۵-۲- مبثنی بر نیازمندی‌های وظیفه‌ای و غیروظیفه‌ای بودن

۲-۵-۱- بررسی متدولوژی‌ها

| | | |
|--|---------------------------------|----------------------|
| <p>در فاز inception نیازمندی‌های اولیه استخراج می‌شوند و بر مبنای آن‌ها در فاز construction مجموعه‌ای از work item ها ایجاد می‌شوند. در هر iteration ,work item هایی با بیشترین اولویت انتخاب شده و backlog را تشکیل می‌دهند. سپس iteration backlog به مجموعه‌ای از تسک‌ها شکسته شده و در آن iteration این تسک‌ها انجام می‌شوند. بنابراین میتوان گفت این متدولوژی requirements driven است.</p> | <p>نیازمندی‌های وظیفه‌ای</p> | <p>DAD</p> |
| <p>این متدولوژی goal driven است اگر نیازمندی‌های غیروظیفه‌ای را به عنوان اهداف پروژه در نظر بگیریم می‌توانیم بگوییم این متدولوژی مبتنی بر نیازمندی‌های غیروظیفه‌ای است.</p> | <p>نیازمندی‌های غیروظیفه‌ای</p> | |
| <p>در فاز chartering در زمان انجام کاوش ۳۶۰ درجه نیازمندی‌های سطح بالا استخراج می‌شوند و در قالب نمودار usecase ثبت می‌شوند. در ادامه این usecase ها به تعدادی تسک شکسته می‌شوند. این متدولوژی مبتنی بر نیازمندی نیست زیرا برنامه‌ریزی و فعالیت‌ها در آن لزوماً بر مبنای نیازمندی‌ها نیستند. به عنوان مثال blitz planning بر مبنای تسک‌ها انجام می‌شود.</p> | <p>نیازمندی‌های وظیفه‌ای</p> | <p>Crystal Clear</p> |
| <p>در Crystal Clear به دلیل وجود معماری نیازمندی‌های غیروظیفه‌ای مورد توجه قرار گرفته است.</p> | <p>نیازمندی‌های غیروظیفه‌ای</p> | |

۲-۵-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- DAD مبتنی بر نیازمندی است در حالی که Crystal Clear در بعضی از مواقع مبتنی بر نیازمندی و در بقیه مواقع مبتنی بر تسک است.

نتیجه‌گیری: با این معیار DAD برتر از Crystal Clear است.

۲-۶-۲- آزمون‌پذیری، ملموس بودن و قابلیت پیگیری تا نیازمندی

۲-۶-۱- بررسی متدولوژی‌ها

| | | |
|---|---------------------------|---------------|
| تعداد مصنوعات در این متدولوژی نسبتاً زیاد است که آزمون‌پذیری را خدشه دار می‌کند. از سوی دیگر اما محصولات ساده و قابل فهم هستند و یکدیگر را کامل می‌کنند و به یکدیگر جزئیات زائد اضافه نمی‌کنند و وابستگی کمی به یکدیگر دارند که باعث افزایش آزمون‌پذیری می‌شود. | آزمون‌پذیری | DAD |
| محصولات هم برای کاربران و هم برای ایجادکنندگان ملموس و قابل درک است. | ملموس بودن | |
| این متدولوژی requirements driven است و قابلیت پیگیری تا نیازمندی در آن وجود دارد. | قابلیت پیگیری تا نیازمندی | |
| محصولاتی که در این متدولوژی تولید می‌شوند کم هستند. هر یک از این محصولات خود ساده و قابل فهم هستند و یکدیگر را تکمیل می‌کنند. جزئیات زائد نیز در این محصولات مشاهده نمی‌شود. علاوه بر این محصولات در طی فرایند بارها مورد بررسی و تست قرار می‌گیرند که این خود نشان دهنده آزمون‌پذیری بالای محصولات این متدولوژی است. | آزمون‌پذیری | Crystal Clear |
| محصولات هم برای کاربران و هم برای ایجادکنندگان ملموس و قابل درک است. prototypeهایی که ایجاد می‌شوند برای کاربر بسیار ملموس هستند. | ملموس بودن | |
| به خاطر اعتبارسنجی و بازبینی کیفیت که به صورت مداوم انجام می‌شود تا حدودی قابلیت پیگیری تا نیازمندی وجود دارد. ولی از سوی دیگر چون برنامه‌ریزی و فعالیت‌هایی که انجام می‌شوند لزوماً مبتنی بر نیازمندی نیستند (مثلاً blitz planning مبتنی بر تسک است). پیگیری تا نیازمندی با مشکل مواجه می‌شود. | قابلیت پیگیری تا نیازمندی | |

۲-۶-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- محصولات در هر دو متدولوژی آزمون‌پذیر هستند.
- محصولات هم در Crystal Clear و هم در DAD هم برای کاربر و هم برای تیم ایجاد ملموس هستند.
- در هر دو متدولوژی قابلیت پیگیری تا نیازمندی وجود دارد.

