

بسم الله الرحمن الرحيم

تمرین سوم درس متدولوژی

استاد درس: دکتر رامسین

علی ابریشمی

۹۷۳۰۱۷۷۵

پاییز ۹۸

سوال اول) مقایسه متدولوژی DAD با متدولوژی های XP، FDD و DSDM

معیار اول) تعریف شفاف و دقیق متدولوژی

متدولوژی DAD

متدولوژی DAD جزو دسته متدولوژی های چابک محسوب می شود. این متدولوژی تمرکز خود را بر روی نیروی انسانی و راندمان حداکثری آن گذاشته است. اولین و مهم ترین هدف در این متدولوژی ساخت سریع سیستم است که در زمان کمینه آن را بتوان به مشتری ارائه داد. فازهای اصلی این متدولوژی عبارتند از Inception، Construction و Transition. در فاز اول که Inception نام دارد، در ابتدا تیم اولیه تشکیل می شود. سپس چشم انداز پروژه در آینده تبیین می گردد که این کار به وسیله ذی نفعان صورت می گیرد و همگی بر آن توافق می کنند. سپس استراتژی فنی کار به همراه نیازمندی ها مشخص شده و برنامه ریزی اولیه ای برای ادامه کار و فازهای آتی صورت می گیرد. در انتهای این فاز یکی از مهم ترین کارها برای موفقیت پروژه که عبارت است از شناسایی ریسک ها انجام می گیرد.

فاز دوم یعنی Construction در طی iterationهایی به کدزنی و ساخت محصول می پردازد. در طی iterationها به صورت آرام و مستمر محصول ساخته شده و بهبود می یابد. نیازهای مشتری در این میان دقیق تر شده و فیدبک ها در محصول اعمال می شود. به ترتیب که iterationها انجام می شود، محصول به محصول نهایی قابل دیپلوی نزدیک شده و کیفیت آن افزایش می یابد. در فاز انتهایی پروژه یعنی Transition اعتبارسنجی و تست سیستم انجام می شود تا همان محصول مدنظر کارفرما و مشتری تولید شده باشد. در صورت تایید مشتری، تحویل محصول صورت گرفته و در محیط اجرایی دیپلوی می شود.

هرکدام از فازهای متدولوژی DAD خود شامل سه بخش می شود. در بخش اول که Coordinate یا هماهنگی نام دارد، کارهای ابتدایی برای شروع و آغاز آن فاز صورت می گیرند. این کارها برای مثال شامل برنامه ریزی فاز و هماهنگی اعضا و وظایفشان می گردد. بخش دوم که Collaborate یا همکاری نام دارد، کارهای اصلی که هدف آن فاز می باشد و توضیح داده شد، انجام می گیرند. بخش سوم نیز Conclude نام دارد که تکمیل و جمع بندی کارها برای انتقال به مشتری و آغاز فعالیت های فاز بعد در آن انجام می گیرد.

در متدولوژی DAD تیم ها به دو دسته نقش های اولیه و ثانویه شکسته می شود. نقش های اولیه شامل رهبر تیم، صاحب محصول، صاحب معماری و نیروی توسعه می شود. همانگونه که مشهود است، این نقش ها به نوعی فنی می باشد و محصول اصلی را توسعه می دهند. نقش های ثانویه به نوعی افرادی هستند که دارای سواد تخصصی دامنه مسئله می باشند و به تست و اعتبارسنجی سیستم می پردازند. این دسته از نقش ها به این دلیل مهم اند که سیستم نهایی لازم است توسط اشخاصی

با سواد آن حوزه استفاده شود. برای مثال مشتری های نهایی یک سیستم پزشکی، افراد متخصص و با دانش در حوزه پزشکی می باشد. تعداد افراد تیم های DAD معمولاً کم هستند و نهایتاً تا ۴۰ نفر می توانند باشند. همچنین نیاز است از لحاظ جغرافیایی نزدیک به یکدیگر بوده و در یک ساختمان قرار داشته باشند. در صورتی که تعداد اعضا زیاد باشد، آنگاه تیم به زیرتیم هایی شکسته می شود که هر کدام دارای نقش های رهبر تیم، صاحب محصول، صاحب معماری، توسعه دهنده می باشد که با تجمیع محصولات توسعه داده شده، به محصول اصلی و نهایی دست توان یافت.

همانگونه که ذکر شد، این متدولوژی با دقت و جزئیات مبسوط شرح داده شده است و کتاب های متفاوتی طی ویرایش های مختلف به چاپ رسیده است. محصولاتی که در هر فاز لازم است تولید شوند، قواعد و قوانینی که اعضای تیم برای تولید محصول ملزم به رعایت شان هستند، فعالیت های چتری تولید سیستم و به صورت کلی تمام فعالیت های کاری لازمه در این متدولوژی بیان گردیده اند. همچنین نقش های متفاوتی که هر کدام از اعضا برعهده می گیرند نیز در این متدولوژی به دقت بیان شده است. پس در نتیجه می توان گفت متدولوژی DAD از منظر تعریف دقیق و شفاف جزو متدولوژی های قوی محسوب می شود که از این رو تیم استفاده کننده از آن دچار ابهام و دوراهی در طول تولید سیستم نگردد.

متدولوژی XP

متدولوژی XP توسط آقای بک در سال ۱۹۹۶ ارائه شد که انتشار آن باعث اوج گیری و شهرت متدولوژی های چابک گردید. متدولوژی XP از شش فاز اصلی تشکیل شده است که با جزئیات در این بخش بیان می کنیم. در فاز اول که فاز Exploration یا اکتشافی نامیده می شود، لیستی از نیازمندی های اولیه سیستم استخراج می شود. این نیازمندی ها از طریق *user story* یا داستان کاربر مستندسازی می گردند. همچنین در این فاز اعضای تشکیل دهنده تیم که شامل مربی، تیم توسعه، مسئول کسب و کار و مشتری است، تعیین شده و شروع به کار می کنند. در انتهای این فاز نیز معماری کلی سیستم از طریق چندین پروتوتایپ مشخص شده و به مشتریان ارائه می گردد. این معماری سطح بالا و کلی که متافور نامیده می شود، به عنوان مبنا برای تولید سیستم مورد استفاده قرار می گیرد.

فاز دوم به نام *Planning* یا برنامه ریزی با هدف ترسیم آینده راه انجام می گیرد. در این فاز به ترتیب تخمینی از میزان ساعات لازم برای انجام تسک ها صورت گرفته و سپس اولویت بندی کارها انجام می گیرد. در انتها یکی سری وظیفه برای انجام در *iteration* اول برگزیده می شود. فاز سوم *Iterations to first release* نام دارد. در این فاز ابتدا برنامه مد نظر در طی *iteration* آماده می شود. کارهای مدنظر انتخاب شده و هر عضو یک مجموعه از کارها را برای پیاده سازی انتخاب می کند. سپس شروع به توسعه محصول می شود و طی گام های تحلیل، طراحی، کدزنی، تست و یکپارچه سازی محصول آماده انتشار اولیه می گردد. فاز چهارم متدولوژی به نام *Productionizing* است که طی آن ابتدا عملیات V&V انجام شده و در صورت قبول شدن تست *Acceptance* سیستم به محیط عملیاتی دیپلوی می گردد. در طی عملیات دیپلوی لازم است درستی کار با دیگر سیستم ها نیز چک گردد.

فاز پنجم از متدولوژی Maintenance نام دارد که باقی **user story** ها انتخاب شده و پیاده‌سازی می‌شوند. در این فاز به نوعی فازهای دوم، سوم و چهارم به ترتیب و به صورت دوره‌ای اجرا می‌شوند. این چرخه تا جایی ادامه می‌یابد که یا **user story** ها به اتمام رسیده باشند یا اینکه تصمیم به بازنشستگی سیستم گرفته شود. فاز ششم و نهمی از متدولوژی به نام مرگ یا **Death** است که طی آن سیستم بازنشسته اعلام می‌شود و عملیات مستندسازی مرگ سیستم اعمال می‌گردد.

همانگونه که مشاهده می‌شود، این متدولوژی یک متدولوژی با جزئیات لازمه برای استفاده در یک تیم تولید نرم‌افزار است. تمامی فعالیت‌های کاری در آن تعریف شده و خروجی‌های هرگام به همراه **practice** های آن به صورت دقیقی بیان گردیده است. نکته منفی درباره این متدولوژی عدم بیان نقش‌های افراد تیم است. اما از طرفی دیگر تکنیک‌ها و قواعد لازمه برای کدزنی و همچنین مرتبط با فعالیت‌های چتری تولید نرم‌افزار به دقت و با جزئیات فراوان اشاره گردیده است. به همین دلیل این متدولوژی یکی از متدولوژی‌های پرتعداد و پراستفاده میان تیم‌های تولید نرم‌افزار گردیده است.

متدولوژی FDD

متدولوژی FDD در سال ۱۹۹۹ توسط آقایان دلوکا و کاد ارائه شد. هدف اصلی این متدولوژی ارائه مفهوم **Feature** (که از اینجا متن به بعد به آن ویژگی می‌گوییم) بود که به وسیله آن نیازمندی‌های کاربر مدل می‌شد. این متدولوژی به صورت **full lifecycle** نیست و بسیاری از فعالیت‌های آغاز و اتمام پروژه را دارا نیست که در بخش‌های بعدی به صورت مفصل بررسی می‌نماییم.

هر ویژگی به صورت یک عملکرد ریزدانه سیستم تعریف می‌گردد. برای مثال چک کردن موجودیت صندلی در یک پرواز. هر ویژگی در مدت زمان کمتر از دو هفته توسط تیم توسعه قابل پیاده‌سازی است. به چندین ویژگی که در راستای یک هدف قرار دارند، یک **Activity** یا فعالیت گفته می‌شود (البته با نام مجموعه ویژگی نیز بیان می‌گردد). مجموعه‌ای از فعالیت‌ها متعلق به یک **Area** یا حوزه می‌شوند. پس می‌توان این سه مفهوم ویژگی-فعالیت-ناحیه را به صورت زیربنای تولید سیستم در نظر گرفت.

در متدولوژی FDD فرایندها به ۵ فاز اصلی تقسیم می‌شوند که در ادامه بیان می‌کنیم. ابتدا یک مدل کلی از سیستم ترسیم می‌گردد. تیمی متشکل از افراد خبره دامنه و افراد فنی گرد هم جمع شده و با بررسی دامنه مسئله به مدلی برای پیاده‌سازی نرم‌افزاری می‌رسند. در این حین مستندات فنی مسئله خوانده شده، مجموعه‌ای از مدل‌ها در حوزه‌های مختلف ترسیم شده و با تجمیع‌سازی و تایید مدل‌ها به مدل اصلی می‌رسیم. در فاز دوم لیست ویژگی‌ها تعیین می‌گردد. تیمی متشکل از مدیر پروژه و برنامه‌نویسان اصلی دور هم جمع شده و تیم مدلسازی را تشکیل می‌دهند. لیست ویژگی‌ها که یک سلسله مراتب سه لایه‌ای است تشکیل می‌گردد. این سلسله مراتب عبارتند از: ناحیه‌ها-فعالیت‌ها-ویژگی‌ها. که برای ترسیم آن از روش بالا به پایین استفاده می‌شود. بدین صورت که ابتدا ناحیه‌ها مشخص می‌شود. سپس فعالیت‌های درون هر ناحیه تعیین می‌گردد و سپس تمام ویژگی‌های آن فعالیت مشخص می‌شود.

فاز سوم به نام برنامه ریزی پیرامون ویژگی است. در این فاز مدیر پروژه، برنامه نویسان اصلی و مدیر توسعه با همدیگر برنامه ریزی ترتیب پیاده سازی ویژگی ها را انجام می دهند. همچنین زمان پیاده سازی ویژگی را نیز مشخص می کنند. بعد از آن مجموعه ویژگی ها یا همان فعالیت ها را به برنامه نویسان اصلی تخصیص می دهند تا هر کدام به همراه تیمشان آن فعالیت را پیاده سازی نماید. بعد از آن هر برنامه نویس اصلی هر ویژگی را به یکی از توسعه دهندگان نسبت می دهد تا آن را پیاده سازی نماید.

بعد از برنامه ریزی پیاده سازی، در چندین iteration سیستم را پیاده سازی می کنیم. برنامه نویسان اصلی مسئولیت پیاده سازی مجموعه ویژگی یا فعالیت را دارند. این کار را با تخصیص ویژگی به افراد تیم انجام می دهند. در طول زمان پیاده سازی چندین iteration به صورت همزمان در حال انجام هستند. بعضی از اینها توسط یک برنامه نویس اصلی نظارت می شود. فاز چهارم و پنجم به صورت تکراری طبق موارد ذکر شده انجام میشوند.