نتیجه‌گیری: از نظر آزمون‌پذیری محصولات Crystal Clear بهتر از DAD است. از نظر ملموس بودن محصولات هر دو متدولوژی تقریباً در یک سطح قرار دارند. قابلیت پیگیری تا نیازمندی در DAD بیشتر از Crystal Clear است.

۷-۲- تشویق به مشارکت فعالانه کاربر

۱-۷-۲- بررسی متدولوژی‌ها

| | |
|--|--|
| <p>در این متدولوژی مشارکت کاربر بسیار مشهود است. در این متدولوژی تیم باید با مشتری به صورت مداوم و از نزدیک در ارتباط باشد تا نیازمندی‌ها را به خوبی تشخیص دهد و مرتباً با گرفتن بازخورد از او کیفیت را ارتقاء ببخشد. نقش product owner برای نماینده مشتری در تیم اصلی برای همین منظور در نظر گرفته شده است.</p> | <p>مشارکت فعالانه کاربر</p> <p>DAD</p> |
| <p>نماینده مشتری در قالب نقش ambassador user همواره در کنار تیم ایجاد حضور دارد و به عنوان متخصص در استفاده از سیستم به تیم کمک می‌کند.</p> <p>تامین کننده اجرایی نیز از طرف مشتری در تیم حضور دارد.</p> <p>در جریان هر cyclic delivery سیستمی که یکپارچه سازی شده است به تعداد کمی از کاربران تحویل داده می‌شود. از بازخوردی که کاربران می‌دهند برای بهتر کردن سیستم و بازبینی برنامه‌ها و نیازمندی‌ها استفاده می‌شود.</p> <p>مشاهده می‌شود که سطح مشارکت مشتری در این متدولوژی بسیار بالاست.</p> | <p>مشارکت فعالانه کاربر</p> <p>Crystal Clear</p> |

۲-۷-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- در هر دو متدولوژی مشتری در قالب یک نقش در کنار تیم ایجاد حضور دارد.
- در هر دو متدولوژی پیوسته از مشتری بازخورد گرفته می‌شود.

نتیجه‌گیری: در این دو متدولوژی مانند سایر متدولوژی‌های چابک مشارکت کاربر در سطح بالایی قرار دارد و نمی‌توان یکی را بر دیگری برتر دانست.

۸-۲- قابلیت اجرا و کارایی

۸-۲-۱- بررسی متدولوژی‌ها

| | | |
|--|-------------|---------------|
| فرایند این متدولوژی پیچیده نیست و از قابلیت اجرای بالایی برخوردار است. | قابلیت اجرا | |
| <p>از مواردی که به کارایی این متدولوژی کمک می‌کنند می‌توان به موارد زیر اشاره کرد:</p> <ul style="list-style-type: none"> ▪ وابسته نبودن به ابزار خاص ▪ عدم توجه ایجادکنندگان به جزئیاتی که آن‌ها را از فعالیت‌های اصلی دور می‌کند. ▪ مدیریت پروژه قوی ▪ طراحی معماری زودهنگام ▪ عدم وابستگی به تکنیک‌های خطاخیز | کارایی | DAD |
| رنگ متدولوژی‌های خانواده crystal نشان دهنده میزان پیچیدگی نسبی آن‌ها است. Crystal Clear همان طور که از نام آن مشخص است کمترین پیچیدگی را در بین اعضای این خانواده دارد. همین مسئله باعث افزایش قابلیت اجرای آن می‌شود. | قابلیت اجرا | |
| <p>از مواردی که به کارایی این متدولوژی کمک می‌کنند می‌توان به موارد زیر اشاره کرد:</p> <ul style="list-style-type: none"> ▪ پیچیدگی کم واحدهای فرایند ▪ وابسته نبودن به ابزار خاص ▪ عدم توجه ایجادکنندگان به جزئیاتی که آن‌ها را از فعالیت‌های اصلی دور می‌کند. ▪ مدیریت پروژه مناسب <p>مواردی که باعث آسیب زده به کارایی متدولوژی می‌شوند:</p> <ul style="list-style-type: none"> ▪ تکیه بیش از اندازه به ارتباطات بین انسانی ▪ عدم پشتیبانی از تیم‌های توزیع شده | کارایی | Crystal Clear |

۸-۲-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- واحدهای فرایندی در هر دو متدولوژی پیچیده نیستند.
- هیچ کدام از متدولوژی‌ها به ابزار خاصی وابسته نیستند.
- هر دو متدولوژی از مکانیزم‌هایی برای مدیریت پروژه استفاده می‌کنند.

نتیجه گیری: هر دو متدولوژی از قابلیت اجرای مناسبی برخوردارند. از نظر کارایی نیز DAD برتری محسوسی نسبت به Crystal Clear دارد.

۹-۲- مدیریت پیچیدگی فرایند

۹-۲-۱- بررسی متدولوژی‌ها

| | | |
|---|-----------------------|----------------------|
| <p>فازهای متدولوژی به صورت زیر است:</p> <p>۱- Inception ۲- Construction ۳- Transition</p> <p>هر یک از فازها خود شامل سه مرحله زیر است:</p> <p>۱- Coordinate ۲- Collaborate ۳- Conclude</p> <p>با شکستن فرایند به چندین فاز و شکستن هر فاز به چندین مرحله پیچیدگی مدیریت شده است.</p> | <p>مدیریت پیچیدگی</p> | <p>DAD</p> |
| <p>فرایند این متدولوژی از سه فاز chartering، cyclic delivery و wra up تشکیل شده است. فاز cyclic delivery از چند delivery تشکیل شده است. هر delivery از چند iteration تشکیل شده است. هر iteration در چند روز و در هر روز چند integration انجام می‌شود. هر integration نیز از چند episode تشکیل شده است. در هر episode طراحی، برنامه‌نویسی و آزمون واحد انجام می‌شود. مشاهده می‌شود که این متدولوژی با partitioning و layering پیچیدگی فرایند را مدیریت کرده است.</p> | <p>مدیریت پیچیدگی</p> | <p>Crystal Clear</p> |

۹-۲-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- هر دو متدولوژی با شکستن فرایند به بخش‌های کوچک‌تر سعی کرده‌اند پیچیدگی واحدهای فرایندی را کاهش دهند.

- تقسیم فرایند به چند iteration در جهت مدیریت پیچیدگی در هر دو متدولوژی مشاهده می‌شود.

نتیجه‌گیری: در هر دو متدولوژی پیچیدگی تا حدی مدیریت شده است.

۲-۱۰- توسعه پذیری، مقیاس پذیری، قابلیت پیکربندی و انعطاف پذیری

۲-۱۰-۱- بررسی متدولوژی‌ها

| | | |
|---|-----------------|-----|
| هیچ صحبتی از هسته قابل توسعه و نقاط توسعه نشده است در نتیجه می‌توان گفت توسعه پذیر نیست. | توسعه پذیری | DAD |
| مقیاس پذیری را از دو منظر می‌توان بررسی کرد: ۱- سطح بحرانی: بدلیل عدم پوشش فرمالیسم مناسب پروژه‌های life critical نیست ولی مشکلی برای استفاده در سایر سطوح بحرانی ندارد. ۲- اندازه پروژه: تیم‌ها در DAD کوچک یا متوسط هستند. تیم‌هایی که از کمتر از ۱۵ نفر تشکیل شده‌اند تیم کوچک و تیم‌هایی که بین ۱۰ تا ۴۰ عضو دارند تیم متوسط در نظر گرفته می‌شوند. علاوه بر این چون در زمان ساخت این متدولوژی نگرانی‌های سازمانی نیز در نظر گرفته شده است می‌توان گفت این متدولوژی تا سطح سازمانی مقیاس پذیر است. | مقیاس پذیری | |
| این متدولوژی یک چهارچوب سطح بالاست که جزئیات آن با توجه به پروژه مشخص می‌شوند. همان طور که مشاهده می‌شود این متدولوژی نه تنها قابل پیکربندی است بلکه پیکربندی آن ضروری است. | قابلیت پیکربندی | |
| در انتهای فاز construction و در مرحله conclude با برگزاری جلسات retrospective فرایند مورد مرور و بازبینی قرار می‌گیرد و در صورت نیاز تغییراتی در آن اعمال می‌شود. چون فرایند این متدولوژی به صورت چرخشی است می‌توانیم از تجربیاتی که در چرخه‌های گذشته یاد گرفته‌ایم برای اصلاح فرایند در چرخه‌های آینده استفاده کنیم. | انعطاف پذیری | |
| این متدولوژی یک چهارچوب فرایندی سطح بالا برای ایجاد نرم‌افزار را اجبار می‌کند. استفاده از تعدادی از عناصر فرایندی سطح بالا و عمومی نیز در آن ضروری است. هم چنین بعضی از محصولات خاص نیز باید در آن ساخته شوند. با این حال بسیاری از جزئیات به تیم ایجاد سپرده شده است تا با توجه به پروژه در مورد آن‌ها تصمیم گرفته شود. به ایجادکنندگان حتی این اجازه داده می‌شود که از تکنیک‌های موجود در سایر متدولوژی‌ها نیز استفاده کنند. هم چنین سایر اعضای خانواده متدولوژی‌های crystal برای پروژه‌هایی با اندازه و سطح‌های حساسیت دیگر نیز تعریف شده‌اند که می‌توان از آن‌ها نیز استفاده کرد. بنابراین این متدولوژی توسعه پذیر است. | توسعه پذیری | |
| مقیاس پذیری را از دو منظر می‌توان بررسی کرد: ۱- سطح بحرانی: به صراحت ذکر شده است که تنها مناسب سطوح بحرانی C و D است و برای سایر سطوح بحرانی قابل استفاده نیست. در سطح بحرانی C در صورت بروز خطا در سیستم | مقیاس پذیری | |