فاز چهارم یا طراحی پیرامون ویژگی برای طراحی موارد قابل پیاده سازی است. ابتدا یک بررسی بر دامنه مسئله صورت می گیرد. مستندات موجود مطالعه می شود. نمودار توالی رسم شده که در آن نحوه ارتباط میان اشیا نمایش داده می شود. بعد از آن طراحی کلی از کلاس و متدهای آن صورت می گیرد. و در آخر نیز پکیج طراحی تولید می شود که شامل نمودار توالی، بهبودهای object model و محدودیت ها و فرضیات است.

فاز پنجم به نام ساخت پیرامون ویژگی است. در این گام پیاده سازی کلاس ها و متدهای آن انجام می شود. این پیاده سازی ها بایستی کاملاً منطبق بر پکیج طراحی باشد و وظیفه آن برعهده صاحب کلاس است. بعد از آن تست واحد کلاس ها و سپس تست کلی تر سیستم انجام می شود. بعد از آن نیز کد در محیط عملیاتی دیپلوی می گردد.

همانگونه که ذکر شد این متدولوژی به صورت کامل و صریح مراحل و فعالیت ها را ذکر کرده است. همچنین در هر بخش محصولات و مصنوعات را تشریح کرده تا دقیقاً مطابق استانداردها تولید گردد. همچنین این متدولوژی نقش افراد مختلف در توسعه سیستم را به صورت دقیق ذکر کرده تا هیچ کجا ابهامی وجود نداشته باشد. تنها مورد منفی آن است که این متدولوژی به صورت full lifecycle نیست و استفاده کنندگان از متدولوژی در برخی جاها دچار خلل در فعالیتها می شوند که لازم است خود فرایندهایی برای آن تشریح کنند.

متدولوژی DSDM

متدولوژی DSDM که یک متدولوژی چابک است، در سال ۱۹۹۴ ارائه گردید. هدف این متدولوژی توسعه سریع نرم افزار می باشد و امروزه استاندارد اصلی در این حوزه شده است. ویژگی مهم این متدولوژی قابلیت انعطاف پذیری و شخصی سازی است.

این متدولوژی از سه فاز اصلی تشکیل شده است. در فاز اول که pre-project نام دارد، کسب و کار بررسی شده و بر مبنای آن استارت کردن پروژه چک می گردد.

فاز دوم این متدولوژی با نام **project-proper** است. در این فاز چهار فعالیت اصلی صورت می‌گیرد. فعالیت اول تحلیل امکان‌سنجی پروژه است که آیا از دید تکنیکی پروژه قابل انجام است یا خیر. همچنین دید بیزینسی نیز بررسی می‌گردد تا امکان‌سنجی مالی نیز انجام شود. در این فعالیت ممکن است که پروژه متوقف شود و ادامه نیابد. فعالیت بعدی با هدف مشخص‌سازی کسب و کار پروژه انجام می‌شود. در این فعالیت حوزه و دامنه مسئله مشخص می‌گردد. فعالیت سوم از این فاز توسعه کد به صورت **iterative** است. به صورت **iterative** کد تولید می‌گردد تا مشخصات کسب و کاری تعیین شده را برآورده کند. در آخرین فعالیت این فاز نیز کد توسعه یافته دیپلوی می‌گردد. در این فعالیت محصول مرور می‌گردد تا استانداردها رعایت شده باشد و اعتبارسنجی سیستم انجام می‌گیرد. در آخر نیز پروژه دیپلوی می‌گردد.

فاز سوم نیز به نام **post-project** است که به صورت مجدد اهداف بیزینسی چک می‌گردد تا برآورده شدنشان قطعی گردد.

همانگونه که مشاهده می‌شود این متدولوژی به صورت شفاف و دقیق تمامی مراحل کار را توضیح داده و پوشانده است. این متدولوژی نقش‌های موجود در پروژه را نیز بیان نموده است که شامل افراد بیزینسی، فنی، فرایندی و مدیریتی می‌گردند. همچنین جزئیات فراتر مانند محصولات تولید شده و **practice** های لازم در طول پروژه به صورت کامل بیان گردیده است. در نتیجه این متدولوژی در این معیار امتیاز بالایی کسب می‌نماید.

مقایسه دو متدولوژی DAD و XP

هر دو متدولوژی دارای جزئیات فراوان و دستورالعمل‌های مشخص در رابطه با فعالیت‌های کاری توسعه نرم‌افزار هستند. از طرفی هر دو متدولوژی جزو متدولوژی‌های زنده و پر استفاده در میان تیم‌های نرم‌افزاری می‌باشند. مطالب بیان شده در آن‌ها به صورت صریح، شفاف، کامل، دقیق و جزئی به گونه‌ای بیان شده است که تیم تولیدکننده نرم‌افزار در هیچ جایی از تولید سیستم دچار مشکل نشوند.

اما اگر بخواهیم به صورت مقایسه‌ای یکی را بر دیگری برتری دهیم، انتخاب متدولوژی **XP** خواهد بود. با جستجو در اینترنت در رابطه با کتاب‌های منتشر شده درباره هر کدام از این دو متدولوژی به بیشتر بودن منابع در رابطه با **XP** خواهیم رسید. این متدولوژی به صورت مفصل‌تری در منابع اینترنتی بیان شده است و استفاده بیشتر تیم‌های تولید نرم‌افزار یکی از دلایل آن است. همچنین یکی دیگر از دلایل این رجحان وجود **practice** ها و **principle** هایی است که متدولوژی **XP** با خود آورده و بیان نموده است (به این موارد به صورت مفصل در معیارهای بعدی اشاره خواهد شد)

مقایسه دو متدولوژی DAD و FDD

هر دو متدولوژی به صورت دقیق و کاملی فعالیتها و واحدهای کاری را تشریح کرده اند. اما متدولوژی FDD به دلیل full lifecycle نبودن دارای اشکال است. تیم توسعه دهنده در برخی گام ها چون راه حلی در پیش ندارد، بایستی خود اقدام به اضافه کردن فرایند به متدولوژی کند. اما متدولوژی FDD دارای یک مزیت بسیار مهم نیز هست که به صورت دقیق مراحل پیاده سازی نیازمندی ها را به وسیله مفهوم ویژگی ارائه می دهد. در کل می توان گفت DAD به دلیل full lifecycle بودن تعریف شفاف تر و دقیقتری دارد و تیم نرم افزاری در استفاده از آن راحت تر خواهد بود.

مقایسه دو متدولوژی DAD و DSDM

متدولوژی DSDM به دلیل پوشش practice ها و نقش ها به صورت کامل تر نسبت به DAD انتخاب اول خواهد بود. DSDM محصولات و نقش ها را نیز به صورت دقیق بیان کرده که بایستی در طول پروژه رعایت گردد. همچنین این متدولوژی امروزه استاندارد اصلی در حوزه توسعه سریع نرم افزار است که باعث گردیده اقبال به آن بسیار زیاد گردد و به همین دلیل تمام فرایندها و محصولات به صورت دقیق تر در دسترس باشد.

معیار دوم) پوشش فعالیت‌های عمومی چرخه توسعه نرم‌افزار

متدولوژی DAD

متدولوژی DAD یک متدولوژی full lifecycle می‌باشد که تمامی فعالیت‌های چرخه تولید نرم‌افزار را پوشش می‌دهد. در فاز Inception چشم‌انداز اولیه از پروژه تهیه می‌شود، نیازمندی‌های سیستم مشخص شده و ریسک‌هایی که در پیش روی پروژه هست، استخراج می‌شود. همچنین تحلیل امکان‌سنجی در همین فاز انجام می‌شود. به نوعی می‌توان گفت این فاز که اولین فاز از پروژه است، تمامی فعالیت‌های مرتبط با تعریف و شروع پروژه را پوشش می‌دهد.

در فاز دوم یعنی Construction فعالیت‌های مرتبط با توسعه و ساخت سیستم صورت می‌گیرند. در این فاز معماری سیستم تعیین می‌گردد تا در طی iteration ها سیستم مطابق با آن تولید شود. نقش معمار که یکی از نقش‌های اصلی در این متدولوژی است، وظیفه مدیریت آن را برعهده دارد و در طول زمان پروژه مطابقت با معماری را به صورت دقیق رصد می‌کند. سپس طی iteration های متوالی فعالیت‌های ساخت، تست و دیپلوی به صورت مرتب انجام می‌گیرند. متدولوژی DAD شامل یک سری practice است که تست‌های مختلف و در سطوح متفاوت نوشته و اجرا می‌شوند. از جمله می‌توان به TDD ، ATDD و تست موازی اشاره کرد. همچنین در این فاز فعالیت‌های یکپارچه سازی و دیپلوی به صورت دوره‌ای اجرا می‌گردد.

دیپلوی نهایی سیستم برای استفاده مشتریان در طی گام Transition انجام می‌شود. تست‌های نهایی نیز در این گام صورت می‌گیرد تا پس از آن آموزش مشتری و تحویل سیستم صورت گیرد. در این متدولوژی اما فعالیت maintenance به صورت مجزا اشاره نشده است و لازم است تیم توسعه با شروع یک فرایند خود این کار را به متدولوژی اضافه نمایند.

متدولوژی XP

متدولوژی XP یک متدولوژی full lifecycle می‌باشد. در این متدولوژی ابتدا در فاز اول نیازمندی‌ها استخراج شده و تحلیل امکان‌سنجی سیستم از طریق پروتوتایپینگ که در این متدولوژی به نام Spiking شناخته می‌شود، صورت می‌گیرد. همچنین معماری کلی سیستم در این فاز مشخص می‌گردد. در فازهای دوم تا چهارم که به نوعی موتور توسعه نامیده می‌شوند، مراحل کدزنی، تست و دیپلوی کردن سیستم به صورت تکراری و دوره‌ای انجام می‌شوند. این چرخه تا اتمام تسک‌ها ادامه خواهد یافت. گام مهم maintenance نیز یک فاز از پروژه را تشکیل می‌دهد. در انتها نیز یک فاز به نام مرگ وجود دارد که بازنشسته کردن سیستم و فعالیت‌های پس از مرگ در آن صورت می‌گیرد.

متدولوژی FDD

این متدولوژی به عنوان یک متدولوژی full lifecycle محسوب نمی‌شود. پروژه هنگامی شروع می‌شود که تحلیل امکان‌سنجی اولیه از قبل انجام شده باشد. همچنین برنامه‌ریزی انجام شده و توافقات تیم ذی‌نفع از قبل حاصل شده باشد. همچنین این متدولوژی شامل فعالیت‌های پس از دیپلوی نمی‌گردد. برای مثال اعتبارسنجی سیستم و درستی آزمایشی آن جزو متدولوژی محسوب نمی‌شود. فعالیت‌های دیپلوی نهایی و maintenance نیز در ذیل آن جا نمی‌گیرد. اما باقی فعالیت‌های توسعه نرم‌افزار یعنی استخراج و مدلسازی نیازمندی‌ها، طراحی منطقی و معماری، کدزنی و تست به صورت خوب و دقیقی در آن ذکر شده است.

متدولوژی DSDM

متدولوژی DSDM یک متدولوژی full lifecycle است. فعالیت تحلیل امکان‌سنجی و استخراج نیازمندی‌ها در دو فاز project-proper به صورت کامل و دقیق انجام می‌شود. همچنین این متدولوژی یک گام فراتر رفته و اهداف بیزینسی را به صورت جزئی بررسی می‌کند و تطابق سیستم توسعه یافته را با آن می‌سنجد. این متدولوژی در محصولات خود معماری جزئی و در دامنه راه‌حل را نیز تبیین می‌کند. بعد از تعیین معماری اصلی سیستم، مراحل کدزنی، تست و دیپلوی نیز ذیل همین فاز انجام می‌گیرد. اما از فعالیت maintenance مطالبی بیان نشده است.

مقایسه دو متدولوژی DAD و XP

هر دو متدولوژی جزو متدولوژی‌های full lifecycle قرار می‌گیرند که به صورت حدوداً کاملی فعالیت‌های چرخه تولید نرم‌افزار را پوشش می‌دهند. متدولوژی DAD تحلیل امکان‌سنجی را به صورت کامل تری انجام می‌دهد و در XP تنها از طریق Spiking به تحلیل امکان‌سنجی پرداخته می‌شود. اما XP فازهای توسعه و تست را به صورت کامل تری پوشش داده و جزئیات بسیار بیشتری درباره practice های آن بیان می‌نماید. همچنین فاز maintenance آن باعث می‌گردد که پس از ساخت سیستم نیز امکان بهبود و تکمیل سیستم وجود داشته باشد. در انتها نیز فاز مرگ قرار داده است تا درس‌های کسب شده از تولید سیستم مستند شده و در آینده مورد استفاده قرار گیرد.