| | | |
|--|-------------------------|--|
| <p>راحتی کاربر با مشکل مواجه می‌شود. در سطح بحرانیت D در صورت بروز خطا در سیستم پول در اختیار سازمان با خطر مواجه می‌شود.</p> <p>۲- اندازه پروژه: حداکثر تعداد افرادی است که به صورت همزمان در پروژه مشغول به کار هستند در این متدولوژی ۶ نفر است. در نتیجه مناسب پروژه‌های بزرگ نیست.</p> | | |
| <p>هر کدام از اعضای خانواده crystal یک چهارچوب فرایندی سطح بالا برای ایجاد نرم‌افزار اجبار می‌کنند. استفاده از تعدادی از عناصر فرایندی سطح بالا و عمومی نیز در آن‌ها ضروری است. هم چنین بعضی از محصولات خاص نیز باید در آن‌ها ساخته شوند. با این حال بسیاری از جزئیات به تیم ایجاد سپرده شده است تا در مورد آن‌ها تصمیم گرفته شود. به ایجادکنندگان حتی این اجازه داده می‌شود که از تکنیک‌های موجود در سایر متدولوژی‌ها نیز استفاده کنند.</p> <p>در فاز chartering یکی از کارهایی که انجام می‌شود شکل دهی و دقیق سازی متدولوژی است.</p> <p>با توجه به نکاتی که بیان شد می‌توان گفت قابلیت پیکر بندی این متدولوژی بسیار بالاست.</p> | <p>قابلیت پیکر بندی</p> | |
| <p>در خانواده متدولوژی‌های crystal به طور کلی به خاطر مرور و بازنگری فرایند ایجاد به صورت مداوم در قالب کارگاه‌های تفکر (reflectio workshop) سطح بالایی از انعطاف پذیری وجود دارد.</p> <p>تیم ایجاد یک متدولوژی پایه به شکل مجموعه کمینه‌ای از قراردادهای کاری را در شروع پروژه انتخاب می‌کند و در طول پروژه با توجه به تجربیاتی که در طول iteration بدست آورده‌اند در انتهای هر iteration فرایند را مورد مرور و بازنگری قرار می‌دهند و مجموعه قراردادهای کاری را به مرور کامل می‌کنند.</p> <p>به طور خاص در Crystal Clear در فاز cyclic delivery در انتهای هر چرخه تحویل reflection workshop برگزار می‌شود و متدولوژی مورد مرور و بازنگری قرار می‌گیرد. هم چنین هر چرخه تحویل از یک یا چند iteration ساخته شده است که در انتهای هر iteration نیز یک reflection workshop برگزار می‌شود. چرخشی بودن فرایند این متدولوژی به انعطاف پذیری آن کمک شایانی می‌کند. در کل می‌توان گفت انعطاف پذیری این متدولوژی بسیار بالاست.</p> | <p>انعطاف پذیری</p> | |

۲-۱۰-۲- مقایسه متدولوژی‌ها

شباهت‌ها و تفاوت‌ها

- هر دو متدولوژی قابل استفاده در پروژه‌هایی با سطح بحرانیت C و D هستند.
- از DAD برخلاف Crystal Clear می‌توان در پروژه‌هایی با سطح بحرانیت E استفاده کرد.