در نتیجه به صورت کلی می‌توان گفت که XP به نسبت از DAD در پوشش فعالیت‌های تولید نرم‌افزار کامل‌تر می‌باشد.

مقایسه دو متدولوژی DAD و FDD

همانگونه که بررسی شد متدولوژی FDD بسیاری از فعالیت‌های مهم ذیل آغاز و اتمام پروژه را شامل نمی‌شود. همین مورد باعث شده است که یک متدولوژی full lifecycle لقب نگیرد. از این رو متدولوژی DAD از آن برتر بوده و به عنوان یک متدولوژی full lifecycle قابل استفاده است.

مقایسه دو متدولوژی DAD و DSDM

متدولوژی DSDM یک متدولوژی کامل با انواع جزئیات از شروع تحلیل امکان سنجی پروژه تا تحلیل موارد کسب و کار و سپس کدزنی و دیپلوی آن است. همانگونه که می‌دانیم DAD نیز این فعالیت‌ها را شامل می‌شود اما تمرکز بهتر و بیشتر DSDM بر محصولات تولیدی، نقش‌های افراد تیم و practice های لازمه برای انجام باعث می‌گردد که DSDM به نسبت DAD برتری یابد.

معیار سوم) پشتیبانی از فعالیت‌های چتری توسعه نرم افزار

متدولوژی DAD

برای مدیریت ریسک فعالیت‌هایی در این متدولوژی در نظر گرفته شده است. در ابتدا تحلیل امکان سنجی و مشخص‌سازی ریسک‌ها در فاز Inception باعث می‌شود تا در طول انجام پروژه ریسک موجود مینیمم شود. همچنین در برنامه‌ریزی‌ها جوری در نظر می‌گیرند تا ریسک‌ها به حداقل ضرر منجر شوند. در این متدولوژی پس از اتمام هر iteration کاربران و ذی‌نفعان جمع شده و نظر خود را درباره سیستم می‌دهند تا اعتبارسنجی سیستم انجام گیرد. همچنین integration به صورت پیوسته و تست‌های منظم باعث می‌شوند تا هم verification و هم validation سیستم به صورت مرتبط و مستمر صورت گیرد.

مدیریت پروژه در این متدولوژی به صورت پیوسته و در تمامی فازها به دقت صورت می‌گیرد. در ابتدای هر فاز بخش اول یعنی Coordinate برای هماهنگی و برنامه‌ریزی تمامی فعالیت‌های آن فاز انجام می‌گیرد. برای مثال برنامه‌ریزی کارها و تقسیم وظایف در این بخش انجام می‌شود. در بخش conclude نیز ارزیابی از موارد انجام شده صورت می‌گیرد تا مطابقت آن با برنامه چک شود. پس ریویو مستمر باعث می‌شود تا برنامه کیفیت خود را از دست ندهد. همچنین وجود نقش‌های مختلف و تیم‌های گوناگون در پروژه باعث می‌شود تا مدیریت کارها و وظایف به شدت دقیق گردد. وجود burndown chart در این متدولوژی نیز در همین جهت بوده تا مدیریت وظایف به خوبی صورت پذیرد.

مدیریت کیفیت نیز به دلیل وجود practice های متعدد مرتبط با کیفیت نرم‌افزار از قبیل تست‌های گوناگون و یکپارچه‌سازی پیوسته باعث می‌شود کیفیت محصول در حد اعلائی خود قرار گیرد. وجود فرایندهای iterative نیز باعث شده تا کیفیت از درجه بالاتری برخوردار شود. همچنین به دلیل قابلیت ردیابی به نیازمندی‌ها نیز کیفیت کارها و مصنوعات افزایش می‌یابد.

متدولوژی XP

فعالیت‌های چتری توسعه نرم‌افزار به صورت کاملی در این متدولوژی پوشش داده شده است. مدیریت ریسک به وسیله عملیات Spiking و آزمایش پروتوتایپ‌های مختلف از ابتدای پروژه صورت می‌گیرد. همچنین وجود برنامه ریزی طی گام‌های مختلف باعث می‌شود که پایش و کنترل پروژه بسیار بهتر صورت گیرد. همچنین در این متدولوژی همانند دیگر متدولوژی‌های چابک کاربر سفیر به صورت فعال در تیم وجود دارد و در صورتی که فیدبکی مدنظر باشد یا نیازمندی عوض شده باشد، به تیم گوشزد خواهد کرد. نکته مهم دیگر در مورد مدیریت ریسک انتشار سریع نسخه اولیه از سیستم است. در متدولوژی XP کدها به صورت collective code ownership در یک منبع یکسان جمع می‌شود و به همین دلیل یکپارچه‌سازی به صورت مستمر صورت می‌گیرد.

مدیریت پروژه در این متدولوژی به صورت کامل انجام می‌گیرد. برنامه‌ریزی در یک فاز جداگانه در ابتدای هر iteration انجام می‌گیرد در نتیجه زمان‌بندی و کنترل پروژه بسیار راحت‌تر صورت می‌گیرد. همچنین در جلسات انتهایی هر iteration فیدبک‌ها گرفته شده و اعمال می‌شوند.

مدیریت کیفیت در این متدولوژی در حد اعلا صورت می‌گیرد. بدین صورت که یک سری practice بیان شده است که تیم ملزم به انجام آن‌ها به صورت مرتبی می‌باشد. برخی از آنها عبارتند از: جلسات استنداپ روزانه، برنامه نویسی دونفره، ریفتورینگ، جابجایی افراد در قسمت‌های مختلف کد و استفاده از کارت‌های CRC. در نتیجه با استفاده و انجام این موارد کیفیت کد تولید شده بسیار بالا خواهد رفت. از طرفی دیگر به دلیل قابلیت ردیابی به نیازمندی‌ها و V&V پیوسته باعث شده است کیفیت ارتقا یابد.

متدولوژی FDD

مدیریت ریسک در این متدولوژی به دلیل نبود تحلیل امکان‌سنجی، برنامه‌ریزی مبتنی بر ریسک، دخیل نکردن کاربر و ... دچار نقایص عمده‌ای است. همچنین با وجود اینکه این متدولوژی دارای ذات iterative است، اما چون کاربر را دخیل نمی‌کند و سیستم به او دمو داده نمی‌شود، برخی از ریسک‌ها با قوت قبلی باقی می‌مانند. اما iterative incremental بودن آن، V&V بعد از هر iteration باعث می‌شود که ریسک‌ها کمی کمتر گردد. مدیریت پروژه در این متدولوژی دارای نقاط قوت و برخی نقاط ضعف است. به دلیل برنامه‌ریزی دقیق زمان هر ویژگی و ساختاربندی خوب پروژه، زمان‌بندی‌ها بسیار دقیق و مطمئن هستند. همچنین وجود سلسله مراتب نقش‌ها و همچنین مفهوم صاحب کلاس باعث می‌گردد که به خوبی مسئولیت‌ها توزیع شده باشد. اما یک نقص آن است که به جز پیاده‌سازی سیستم، دیگر فعالیت‌های جانبی دارای برنامه‌ریزی نیستند و رها مانده‌اند.

مدیریت کیفیت در این متدولوژی بسیار خوب انجام می‌گیرد. دلیل اصلی آن وجود خاصیت ردیابی به نیازمندی‌ها است. به همین دلیل مدیر تیم‌ها و صاحبان کلاس‌ها مسئولیت دارند که به نحو بسیار خوب و دقیق موارد خواسته شده را پیاده‌سازی نمایند.

در مجموع فعالیت‌های چتری در این متدولوژی دارای نقاط قوت و ضعف است و به صورت کامل پوشش داده نشده است.

متدولوژی DSDM

مدیریت ریسک در این متدولوژی از ابتدای پروژه یعنی در فاز pre=project آغاز شده و امکان‌سنجی کسب و کاری صورت می‌گیرد. سپس با فرایندهای iterative این کار به صورت مداوم صورت می‌گیرد. همچنین فعالیت‌های تحلیل امکان‌سنجی و foundation که در فاز دوم قرار دارند باعث می‌شود ریسک‌ها شناسایی شده و حذف گردد.

مدیریت پروژه نیز به دلیل برنامه‌ریزی و مستندسازی آن در محصولات متدولوژی انجام می‌گیرد. همچنین مدیریت کیفیت نیز به دلیل وجود **practice** های هشت گانه در این متدولوژی به خوبی صورت می‌گیرد. این موارد شامل کارهای زیر می‌شود. کنترل و پایش مداوم پروژه، مهندسی نیازمندی‌ها، ارتباط موثر میان افراد، مدلسازی و پروتوتایپینگ و ... در نتیجه همانگونه که بیان شد و مشاهده می‌شود، این متدولوژی در پوشش فعالیت‌های چتری مهندسی نرم‌افزار بسیار کامل و دقیق است.

مقایسه دو متدولوژی DAD و XP

از جنبه ارزیابی مدیریت ریسک می‌توان گفت متدولوژی **DAD** برتر است. دلیل نیز وجود فاز جداگانه **Inception** است که به صورت متمرکز همین هدف را دنبال کرده و ریسک‌ها را ارزیابی می‌نماید. در جنبه مدیریت پروژه و مدیریت کیفیت می‌توان گفت متدولوژی **XP** بسیار برتر است. دلیل آن وجود **practice** های ۱۲ گانه برای اعمال در حین کدزنی سیستم است.

مقایسه دو متدولوژی DAD و FDD

همانگونه که در بخش‌های ذیل هر دو متدولوژی بحث و بررسی شد، هرکدام دارای نقاط ضعف و قوت هستند. در بحث مدیریت ریسک **DAD** به دلیل وجود فاز **Inception** بسیار خوب و دقیق کار می‌کند. برای مثال تهیه سند چشم‌انداز و تحلیل امکان‌سنجی باعث می‌شود که ریسک‌ها بسیار کم گردد. مدیریت پروژه کلی در **DAD** بهتر انجام می‌شود. یعنی مدیریت زمان‌بندی‌ها، تحویل پروژه و ... بسیار خوب صورت می‌گیرد. مدیریت توسعه کد اما در **FDD** بسیار بسیار خوب می‌باشد. به دلیل آنکه ساختار بندی پروژه بسیار خوب صورت می‌گیرد و وجود مسئولیت‌ها باعث می‌شود که کیفیت در سطح بسیار بالایی قرار گیرد. مدیریت کیفیت در **FDD** بهتر است. دلیل این مورد آن است که قابلیت ردیابی به نیازمندی‌ها وجود دارد. همچنین وجود مفهوم ناحیه-فعالیت-ویژگی بسیار به کیفیت کد تولید شده کمک خواهد نمود.

مقایسه دو متدولوژی DAD و DSDM

در بخش مربوط به **DSDM** با جزئیات دقیق بیان شد که این متدولوژی دارای فرایندها و محصولاتی است که فعالیت‌های چتری یعنی مدیریت ریسک، مدیریت پروژه و مدیریت کیفیت را در حد اعلا پوشش می‌دهد. این موارد باعث شده اند که

این متدولوژی جزو متدولوژی های محبوب قرار بگیرد. متدولوژی DAD نیز این سه مورد را برآورده می سازد اما وجود جزئیات و فرایندهای بیشتر در DSDM باعث می شود که به انتخاب اول تبدیل شود.

معیار چهارم) بی درزی و همواری

متدولوژی DAD

همواری به دلیل وجود فازهای مختلف و همچنین وجود iteration در هرکدام از فازها به خوبی در این متدولوژی برقرار است. همچنین چون محصولات و مصنوعات جدیدی در طی فازها تولید نمی‌شود، خاصیت همواری یا همان smoothness در حد بالایی برقرار است. اما به دلیل مدلسازی نیازمندی‌ها از طریق use case خاصیت بی‌درزی یا seamlessness دچار ضربه و اشکال می‌شود. در نتیجه خاصیت بی‌درزی در این متدولوژی به صورت ناقص وجود خواهد داشت.

متدولوژی XP

این متدولوژی دارای خاصیت همواری میان فعالیت‌ها می‌باشد. دلیل آن نیز وجود ذات و ماهیت iterative incremental در آن است. فعالیت‌های به صورت آرام و هموار انجام می‌گیرند تا به مرور سیستم ساخته شود. اما بی‌درزی در این متدولوژی به دلیل دره میان نیازمندی‌ها و کدزنی وجود دارد. بعد از مدلسازی نیازمندی‌ها به وسیله داستان‌های کاربر سپس ساخت و کدزنی سیستم انجام می‌شود. این باعث می‌شود که دچار شیفت در پارادایم شویم و گذار بدون دره اتفاق نیافتد.

متدولوژی FDD

بی‌درزی در این متدولوژی به صورت کامل وجود دارد. دلیل آن است که مدلسازی به وسیله مفهوم ویژگی انجام می‌شود. سپس همین ویژگی‌ها به افراد تخصیص داده می‌شود و از روی این موارد کد برنامه تولید می‌شود. در این میان نمودار توالی نیز به نبود درز کمک شایانی می‌کند. همانگونه که می‌دانیم وجود user story و use case باعث به وجود آمدن درز می‌شد که این متدولوژی این مورد را برطرف کرده است. همچنین همواری نیز در این متدولوژی به دلیل وجود ذات iterative incremental در آن به صورت بسیار خوبی موجود است.

پس می‌توان این متدولوژی را در معیار بی‌درزی و همواری جزو بهترین متدولوژی‌ها دانست.

متدولوژی DSDM

متدولوژی DSDM همانند دیگر متدولوژی های چابک دارای همواری به دلیل وجود فرایندهای iterative incremental است اما بی درزی را به دلیل استفاده از user story و use case در مهندسی نیازمندی ها دچار اخلاص و ضعف می کند.

مقایسه دو متدولوژی DAD و XP

هر دو متدولوژی دارای خاصیت همواری میان فازها هستند اما هر دو دارای ضعف در بی درزی می باشند. دلیل آن است که در یکی نیازمندی ها به وسیله use case مدل می شود و در دیگری نیازمندی ها به وسیله داستان کاربر مدل می شود. در نتیجه هنگام تولید کد و دیگر فعالیت ها، دچار درز می شویم.

مقایسه دو متدولوژی DAD و FDD

همانگونه که اشاره شد به دلیل مفهوم ویژگی متدولوژی FDD دارای خاصیت بی درزی است. اما در DAD چون از use case بایستی کدها تولید شود، دچار درز می شویم. اما هر دو متدولوژی دارای خاصیت همواری می باشند. در مجموع در این معیار بدون هیچ شک و شبهه ای می توان گفت که FDD بسیار برتر است.

مقایسه دو متدولوژی DAD و DSDM

هر دو متدولوژی در خاصیت همواری قابل قبول هستند ولی به دلیل ماهیت استخراج نیازمندی ها و سپس کدزنی سیستم، دچار مشکل در بی درزی می شوند. پس نمی توان یکی را به صورت برتر از دیگری دانست.

معيار پنجم) محوريت بر نيازمندي هاي عملكردي و غير عملكردي

متدولوژی DAD

در اين متدولوژی نيازمندي ها به صورت سريع و در ابتدا استخراج شده و توسط use case مدل می شود. اين نيازمندي ها هم شامل موارد عملكردي و هم غير عملكردي می باشد. سپس برطبق اين نيازمندي ها فازها و فعاليت هاي بعدي پروژه انجام می شوند. اما نمی توان گفت که اين متدولوژی یک متدولوژی بر محور نيازمندي است.

نکته مهم ديگر در مورد اين متدولوژی امکان فيدبک کاربران و ذی نفعان و تغيير در نيازمندي ها است. پس از انجام هر iteration در طی جلسه کاربران و ذی نفعان نظرات خود را ارائه می دهند تا در صورت لزوم نيازمندي (هايی) تغيير يابند. به صورت کلی می توان گفت نيازمندي هاي ابتدایی که در فاز Inception استخراج می شوند و همچنين نيازمندي هاي ثانويه که پس از هر iteration استخراج می شوند، با اولويت و اهميت بالا در پيشرفت محصول و پروژه تاثير گذارند.

متدولوژی XP

در اولين فاز از متدولوژی که به نام Exploration است، نيازمندي هاي سيستم به صورت داستان کاربر استخراج می گردد. سپس در ادامه پروژه و در فازهاي بعدي هنگامي که فعاليت هاي تحليل، طراحی، کدزنی و تست صورت می گیرند، مبدا و نقطه ابتدایی همين داستان هاي کاربر استخراج شده می باشد. مورد مهم ديگر وجود متافور سيستم می باشد که در آن نيازمندي هاي غير عملكردي همانند سرعت، امنيت، قابليت تست و ... دیده شده و مدل سازی شده اند.

متدولوژی FDD

در اين متدولوژی نيازمندي ها به وسيله مفهوم ويژگی مدلسازی می شوند. سپس بر مبنای ويژگی ها، فعاليت ساخته شده و با استفاده از فعاليت، حوزه تشکيل می شود. در نتيجه تمامی مفاهيم و موجوديت ها برپايه ويژگی که همان نيازمندي سيستم است، بنا شده اند. پس می توان گفت اين متدولوژی به صورت نيازمندي محور است که بالاترين سطح را نشان می دهد. پس در مجموع اين متدولوژی را می توان یکی از کامل ترين متدولوژی ها در اين معيار يعنی بر مبنای نيازمندي ها دانست.

متدولوژی DSDM

نيازمندي ها در اين متدولوژی نقش محوری دارد. در فاز اول و اوایل فاز دوم، نيازمندي ها به دقت جمع آوری شده و ارتباطشان با مباحث کسب و کاری مدل می گردد. اين مسئله اهميت دادن اين متدولوژی به نيازمندي ها را بيان می کند.

همچنین این متدولوژی اجازه تکمیل شدن نیازمندی ها در طول انجام پروژه را می دهد. پس این متدولوژی در این معیار امتیاز خوبی کسب می کند.

مقایسه دو متدولوژی DAD و XP

هر دو متدولوژی به روش های مختلف نیازمندی ها را استخراج کرده و مدل می کنند. یکی به وسیله use case و دیگری به وسیله داستان کاربر. این نیازمندی ها در هر دو قابلیت تکمیل، اضافه، کم و تغییر را دارند. در نتیجه می توان گفت که هر دو در این معیار نمره بالایی کسب کرده و نمی توان یکی را بر دیگری رجحان داد.

مقایسه دو متدولوژی DAD و FDD

متدولوژی FDD در معیار محوریت بر نیازمندی یکی از برترین متدولوژی ها است. دلیل نیز وجود مفهوم ویژگی و بنا شدن ادامه فعالیت ها بر آن است. متدولوژی DAD نیز بر محور use case جلو می رود اما نقشی محوری همانند مفهوم ویژگی ندارد. پس می توان گفت FDD در این معیار برتر است.

مقایسه دو متدولوژی DAD و DSDM

هر دو متدولوژی نیازمندی ها را استخراج کرده و مدل می کنند. همچنین اجازه تکمیل شدن و تغییر در آنها را می دهند. اما هیچکدام نقطه قوت و متمایز کننده ای نسبت به دیگری ندارد و هر دو در یک سطح می باشند.

معيار ششم) قابل تست بودن، ملموس بودن و قابل ردیابی به نیازمندی ها

متدولوژی DAD

قابل تست بودن، ملموس بودن و ردگیری به نیامندی‌ها در متدولوژی DAD تا حد مناسبی وجود دارند. به دلیل کم بودن تعداد مصنوعات در این متدولوژی، قابلیت فهم و استفاده از آنها افزایش چشمگیری می‌یابد. همچنین در انتهای هر فاز کد و سیستم ساخته شده به کاربر دمو داده می‌شود تا فیدبک‌ها از وی اخذ شود. بدین صورت محصولات و مصنوعات هم برای کاربر و هم تیم توسعه دهنده کاملاً ملموس خواهند بود.

از طرفی دیگر نیازمندی‌ها ابتدا به صورت سطح بالا و سپس به صورت جزئی و دقیق مدلسازی می‌شوند و فعالیت‌های آتی اعم از کدزنی و ساخت سیستم به وسیله همین نیازمندی‌ها اعتبارسنجی می‌شوند. بدین صورت قابلیت ردیابی قابلیت‌های سیستم به نیازمندی‌ها وجود خواهد داشت.

در مجموع چون این متدولوژی بر نیازمندی‌های اولیه و سپس نیازمندی‌های کاربر تاکید و تمرکز ویژه‌ای دارد، در این معیار نمره قابل قبول و بالایی کسب می‌کند.

متدولوژی XP

همانگونه که به صورت مفصل در معیار قبلی توضیح داده شد نیازمندی‌ها در این متدولوژی به صورت داستان کاربر استخراج شده و مستند می‌گردد. این باعث می‌گردد که قابلیت ردیابی به نیازمندی در حد خوبی به وجود آید. از طرفی این متدولوژی همانند اسکرام تا جای ممکن جلوی مدلسازی را گرفته و تعداد مدل‌ها را کمینه نگه می‌دارد. لقب model-phobic به این متدولوژی نیز اطلاق می‌گردد. پس می‌توان گفت این متدولوژی از منظر قابلیت تست و ملموس بودن مصنوعات نمره خوبی کسب خواهد کرد. این متدولوژی بر تست نرم‌افزار و کد تاکید ویژه‌ای دارد و انواع مختلف تست‌ها را بر آن اعمال می‌نماید.

متدولوژی FDD

تمامی نمودارها و محصولات این متدولوژی برای توضیح و تشریح ویژگی‌های سامانه است. این متدولوژی از سه نمودار اصلی استفاده می‌کند. اولی object model است که به وسیله آن تمامی ویژگی‌های سامانه در قالب فعالیت و ناحیه ذکر شده است. دومی نمودارهای توالی است که نحوه ارتباط اشیا برای انجام کار را نشان می‌دهد. سومی نیز ویژگی‌ها است که هر کدام به یک مسئول کلاس تخصیص می‌یابد تا پیاده سازی نماید.

در نتیجه همانگونه که مشهود است تعداد نمودارها و محصولات بسیار کم است. همچنین این نمودارها یکدیگر را تکمیل کرده و باعث سردرگمی توسعه دهندگان نخواهد شد. پس قابلیت تست محصولات در این متدولوژی برقرار است.

این محصولات و مصنوعات چون به نوعی قابلیت های سیستم است، برای کاربر نیز بسیار ملموس خواهد بود. همچنین توسعه دهنده نیز می تواند به صورت مستقیم کد را از روی ویژگی ها تولید نماید. پس قابلیت ملموس بودن نیز در این متدولوژی وجود دارد.

سومین جنبه نیز قابلیت ردیابی به نیازمندی هاست که همانگونه که بیان شد مفهوم ویژگی باعث می شود که به صورت مستقیم بتوان هر قابلیت پیاده سازی شده را بتوان به نیازمندی مربوطه متصل کرد. پس در این جنبه نیز نمره بالایی کسب می نماید.

متدولوژی DSDM

هر سه ویژگی قابل تست بودن، ملموس بودن و ردیابی به نیازمندی ها در این متدولوژی برآورده شده است. دلیل نیز وجود نیازمندی ها به صورت محوری در این متدولوژی است. اما یک ایرادی که این متدولوژی دارد تعداد زیاد محصولات و مدل سازی ها به نسبت دیگر متدولوژی های چابک است که قابلیت تست را کاهش می دهد.

مقایسه دو متدولوژی DAD و XP

هر دو متدولوژی از منظر تولید مدل همانند هم هستند اما XP به صورت مدل فوبیک تر از تولید مدل و نمودار جلوگیری می نماید. از طرفی دیگر هر دو قابلیت ردیابی به نیازمندی ها را فراهم می نمایند. در کل نمی توان یکی را بر دیگری رجحان داده و انتخاب نمود. پس می توان گفت در این معیار هر دو در یک رده قرار دارند.

مقایسه دو متدولوژی DAD و FDD

در هر سه جنبه از این معیار می توان گفت متدولوژی FDD برتر است. دلیل اصلی نیز وجود مفهوم ویژگی است که قابلیت ردیابی و اتصال مستقیم سیستم تحت کار را به نیازمندی ها دارد.

مقایسه دو متدولوژی DAD و DSDM

هر دو متدولوژی در ویژگی ها ملموس بودن و ردیابی به نیازمندی دارای امتیاز خوب هستند. اما به دلیل تعداد محصولات زیاد در DSDM قابلیت تست پذیری آن کاهش می یابد. در نتیجه می توان DAD را در این معیار برتر دانست.