- هر دو متدولوژی مناسب پروژه‌های life critical نیستند.
 - سقف اندازه تیم‌ها در Crystal Clear شش و در DAD چهل است.
 - هر دو متدولوژی یک چهارچوب سطح بالا محسوب می‌شوند و باید متناسب با پروژه پیکربندی شوند.
 - در هر دو متدولوژی جلساتی بین اعضای تیم برای بازبینی فرایند متدولوژی در نظر گرفته شده است.
 - در هر دو متدولوژی از تجربیاتی که در چرخه‌های قبلی بدست آمده است برای بهبود فرایند استفاده می‌شود.
- نتیجه‌گیری:** توسعه‌پذیری Crystal Clear بیشتر از DAD است ولی DAD مقیاس‌پذیری بیشتری دارد. قابلیت پیکربندی و انعطاف‌پذیری در هر دو متدولوژی در سطح خوبی قرار دارند.

| | | |
|---|-------------|---------------|
| <p>چون فاقد فرمالیسم و مدل‌سازی است مناسب پروژه‌های life critical نیست. ولی برای سایر سطوح حساسیت مناسب است. چون امکان تغییر نیازمندی‌ها در میانه فرایند را دارد برای پروژه‌هایی با نیازمندی‌های متغیر مناسب است. چون این متدولوژی قابل سفارشی‌سازی است حوزه کاربرد گسترده‌ای دارد. این متدولوژی enterprise aware است یعنی به دغدغه‌های سازمانی توجه دارد و در سطح سازمانی نیز می‌توان از آن استفاده کرد.</p> | حوزه کاربرد | DAD |
| <p>در Crystal Clear پروژه‌های C6 و D6 هدف قرار داده شده‌اند. C و D نشان دهنده سطوح بحرانی هستند. در سطح بحرانی C در صورت بروز خطا در سیستم راحتی کاربر با مشکل مواجه می‌شود. در سطح بحرانی D در صورت بروز خطا در سیستم پول در اختیار سازمان با خطر مواجه می‌شود. ۶ نیز نشان دهنده حداکثر تعداد افرادی است که به صورت همزمان در پروژه مشغول به کار هستند. در نتیجه مناسب پروژه‌هایی در سایر سطوح بحرانی و با تعداد نفرات بیشتر نیست. برای این گونه از پروژه‌ها می‌توان از سایر اعضای خانواده crystal استفاده کرد. بدلیل این که فاقد مکانیزم نگهداری است مناسب پروژه‌هایی که نگهداری در آن‌ها مهم است نیست. مگر این که از مکانیزم‌های رایج نگهداری در کنار آن استفاده شود. چون امکان تغییر نیازمندی‌ها در میانه اجرای فرایند را می‌دهد برای پروژه‌هایی با نیازمندی‌های متغیر مناسب است.</p> | حوزه کاربرد | Crystal Clear |

شباهت‌ها و تفاوت‌ها

- هر دو متدولوژی از فرمالیسم پشتیبانی نمی‌کنند و مانند سایر متدولوژی‌های چابک مناسب پروژه‌های life critical نیستند.
- در پروژه‌هایی با سطح بحرانی E که در آن‌ها در صورت شکست پروژه سرمایه شرکت از بین می‌رود و شرکت ورشکسته خواهد شد می‌توان از DAD استفاده کرد ولی از Crystal Clear نمی‌توان استفاده کرد.
- از هر دو متدولوژی می‌توان در پروژه‌هایی با سطح بحرانی C و D استفاده کرد.
- هر دو متدولوژی مناسب پروژه‌هایی با نیازمندی‌های متغیر هستند.
- DAD به خاطر توجه‌ای که به دغدغه‌های سازمانی دارد مناسب پروژه‌هایی در سطح سازمان است ولی Crystal Clear مناسب پروژه‌های سازمانی نیست.
- DAD قابل استفاده در پروژه‌هایی با تعداد افراد زیاد است در حالی که Crystal Clear مناسب پروژه‌هایی با تعداد افراد کم است.

- در پروژه‌هایی که دغدغه نگهداری بیشتر است DAD مناسب‌تر از Crystal Clear است.

نتیجه‌گیری: DAD حوزه کاربرد بسیار گسترده‌تری از Crystal Clear دارد.