معيار هفتم) مشاركت فعال مشتری

متدولوژی DAD

مشتری از دیدگاه این متدولوژی بسیار حائز اهمیت می‌باشد. دلیل نیز آن است که در نقش‌های هم اولیه و هم ثانویه کاربران و متخصصان حوزه کاری دارای نقش هستند. این کاربران در طی هر iteration در مشاهده سیستم کمک کرده و فیدبک‌های خود را به تیم توسعه دهنده ارائه می‌کنند. جلسات ریویو نیز بر طبق همین مبنا تشکیل می‌شوند و نظرات کاربران از ایشان اخذ می‌گردد.

متدولوژی XP

در XP پس از انجام هر iteration کاربر دخیل شده و نظرات و بازخوردها را نسبت به محصول تولید شده می‌دهد. همچنین به دلیل قابلیت continuous integration و continuous deployment در این متدولوژی کاربر می‌تواند به صورت مداوم سیستم را استفاده و تست کند و اعتبارسنجی سیستم را انجام دهد.

متدولوژی FDD

این متدولوژی به صورت iterative incremental اجرا می‌گردد. با اینکه در این متدولوژی درباره دخیل کردن کاربر صحبتی به میان نیامده است و این متدولوژی تاکید و اهمیت اصلی را بر پیاده سازی قابلیت‌ها گذاشته است، اما می‌توان در انتهای هر iteration کاربر و مشتری را دعوت کرده تا نظرات از وی کسب گردد.

اما یک نکته که از نقاط مثبت این متدولوژی محسوب می‌گردد، وجود خبره مسئله در تیم توسعه است. خبره می‌تواند در مدلسازی و ساخت object model تاثیر بسیار زیادی گذاشته و تیم را در جهت مناسب هدایت نماید.

متدولوژی DSDM

کاربر سفیر در این متدولوژی نقش بسیار مهمی دارد. نقش‌های business analyst، business advisor، business ambassador جزو نقش‌هایی هستند که این متدولوژی برای دخیل کردن کاربر به پروژه قرار داده است. همچنین در انتهای iteration نیز دمو دادن به کاربران انجام می‌گیرد. در نتیجه نمره این معیار بالا خواهد بود.

مقایسه دو متدولوژی DAD و XP

هر دو متدولوژی کاربر را دخیل کرده و نظرات وی را در سیستم اعمال می دارند. دلیل نیز آن است که متدولوژی‌های چابک بسیار بر این موضوع تاکید داشته و اجازه تکامل نیازمندی‌ها را در طی تولید سیستم به کاربر می‌دهند.

اما اگر بخواهیم یکی را در این معیار متدولوژی برتر بنامیم، می‌توان DAD را انتخاب کرد. دلیل آن است که این متدولوژی یک نقش جداگانه برای کاربر در نظر گرفته و جزو نقش‌های اولیه محسوب می‌نماید. این نقش همیشه در کنار تیم وجود دارد و اعمال نظر می‌کند. اما نمی‌توان از این نکته غافل شد که متدولوژی XP نیز به خوبی کاربر را در فرایند دخیل کرده و بازخوردها را اعمال می‌کند.

مقایسه دو متدولوژی DAD و FDD

متدولوژی DAD با قرار دادن نقش جداگانه برای کاربر سفیر بسیار این مسئله را خوب در نظر داشته است. همچنین در انتهای هر iteration جلسه مرور با کاربر را قرار داده است تا نظرات وی دخیل گردد. اما متدولوژی FDD این نکات را مدنظر قرار نداده است. با این حال متدولوژی FDD دارای نکته ای است که در ابتدای مدلسازی از کاربر خبره مسئله استفاده نموده است که نکته قابل تاملی می‌باشد.

در مجموع می‌توان DAD را در این معیار برتر دانست.

مقایسه دو متدولوژی DAD و DSDM

همانگونه که بیان شد و نام برده شد، DSDM دارای نقش‌های متنوع برای کاربران سفیر میباشد. به همین دلیل یکی از برترین متدولوژی‌ها در دخیل کردن کاربر در پروژه است. در نتیجه متدولوژی DSDM بر DAD در این معیار برتری خواهد داشت.

معیار هشتم) practical و practicable بودن

متدولوژی DAD

قابلیت practicality و practicability در این متدولوژی در سطح بسیار بالایی برقرار است. این متدولوژی دارای سه فاز است که هر کدام گامی به سوی ساخت و تکمیل سیستم برمی‌دارند. نقش‌های مختلف افراد در این متدولوژی به خوبی از یکدیگر تفکیک شده و هر کدام گوشه‌ای از کار را برداشته و انجام می‌دهند. بدین صورت پیچیدگی واحدهای کاری تا حد بسیار خوبی شکسته می‌شود. گام‌های لازم در هر فاز به خوبی ذکر شده تا تیم از سردرگمی درآمده و به صورت مستقیم به هدف سیستم که نرم‌افزار اصلی می‌باشد برسند. همچنین استفاده از این متدولوژی نیاز به ابزار خاص و کمیابی ندارد تا تیم دچار مشکل گردد. از سویی دیگر به دلیل مدیریت پروژه مناسب و کامل در این متدولوژی، این ویژگی و معیار به خوبی برقرار می‌گردد.

متدولوژی XP

این متدولوژی دارای فعالیت‌های generic از چرخه تولید نرم‌افزار است. فعالیت‌های و فازها به گونه‌ای ساخته شده‌اند که نیازمندی به ابزار خاصی برای آنها نباشد. تنها موردی که کمی چالش برانگیز است محیط collective code ownership است که امروزه با شیوع استفاده از نرم‌افزار گیت این نیاز به سادگی برطرف خواهد شد.

برای مدیریت داستان‌های کاربر در طی پروژه لازم است که نرم‌افزار issue management وجود داشته باشد تا ثبت و دنبال کردن این موارد با سهولت انجام گیرد. همچنین گزارش باگ‌ها و خطاها می‌تواند توسط سیستم‌های help desk امروزی به خوبی انجام گیرد. همچنین نیاز به سیستم continuous integration است تا تغییرات به محیط production رفته و اعمال گردد.

در مجموع می‌توان گفت با پیشرفت حوزه مهندسی نرم‌افزار این موارد امروزه چالش جدی محسوب نمی‌شوند و راهکارهای رایگان و متن‌باز برایشان وجود دارد.

متدولوژی FDD

متدولوژی FDD به دلیل full lifecycle در معیار practicality و practicability دارای نقایصی است. چون تیم توسعه نمی‌داند از چه practice هایی بایستی استفاده کند و دچار ضرر خواهد شد. اما از طرفی دیگر در هنگام توسعه

محصول با شکستن بسیار خوب مسئله به حوزه‌های مختلف باعث می‌شود تیم در توسعه محصول کار بسیار راحتی داشته باشد.

ناگفته نماند که شکسته شدن تیم به زیرتیم‌های مختلف دارای یک نکته منفی است. دلیل آن است که هنگام توسعه محصول در جاهایی که لازم است یک ویژگی تغییر کند، نیاز است تا صاحب کلاس فراخوانده شود و موضوع به او ارجاع داده شود تا پیاده سازی صورت گیرد. این کار در پروژه‌های بزرگ شاید دست و پاگیر شود و سرعت توسعه را کاهش دهد.

متدولوژی DSDM

این متدولوژی از فرایندهای ساده و بدون پیچیدگی استفاده می‌کند. همچنین پیچیدگی فرایندها در آن کم می‌باشد. وجود فعالیت‌های دقیق در مدیریت پروژه باعث می‌گردد ریسک کار پایین آمده و انجام فعالیت‌ها با دقت انجام گیرد. از منظری دیگر نیز انجام برنامه ریزی دقیق در مقاطع مختلف باعث بهبود **practicality** و **practicability** می‌گردد.

مقایسه دو متدولوژی DAD و XP

هر دو متدولوژی در این معیار نمره بالایی می‌گیرند. اما متدولوژی XP برای انجام **practice** های خود در برخی مواقع نیاز به نرم‌افزارهای مخصوص دارد. این کار امروزه چالش اصلی محسوب نمی‌گردد و با به وجود آمدن دواپس می‌توان گفت بسیاری از این کارها اتوماتیک شده و از طرفی باعث سهولت کار می‌گردد. پس نمی‌توان از مزایای این موارد چشم‌پوشی کرد و به همین دلیل می‌توان گفت متدولوژی XP به دلیل این **practice** های با ارزش سهولت بیشتری وارد کرده و از DAD پیشی خواهد گرفت.

مقایسه دو متدولوژی DAD و FDD

متدولوژی FDD در توسعه کد کار را راحت می‌کند. اما نکته منفی ای دارد که ذکر شد. نکته منفی آن است که برای هر تغییر نیاز به فراخوانی صاحب آن کلاس است. این کار باعث می‌شود که نیاز به هماهنگی بسیار زیاد شود. به همین دلیل می‌توان گفت به دلیل سراسر بودن کار توسعه و مدیریت تیم در DAD این متدولوژی نسبت به FDD دارای برتری است.

مقایسه دو متدولوژی DAD و DSDM

متدولوژی DSDM بر محصولات، نقشها و practice ها به خوبی دقت کرده و مانور داده است. به همین کار تیم توسعه را در هنگام ساخت محصول بسیار راحت می کند. در نتیجه این متدولوژی از DAD در این معیار جلوتر است و به عنوان انتخاب اول است.

معیار نهم) قابلیت مدیریت پیچیدگی

متدولوژی DAD

مدیریت پیچیدگی کارها و واحدهای کاری در این متدولوژی از چندین طریق انجام می‌شود. اول آنکه نقش‌های گوناگون برای افراد تیم تعریف می‌شود. یک نفر مسئولیت رهبری تیم را برعهده می‌گیرد. نفری دیگر معماری را در دست گرفته و آن را راهبری می‌کند. شخصی دیگر صاحب محصول شده و پیگیری امور و کیفیت را برعهده می‌گیرد. تمامی این موارد نشانه‌هایی از مدیریت پیچیدگی در این متدولوژی می‌باشد. از طرفی تعریف فازهای مختلف هر کدام با هدفی تعیین شده و وجود واحدهای ریز کاری نیز باعث شده تا پیچیدگی واحدهای کاری تا حد خوبی از بین برود.

نکته آخر ذیل این مبحث آن است که در هر فاز از این متدولوژی **practice** هایی تعیین شده تا طبق آنها هم کیفیت محصول افزایش یابد و هم اینکه پیچیدگی‌های موجود کمر تیم را خم نکند.

متدولوژی XP

متدولوژی XP پیچیدگی واحدهای کاری را به وسیله تقسیم کارها به فازها و فعالیت‌های درونشان تا حد خوبی کاهش داده است. همچنین تولید معماری در ابتدای پروژه بر کاهش پیچیدگی پیاده سازی تاثیر مثبتی خواهد داشت.

این متدولوژی به وسیله **practice** های اعمال شده میتواند کیفیت کد تولید شده را بسیار ارتقا دهد. در نتیجه پیچیدگی کد در آینده و با پیشرفت پروژه تا حد امکان زیاد نخواهد شد. یک تکنیک بسیار اصلی به نام ریفتورینگ باعث سادگی کد می‌شود.

متدولوژی FDD

متدولوژی FDD ابتدا سیستم را به ناحیه های مجزا می‌شکند. سپس هر ناحیه از تعدادی فعالیت تشکیل می‌گردد که واحد هر فعالیت نیز ویژگی است. این سلسله مراتب به وجود آمده تاثیر شگرفی بر مدیریت پیچیدگی خواهد داشت. به همین دلیل می‌توان گفت که متدولوژی FDD در این معیار نمره بالایی خواهد گرفت. همچنین وجود فعالیت‌ها به صورت **iterative** باعث می‌گردد که محصول طراحی شده و پیاده سازی گردد. که در مجموع سرعت نهایی را بالا خواهد برد.

متدولوژی DSDM

وجود نقش های مختلف و جداسازی وظایف از هم باعث می شود که پیچیدگی به خوبی مدیریت گردد. همچنین فازبندی مراحل و تولید محصولات گوناگون باعث می گردد که پیچیدگی کاهش یابد. وجود برنامه ریزی نیز این پیچیدگی را کاهش می دهد.

مقایسه دو متدولوژی DAD و XP

هر دو متدولوژی به صورت خوبی بر کاهش پیچیدگی سیستم مانور داده اند. اما اگر بخواهیم یکی را انتخاب کنیم به دلیل وجود practice هایی مانند ریفتورینگ در XP که باعث سهولت بسیار زیاد و سادگی می گردد، می توان گفت XP از DAD پیشتر است.

اما متدولوژی DAD نیز با استفاده از نقش های مختلف در تولید سیستم توانسته است بسیار به کاهش پیچیدگی کارها و فعالیت های افراد توسعه دهنده کمک نماید.

مقایسه دو متدولوژی DAD و FDD

ایده ای که متدولوژی FDD در پیاده سازی ویژگی ها استفاده کرده است باعث می شود که پیچیدگی تا حد بسیار زیادی کاهش یابد. وجود تیم های مختلف با مسئولانی که وظایف را تقسیم می کنند، پیچیدگی را بسیار کاهش می دهد. متدولوژی DAD نیز واحدهای کاری مختلف و نقش بندی های مختلف را برای مدیریت پیچیدگی قرار داده است اما من حیث المجموع متدولوژی FDD بسیار در این معیار برتر است.

مقایسه دو متدولوژی DAD و DSDM

وجود practice های غنی تر در DSDM و توجه مناسب به مباحث کسب و کاری باعث می گردد که DSDM دارای برتری نسبت به DAD گردد.

معیار دهم) قابلیت گسترش / مقیاس پذیر / قابلیت شخصی سازی / انعطاف پذیری

متدولوژی DAD

قابلیت گسترش و شخصی سازی در این متدولوژی دیده شده است. این متدولوژی در انتهای هر iteration و هر فاز جلسه مرور و ریویو قرار داده تا فرایندها و فعالیتها دچار بازبینی گردند. این کار امکان تغییر و به روزرسانی متدولوژی را به افراد تیم اعطا می کند.

قابلیت مقیاس پذیری نیز در این متدولوژی به خوبی دیده شده است. تیم های استفاده کننده از این متدولوژی می توانند بین ۲ تا ۴۰ نفر باشند. اگر زیر ۱۵ نفر باشند، سایز تیم کوچک در نظر گرفته شده و اگر بالای آن باشد، سایز متوسط لقب می گیرد. در صورتی که سایز تیم متوسط باشد، زیر تیم هایی تشکیل می شود که خود دارای نقش های اصلی و ثانویه است. در نتیجه می توان سیستم را به زیرسیستم هایی ساخت که هر زیر تیم مسئول توسعه یک یا چند زیرسیستم خواهد بود. پس سامانه های با مقیاس بزرگ را به وسیله این متدولوژی می توان توسعه داد.

متدولوژی XP

متدولوژی XP به صورت صریح درباره مکانیزم های گسترش، شخصی سازی و انعطاف پذیری مطلبی را بیان نکرده است. اما در طی تولید سیستم، هم در ابتدای شروع iteration و هم در انتهای آن جلسات برنامه ریزی و ریویو قرار داده شده است تا بتوان فرایندها را تنظیم کرده و بر آن توافق نمود. این قابلیت تاحدی به وجود انعطاف پذیری و شخصی سازی کمک می کند.

قابلیت مقیاس پذیری نیز در این متدولوژی بیان نشده است. اما می توان گفت به دلیل استفاده نکردن از مدل سازی و علی الخصوص عدم استفاده از فرمالیزم قابلیت پشتیبانی از پروژه های با سطح بحرانیت بالا و پروژه های مقیاس بزرگ وجود نخواهد داشت.

متدولوژی FDD

متدولوژی FDD به صورت کلی بیان نموده است که چگونه سیستم به ناحیه ها شکسته شده و هر ناحیه چگونه پیاده سازی گردد. در نتیجه می توان این متدولوژی را به صورت یک هسته کلی در نظر گرفت که دست تیم توسعه را در نحوه چگونگی کار و فعالیت های جانبی باز گذاشته است. همچنین به همین دلیل مقیاس پذیری در این متدولوژی بسیار بالاست. چون تیم های مختلف می توانند مسئولیت ناحیه ها را برعهده گرفته و هر کدام یک زیرسیستم را پیاده سازی نماید. در

کل بنابر دلایل بالا می‌توان گفت در قابلیت های گسترش‌پذیری، مقیاس‌پذیری، شخصی‌سازی و انعطاف‌پذیری این متدولوژی نمره بالایی کسب می‌کند.

متدولوژی DSDM

این متدولوژی به صورت فیکس و کامل تمامی مراحل و گام‌ها را ارائه داده است. همچنین در متن خود متدولوژی نقاط شخصی‌سازی و انعطاف را ذکر نکرده است. به همین دلیل نمره پایینی در این معیار کسب می‌کند.

مقایسه دو متدولوژی DAD و XP

متدولوژی DAD به دلیل ذکر صریح موارد قابل شخصی‌سازی در متن متدولوژی در این معیار بالاتر و بهتر است. همچنین عدم model-phobic بودن این متدولوژی نیز به مقیاس‌پذیر بودن آن بسیار کمک می‌کند که متاسفانه متدولوژی XP از این قابلیت بی‌بهره است.

متدولوژی DAD همچنین قابلیت استفاده از اعضای بیشتر در پیشبرد پروژه را فراهم می‌آورد. این متدولوژی می‌تواند تا تیم‌های ۴۰ نفره را پشتیبانی کرده و از آنها در متن پروژه استفاده نماید. در نتیجه قابلیت مقیاس‌پذیری نیز در این متدولوژی بهتر از XP می‌باشد.

مقایسه دو متدولوژی DAD و FDD

متدولوژی FDD در بسیاری موارد دست تیم توسعه را برای شخصی‌سازی و انتخاب نحوه انجام فرایند باز گذاشته است. همچنین قابلیت شکستن مسئله کلی به ناحیه‌های مختلف باعث می‌شود تا مقیاس‌پذیری در سطح خوبی برقرار گردد. متدولوژی DAD نیز توانایی مدیریت تیم‌های تا ۴۰ نفر را داراست و مقیاس‌پذیری مناسبی دارد اما در مجموع FDD در این معیار برتر است.

مقایسه دو متدولوژی DAD و DSDM

متدولوژی DAD دست تیم توسعه را بازتر گذاشته و نقاط شخصی‌سازی را بهتر بیان نموده پس از DSDM در این معیار بهتر است.

معیار یازدهم) حوزه کاربرد (Application scope)

متدولوژی DAD

حوزه کاربرد این متدولوژی سیستم‌های تجاری با سایزهای کوچک، متوسط و بزرگ می‌باشد. در صورتی که بخواهیم پروژه‌های متوسط و بزرگ را با این متدولوژی انجام دهیم، تیم‌های با سایز متوسط لازم است. سیستم‌های اطلاعاتی و تجاری امروزی به خوبی توسط این متدولوژی پوشش داده خواهند شد.

اما نکته منفی درباره این متدولوژی عدم قابلیت پشتیبانی از سیستم‌های با درجه بحرانی بالا است. دلیل نیز آن است که فرمالیزم در این متدولوژی منظور نشده است. تلاش‌های متعددی در متدولوژی‌های چابک گوناگون صورت گرفته تا فرمالیزم به آنها اضافه شود اما متأسفانه بایستی گفت که همگی به در بسته خورده و نتوانسته این گره کور را باز کند. پس متدولوژی DAD نیز از این قاعده مستثنی نبوده و حرفی برای گفتن در باب فرمالیزم نخواهد داشت.

متدولوژی XP

متدولوژی XP برای استفاده در سیستم‌های تجاری که نیازی به فرمالیزم ندارند بسیار مطلوب است. این متدولوژی برای تیم‌های با تعداد افراد کم قابل استفاده است و با صرف هزینه کم می‌توان با سرعت بالا به محصول مورد استفاده رسید. سیستم‌های اطلاعاتی به صورت سیستم‌های عام منظوره توانایی انجام با این متدولوژی را داشته و می‌توان از آن بهره برد. پس این متدولوژی در حوزه کاربرد نیز طیف وسیعی از پروژه‌های موجود را پوشش خواهد داد.

متدولوژی FDD

از FDD می‌توان در پروژه‌های کوچک و بزرگ امروزی استفاده نمود. این متدولوژی فرمالیزم ندارد و به همین دچار محدودیت‌هایی می‌گردد. اما در سیستم‌های تجاری به خوبی قابلیت استفاده را دارد. مدلسازی نیز در این متدولوژی تا حدی انجام می‌شود و قابلیت مقیاس به پروژه‌های بزرگ تر را خواهد داشت.

متدولوژی DSDM

این متدولوژی نیز حوزه کاربرد بسیار مشابه با دیگر متدولوژی‌های چابک دارد و در سیستم‌های اطلاعاتی قابلیت استفاده دارد. همچنین نبود فرمالیزم باعث عدم استفاده در پروژه‌های با بحرانی بالا می‌گردد.

مقایسه دو متدولوژی DAD و XP

هر دو متدولوژی فاقد فرمالیزم هستند. به همین دلیل در سیستم های بحرانی قابلیت استفاده را نخواهند داشت. اما در سیستم های تجاری و اطلاعاتی امروزه که کاربردهای وسیعی دارند، بسیار مناسب برای استفاده می باشند. متدولوژی DAD توانایی پشتیبانی از تیم های با تعداد نفرات بالاتر را داراست. از طرفی دیگر XP به دلیل کیفیت بالای کد تولید شده به نسبت DAD دارای مزیت هایی می باشد. در کل نمی توان یک متدولوژی را بر دیگری برتری داده و انتخاب نمود.

مقایسه دو متدولوژی DAD و FDD

هر دو متدولوژی به دلیل ساپورت نکردن فرمالیزم دچار ضعف شده اند. اما در سیستم های تجاری هر دو به خوبی قابلیت استفاده را دارند. اما چون FDD یک متدولوژی full lifecycle نیست و از آن سو DAD این خاصیت را دارد، پس حوزه کار وسیع تری را نیز خواهد داشت. پس DAD در حوزه کاربرد وسیع تر و انتخاب بهتری است.

مقایسه دو متدولوژی DAD و DSDM

هر دو متدولوژی در یک سطح هستند و تمایزی در پروژه های قابل انجام توسط هر کدام نیست.

معیار اول زبان مدل‌سازی) پشتیبانی از مدل‌سازی شی گرای سازگار، دقیق و بدون ابهام

متدولوژی DAD

این متدولوژی در مورد مدل‌سازی صحبت زیادی نگفته است. متدولوژی‌های چابک اکثراً فاقد مدل‌سازی سنگین و حزئی می‌باشند. این متدولوژی اما **practice** هایی برای مدل‌سازی در خود نهاده است. از جمله این موارد می‌توان به موردهای زیر اشاره نمود:

- **Model Storming**: تولید مدل‌های گوناگون برای اعتبارسنجی و پیدا کردن راهکار
- **Architecture spike**: معماری اولیه در جهت پیدا کردن معماری برتر و اصلی
- **Automated metrics**: معیارهای لازم برای ارزیابی مصنوعات و فرایندها
- **Continuous documentation**: مستندسازی پیوسته از سیستم تولید شده
- **Look-ahead modeling**: مدل‌سازی پیشاپیش برای موارد آتی

مدلسازی‌های موجود بیشتر در سطوح بالا یعنی سطح کسب و کار است و همچنین می‌توان گفت نبود فرمالیزم باعث می‌شود که حوزه استفاده از متدولوژی بسیار کوچک گردد.

متدولوژی XP

متدولوژی XP یک متدولوژی **model-phobic** است که نسبت به مدل‌سازی بسیار مقاوم است و از آن پرهیز می‌کند. در نتیجه از وجود مدل‌های ساختاری، رفتاری و عملکردی بی‌بهره می‌باشد. تنها مدل‌سازی آن ثبت داستان‌های کاربر است که نیازمندی‌ها را استخراج می‌کند. همچنین از فرمالیزم نیز پشتیبانی نمی‌کند. در کل می‌توان گفت که در این معیار اصلاً نمره خوبی کسب نمی‌کند و در نتیجه در پروژه‌های پیچیده قابلیت استفاده را نخواهد داشت.

متدولوژی FDD

متدولوژی FDD به صورت کاملاً شی‌گرا می‌باشد. وجود مفهوم ویژگی و قرار دادن تمام مفاهیم حول آن باعث می‌شود ناسازگاری وجود نداشته باشد. این مدل‌سازی به صورت دقیق و بدون ابهام توسط **object model** انجام می‌گیرد. هر **object model** شامل ناحیه‌های مختلف است که به صورت پیوسته **refine** می‌شود.

مدلسازی‌های انجام شده شامل دیدهای ساختاری (توسط object model)، عملکردی (ویژگیها) و رفتاری (نمودار توالی) می‌باشد. این مدل‌ها هم شامل فضای مسئله و هم فضای راه‌حل می‌باشد. مدلسازی کسب و کار نیز به کمک خبره‌های حوزه مسئله انجام می‌شوند.

این مدلسازی‌ها در سطوح مختلف انجام می‌شود. طبق ادبیات این متدولوژی ناحیه، فعالیت و ویژگی مدل می‌شوند که به نوعی مدلسازی سیستم، زیرسیستم، میان شی و درون شی است. پس به صورت کلی می‌توان گفت مدلسازی بسیار غنی و کامل توسط این متدولوژی انجام می‌شود.

تنها نقطه ضعف آن است که مدلسازی فرمال در آن وجود ندارد.

متدولوژی DSDM

محصولات این متدولوژی عبارتند از: تحلیل امکان سنجی، کد توسعه یافته، برنامه ریزی تایم باکس، مرور پروژه، تحلیل سودمندی، مورد تجاری، معماری فنی و برنامه مدیریتی. همانگونه که بیان شد، در سطوح مختلف کسب و کار و پیاده‌سازی این متدولوژی دارای محصولات و مدلسازی است و در این معیار بسیار قوی تر نسبت به دیگر متدولوژی‌ها است. اما این متدولوژی دارای فرمالیزم نمی‌باشد.

مقایسه دو متدولوژی DAD و XP

متدولوژی DAD در این معیار برتر است. دلیل آن است که به دلیل وجود مدلسازی‌های بیشتر و غنی تر امکان استفاده از متدولوژی را در پروژه‌های بزرگتر مهیا خواهد کرد. متدولوژی XP به دلیل مدل فوییک بودن این امکان را از تیم توسعه خواهد گرفت. تنها نکته مثبت در این متدولوژی یعنی XP آن است که معماری اصلی سیستم را در ابتدای پروژه یعنی فاز اول فراهم می‌نماید. در کل می‌توان گفت متدولوژی DAD در این معیار برتری فاحشی دارد.

مقایسه دو متدولوژی DAD و FDD

همانگونه که در بخش FDD بیان شد، این متدولوژی تمام سطوح و جزئیات را در نظر گرفته و مدل می‌کند. اما DAD کمی از مدلسازی فاصله گرفته و مدلسازی به اندازه FDD غنی نیست. در نتیجه در این معیار FDD بسیار برتر است. متدولوژی FDD انواع دیدهای مختلف، مدلسازی منطقی و فیزیکی و درشت‌دانگی‌های مختلف انجام می‌شود.

مقایسه دو متدولوژی DAD و DSDM

همانگونه که بیان شد متدولوژی DSDM دارای مدلسازی های متنوع در انواع سطوح مختلف است. به همین دلیل مدلسازی شی گرای آن بسیار کامل می باشد. هر دو فاقد فرمالیزم می باشند. در مجموع می توان گفت DSDM در این معیار برتر است.

معيار دوم زبان مدلسازی) تهیه استراتژی ها و تکنیک هایی برای مقابله با ناسازگاری مدل و مدیریت پیچیدگی مدل

متدولوژی DAD

به دلیل وجود مدل های کم در این متدولوژی می توان گفت که ناسازگاری و پیچیدگی تا حد امکان در آن کاهش خواهد یافت. بدین صورت تکنیک و روش برای مدیریت پیچیدگی در آن دیده نشده است.

اما تیم استفاده کننده می تواند به وسیله روش های موجود مثلا در UML در جهت مقابله با این مورد گام بردارد.

متدولوژی XP

متدولوژی XP از مدلسازی تحلیل و طراحی سیستم بسیار پرهیز می کند. اما همانگونه که می دانیم کد تولید شده برنامه نیز جزو مصنوعات و مدل ها محسوب می گردد. متدولوژی XP به دلیل وجود practice های غنی در دل خود توانایی مدیریت کدهای پیچیده را در یک محیط collective code ownership فراهم می نماید. پس در این معیار با استناد به دلایل مطرح شده بسیار قابل قبول خواهد بود.

متدولوژی FDD

در FDD به صورت دقیق مدل ها بازبینی شده و مدام مرور می شود. هر ناحیه به تیمی سپرده شده و آن تیم مسئولیت تمام جزئیات را برعهده خواهد گرفت. زبان مدلسازی آن به صورت دقیق بر مدل ها و نمودارها دقت کرده و جلوی ناسازگاری را می گیرد. در صورت نیاز به تغییر در بخشی، صاحب کلاس فراخوانده شده و از تغییرات مطلع گشته و خود آن را اعمال می کند. در نتیجه در این معیار نیز متدولوژی FDD بسیار نمره بالایی دریافت می کند.

متدولوژی DSDM

در این متدولوژی تعداد محصولات و نمودارها زیاد است. همچنین گستردگی و تنوع در میانشان زیاد می باشد. اما مکانیزم دقیق و متقنی برای رفع ناسازگاری ها ذکر نگردیده است. به همین دلیل در این معیار دچار ضعف است.

مقایسه دو متدولوژی DAD و XP

همانگونه که مطرح شد DAD فاقد مکانیزم صریح برای جلوگیری از ناسازگاری میان مدل‌هاست. از آن سو XP بسیار از تولید مدل جلوگیری می‌کند. اما به وسیله practice های موجود در خود می‌تواند از ناسازگاری موجود در کد و سیستم تولید شده جلوگیری نماید. پس می‌توان در این معیار متدولوژی XP را برتر از DAD دانست و نمره بالاتری به آن اعطا کرد.

مقایسه دو متدولوژی DAD و FDD

همانگونه که بیان شد، متدولوژی FDD به صورت کامل و دقیق بر سازگاری مدل‌ها نظارت می‌کند. تمام مفاهیم یعنی ناحیه، فعالیت و ویژگی با هم در ارتباط بوده و به صورت مدام این جزئیات بررسی می‌گردد. در DAD نیز مدلسازی‌ها به صورت کمینه است و مدل‌ها با هم در ارتباط هستند. اما چون FDD همه را حول مفهوم ویژگی می‌چیند، در نتیجه بسیار در این معیار برتر از DAD خواهد بود.

مقایسه دو متدولوژی DAD و DSDM

متدولوژی DSDM دارای مدل‌ها و محصولات متنوع تری است اما فاقد مکانیزم دقیق برای پایش و کنترل سازگاری میان آنهاست. به همین دلیل در این معیار متدولوژی DAD برتر خواهد بود.

سوال دوم) مقایسه متدولوژی Scrum با Crystal Clear

معیار اول) تعریف شفاف و دقیق متدولوژی

متدولوژی اسکرام

اسکرام در ابتدا توسط شرکت های ژاپنی برای استفاده در کارخانجات معروف در فرایندهای تولید محصولات ابداع و استفاده گردید. سپس در سال ۱۹۹۵ توسط آقایان سادرلند و شوابر برای تولید نرم افزار و محصولات کامپیوتری بیان شد. این متدولوژی بیشتر تمرکز خود را بر روابط میان افراد و نحوه همکاری در جهت پیشینه کردن خروجی گذاشته است.

متدولوژی اسکرام بر پایه ارزش ها، اصول و روش ها بیان شده است. این موارد نحوه ارتباط میان افراد برای پیشینه نمودن کارکرد هر شخص بیان شده اند. اسکرام سعی می نماید تا صداقت، شفافیت، شجاعت، احترام، اعتماد و همکاری میان افراد تیم مطلوب ترین و بالاترین سطح خود را داشته باشد. در نتیجه با فراهم سازی این موارد برای افراد تیم، بهترین عملکرد را از ایشان شاهد خواهیم بود که در نهایت منجر به تولید محصول با کیفیت در سریع ترین زمان ممکن خواهد شد. همچنین تاکید زیاد اسکرام بر نحوه انجام فعالیت ها و توسعه نرم افزار باعث می گردد که تیم نقشه راه مناسبی در زمان تولید نرم افزار انتخاب نماید.

هر تیم اسکرام شامل افراد مختلف با نقش های گوناگون می باشد. این نقش ها شامل صاحب محصول، اسکرام مستر و تیم توسعه می باشد. صاحب محصول شخصی است که موارد و نیازمندی های سیستم را برای پیاده سازی تعیین می کند و بر کیفیت خروجی تیم نظارت می کند. اسکرام مستر شخصی است که بر نحوه اجرای اسکرام نظارت کرده و با بهبود مستمر فعالیت ها باعث می شود تیم از مسیر خود خارج نشود. تیم توسعه نیز وظیفه تولید سیستم نهایی را بر عهده دارد.

نیازمندی ها و قابلیت های محصول در حال توسعه در قالب backlog نگه داری می شود. Backlog محصول شامل ویژگی هایی است که در چشم انداز نهایی پروژه می خواهیم در محصول موجود باشد. سپس در تعدادی اسپرینت به ترتیب اولویت و با تصمیم گیری تیم، کارها از backlog محصول برداشته شده و در backlog اسپرینت قرار داده می شود برای انجام و اعتبارسنجی. هر اسپرینت دارای زمان شروع و پایان می باشد و در این مدت سعی می شود تمام کارهای درون backlog اسپرینت انجام گردد. در انتهای هر اسپرینت یک جلسه مرور اسپرینت برگزار می شود که در آن تمام ذی نفعان، مدیر محصول، اعضای تیم اسکرام، مشتری و افراد علاقه مند دیگر شرکت می کنند تا وضعیت محصول و میزان کارکرد آن را بسنجند. همچنین بعد از جلسه مرور اسپرینت، جلسه ای با نام sprint retrospective برگزار می گردد تا تیم اسکرام در کنار هم جمع شده و با سنجش فرایندها و نحوه انجام کار، موارد نقص را برطرف کنند.

امروزه متدولوژی اسکرام به نوعی معروف‌ترین و پر استفاده‌ترین متدولوژی چابک مورد استفاده بین تیم‌های توسعه نرم افزار می‌باشد. به همین دلیل کتاب‌ها و مقالات گوناگون حول آن نوشته می‌شود. تیم‌های مختلف پس از استفاده از آن در فرایند توسعه نرم‌افزار، تمامی تجربیات و نکات یاد گرفته شده را در کتاب‌ها و صفحات اینترنتی مکتوب می‌کنند. کتاب روبین که یکی از معروف‌ترین کتاب‌ها برای تدریس اسکرام می‌باشد نیز بدین گونه شکل گرفته است. متن اصلی اسکرام توسط آقایان سادرلند و شوابر تحت نام **Scrum Guide** تالیف شده است که یک چارچوب کلی و سطح بالا از اسکرام ارائه داده اند. اما تیم‌ها و افراد مختلف تجربیات و آموخته‌های خود را به صورت پیوسته در اختیار دیگران قرار می‌دهند.

در نتیجه به صورت کلی می‌توان گفت متدولوژی اسکرام از نظر معیار تعریف کامل و شفاف یکی از بهترین متدولوژی‌های موجود می‌باشد. وجود مستندات و کتاب‌های زیاد و جامع، تعریف شفاف و دقیق متدولوژی و توضیحات دقیق از تمامی فعالیت‌های آن موجود می‌باشد. چرخه حیات تولید نرم‌افزار در اسکرام با نام اسپرینت شناخته می‌شود. سپس ذیل هر اسپرینت واحدهای کاری برنامه ریزی اسپرینت، استندآپ روزانه، جلسه مرور اسپرینت و جلسه **retrospective** انجام می‌گیرند. محصولات این متدولوژی عبارت‌اند از بک‌لاگ محصول، بک‌لاگ اسپرینت و محصول قابل فروش و عرضه. همچنین قواعد و تکنیک‌های مورد استفاده توسط تیم در گام‌های مختلف تولید محصول به دقت بیان شده است. فعالیت‌های متناسب با فعالیت‌های چتری که شامل مدیریت ریسک، مدیریت کیفیت و مدیریت پروژه می‌باشد نیز به دقت در این متدولوژی تشریح شده است. اما به دلیل اینکه این متدولوژی به صورت **model-phobic** می‌باشد، توصیف خاصی درباره زبان مدلسازی مورد استفاده نمی‌کند. اما جلوی برنامه نویس را برای مدلسازی نمی‌گیرد.

در کل این متدولوژی درباره تمام جزئیات تولید سیستم نرم‌افزاری فعالیت‌ها و واحدهای کاری را به دقت بیان نموده است و در این معیار نمره بسیار بالایی می‌گیرد.

متدولوژی کریستال کلیر

خانواده متدولوژی کریستال توسط آقای کوبرن در سال ۱۹۹۸ تبیین شد. این دسته از متدولوژی‌ها توانایی پوشش پروژه های با سطوح بحرانی متفاوت به جز سطح L را دارند (یعنی سطوح C, D, E را پوشش می‌دهند). همچنین با اضافه کردن مدلسازی به آن می‌توان تیم‌های با سایزهای مختلف را پشتیبانی نمود. سه متدولوژی مورد استفاده در صنعت از خانواده کریستال عبارتند از کریستال اورنج، کریستال اورنج وب و کریستال کلیر. متدولوژی‌های خانواده کریستال می‌توانند تیم‌های با سایز ۶، ۲۰، ۴۰ و ۸۰ را پشتیبانی نمایند.

این متدولوژی‌ها جزو دسته متدولوژی‌های چابک قرار می‌گیرند و به اصول آن پایبند هستند. اعضای تیم بایستی در کنار یکدیگر و در یک ساختمان قرار بگیرند و به صورت توزیع شده نمی‌توانند در پروژه شرکت داشته باشند. این خانواده اهمیت زیادی بر روابط میان افراد تیم و نحوه انجام کار گروهی می‌دهند. این متدولوژی یک چهارچوب اصلی است که

توسط تیم توانایی شخصی سازی و تغییر را دارند. در جلسه reflection افراد می توانند فرایندها را تغییر داده و متناسب با شرایط تنظیم نمایند.

متدولوژی کریستال کلیر تمرکز اصلی اش را بر پروژه‌های با سطح بحرانیت C و D قرار داده که در تیم های شش نفره ساخته می شوند. این متدولوژی از سه فاز اصلی chartering، تحویل دوره‌ای^۱ و اتمام تشکیل شده است. در فاز اول اعضای تیم انتخاب شده و نقش های مدیر پروژه، متخصص فنی، متخصص آموزش، کاربر سفیر و تست کننده سیستم به اعضا محول می شود. سپس عملیات Exploratory 360 degrees انجام میگیرد که به نوعی تحلیل امکان سنجی سیستم است و نیازمندی های سطح بالای سیستم مشخص می گردد. همچنین مدل های دامنه مسئله استخراج می شود و تکنولوژی های مورد استفاده مشخص می گردد. در انتهای این بخش متدولوژی توسعه نیز مشخص می گردد. در ادامه این فاز قوانین و فرایندهای متدولوژی مشخص می گردد و در آخر برنامه ریزی برای فازهای آتی پروژه صورت می گیرد. در فاز دوم که تحویل دوره ای سیستم است، ابتدا برنامه انتشار سیستم تبیین می گردد. سپس در طی چندین تکرار سیستم توسعه می یابد که در انتهای هر تکرار سیستم به کاربر نهایی عرضه می شود و اعتبارسنجی سیستم صورت می گیرد. در انتهای هر تکرار یک ورکشاپ برگزار می شود که کیفیت محصول ارزیابی شده و متدولوژی بازبینی می گردد. در فاز نهایی نیز تست سیستم صورت می گیرد و سیستم به محیط عملیاتی برده شده و راه اندازی می گردد.

همانگونه که ذکر شد، این متدولوژی به صورت کامل و جامع بیان شده و سه نسخه اورنج، اورنج وب و کلیر به صورت وسیعی مورد استفاده قرار گرفته اند. نقش ها در این متدولوژی مشخص شده و به افراد محول می گردد. قواعد و تکنیک ها تا حد خوبی در این متدولوژی بیان شده و اعضای تیم می توانند در طی جلسات ریویو به شخصی سازی و بهبود آنها بپردازند. همچنین فعالیت های چتری در این متدولوژی به خوبی پوشش داده شده اند. در نتیجه می توان گفت که از نظر تعریف شفاف و دقیق این متدولوژی بسیار قوی می باشد.

مقایسه متدولوژی اسکرام با کریستال کلیر

هر دو متدولوژی بسیار پر استفاده می باشند و این باعث شده است که از نظر تعریف دقیق و شفاف اجزای یک متدولوژی نمره بالایی دریافت کنند. چرخه تولید نرم افزار در هر دو بیان شده و به صورت تکراری فرایند توسعه سیستم صورت می گیرد. نقش های اعضای تیم در هر دو بیان شده است. فعالیت های چتری مهندسی نرم افزار اعم از مدیریت پروژه، برنامه ریزی و کنترل کیفیت به صورت دقیقی در هر دو اجرا می گردد.

در انتها اگر بخواهیم یکی از این دو متدولوژی را به عنوان انتخاب اول برگزینیم، می توان گفت اسکرام به دلیل استفاده بیشتر توسط تیم های توسعه نرم افزار و وجود منابع بیشتر برای تشریح این متدولوژی، از متدولوژی کریستال کلیر برتر

^۱Cyclic delivery
^۲Wrap-up

است. خانواده متدولوژی کریستال دارای اعضای است که به صورت روی کاغذ و اسمی وجود دارند اما در عمل از آنها استفاده نمی شود. برای مثال کریستال قرمز، آبی و بنفش از جمله این متدولوژی ها هستند که دارای تعریف دقیق و شفاف نمی باشند.

معیار دوم) پوشش فعالیت‌های عمومی چرخه توسعه نرم‌افزار

متدولوژی اسکرام

متدولوژی اسکرام بیشتر تاکید خود را بر نحوه‌ی انجام کار و چگونگی ارتباط افراد برای بیشینه کردن بهره‌وری گذاشته است. بیشتر از آنکه به جزئیات فعالیت‌های عمومی تولید نرم‌افزار پرداخته باشد، بر نحوه و چگونگی انجام کار بحث و بررسی قرار داده است. اما می‌توان در قالب کلی‌ای که این متدولوژی ارائه داده است، فعالیت‌های گوناگون تولید نرم‌افزار را گنجانده.

مهم‌ترین مفهوم و فرایند در متدولوژی اسکرام، اسپرینت می‌باشد. برای ساخت هر محصولی با استفاده از اسکرام، در طی تعدادی اسپرینت محصول به صورت پیوسته ساخته می‌شود. هر اسپرینت از تعدادی گام تشکیل می‌شود که عبارتند از: **backlog grooming**، برنامه ریزی اسپرینت، انجام اسپرینت، استندآپ روزانه، تولید محصول قابل استفاده، مرور اسپرینت و جلسه **retrospective** برای اسپرینت.

در متدولوژی اسکرام برخی از فعالیت‌های عمومی تولید نرم‌افزار به صورت مستقیم پوشش داده شده است و برخی نیز به صورت غیرمستقیم انجام می‌گیرند. در ادامه فعالیت‌های عمومی تولید نرم‌افزار بیان شده و متناظر آن در متدولوژی اسکرام بیان می‌شود. تحلیل امکان‌سنجی با روش **fail fast** انجام می‌شود. بدین صورت که برای امکان‌سنجی یک راه‌حل، ابتدا با نمونه‌سازی سنجیده می‌شود که آن کار قابل انجام هست یا خیر. بدین صورت گام و فعالیت جداگانه در متدولوژی اسکرام قرار داده نشده است اما به وسیله نمونه‌سازی این کار در اسپرینت‌های مختلف انجام می‌شود. استخراج نیازمندی‌ها در این متدولوژی توسط صاحب محصول انجام می‌شود. نیازمندی‌ها، قابلیت‌ها، باگ فیکس‌ها و ... به بک‌لاگ محصول اضافه می‌گردد. همچنین در ابتدای هر اسپرینت، مواردی از بک‌لاگ برداشته شده و شکسته می‌شود تا قابلیت انجام در یک اسپرینت را داشته باشد. این تسک‌ها به همراه مورد بک‌لاگ متناظرشان در بک‌لاگ اسپرینت قرار داده می‌شود. اما فعالیت مدل‌سازی دامنه مسئله که از فعالیت‌های ذیل تعریف پروژه می‌باشد، در این متدولوژی مورد بحث قرار نگرفته است. به صورت کلی این متدولوژی یک متدولوژی **model-phobic** است که تا جای ممکن از مدل‌سازی‌های با ضرورت کم پرهیز می‌نماید. به همین دلیل تنها در جاهایی که مدل‌سازی برای یک کار پیچیده (مثل الگوریتمی دشوار) لازم باشد، از انجام آن پرهیز می‌گردد.

متدولوژی اسکرام فعالیت‌های مرتبط با طراحی معماری و طراحی جزئیات سامانه را طی گام جداگانه‌ای در نظر نمی‌گیرد. این متدولوژی تا حد ممکن پیشنهاد می‌دهد که از مدل‌سازی و طراحی پرهیز شود و مبنای پیشرفت پروژه را همانند دیگر متدولوژی‌های چابک، سیستم ساخته شده می‌داند. اما توسعه‌دهندگان می‌توانند برای بیان موارد پیچیده از زبان‌های مدل‌سازی مانند **UML** استفاده نمایند. برنامه‌نویسی، تست و دیپلوی به صورت ویژه‌ای در این متدولوژی مورد توجه قرار گرفته‌اند. تست نرم‌افزار برای تضمین کیفیت آن نقش مهمی ایجاد می‌کند. همچنین در انتهای هر اسپرینت یک محصول

قابل استفاده که تا حد امکان قابلیت‌ها با اطمینان بالایی عمل می‌کنند، به مشتری تحویل داده می‌شود و عمل دیپلوی صورت می‌گیرد. فعالیت نگه‌داری و پشتیبانی از سیستم نیز به صورت فاز جداگانه‌ای در این متدولوژی بیان نشده است اما با تعریف چند اسپرینت برای پشتیبانی از محصول و اضافه سازی قابلیت در فاز نگه‌داری، می‌توانند فاز *maintenance* را فراهم سازند.

متدولوژی کریستال کلیر

این متدولوژی به خوبی فعالیت‌های چرخه تولید نرم افزار را پوشانده است. برای مدلسازی و شناسایی فضای مسئله، استخراج نیازمندی‌ها و تحلیل امکان‌سنجی در فاز *chartering* فعالیت‌هایی قرار داده شده است. در این فاز در فعالیت *Exploratory 360 degrees* ابتدا طرح کسب و کار سیستم شرح داده می‌شود. سپس نیازمندی‌های سطح بالای سیستم استخراج می‌گردد. در گام سوم مدلسازی فضای مسئله انجام می‌شود. همچنین امکان‌سنجی پروژه و مسائل اصلی که باعث موفقیت یا شکست پروژه می‌گردد، استخراج می‌شود. از فعالیت‌های توسعه سیستم، طراحی معماری و طراحی جزئی سیستم در این متدولوژی بیان نشده‌اند. دلیل نیز آن است که سیستم به صورت *iterative* و در طی *iteration*‌های مختلف ساخته می‌شود. اما فعالیت‌های برنامه‌نویسی، تست و دیپلوی به صورت مفصل و دقیق در فاز *cyclic delivery* انجام می‌گیرند. در این فاز دیپلوی سیستم و نمایش به کاربر انجام می‌گیرد تا بازخوردها را از وی دریافت نمایند. فعالیت نگه‌داری سیستم در این متدولوژی بیان نشده است و تیم توسعه در طی *iteration*‌هایی این کار را بایستی خود اضافه نمایند به متدولوژی.

در کل می‌توان گفت که این متدولوژی نمره بالایی در پوشش فعالیت‌های توسعه نرم افزار می‌گیرد.

مقایسه متدولوژی اسکرام با کریستال کلیر

هر دو متدولوژی از دسته متدولوژی‌های چابک هستند و تمرکز اصلی خود را بر راندمان بالای تیم و توسعه سریع نرم افزار می‌گذارند. پوشش فعالیت‌های چرخه تولید نرم افزار در این دو متدولوژی تا حد زیادی به یکدیگر شبیه می‌باشد. اما اگر بخواهیم یک متدولوژی را برتر بنامیم، متدولوژی کریستال به دلیل پوشش بیشتر و بهتر فعالیت‌های *definition* در رتبه بالاتر قرار می‌گیرد. دلیل نیز وجود گام *Exploratory 360 degrees* می‌باشد که به صورت دقیق تحلیل امکان‌سنجی، استخراج نیازمندی و مدلسازی فضای مسئله را انجام می‌دهد. در دیگر فعالیت‌های این دو متدولوژی بسیار با یکدیگر شبیه می‌باشند.

معیار سوم) پشتیبانی از فعالیت‌های چتری توسعه نرم افزار

متدولوژی اسکرام

فعالیت‌های چتری توسعه نرم‌افزار به خوبی در متدولوژی اسکرام دیده شده‌اند. این فعالیت‌ها عبارتند از مدیریت ریسک، مدیریت پروژه و تضمین کیفیت که در ادامه بخش بررسی می‌نماییم.

مدیریت ریسک به دلیل ذات *iterative incremental* این متدولوژی به خوبی دیده شده است. به دلیل اینکه ساخت محصول طی اسپرینت‌های مختلف صورت می‌گیرد، در هر اسپرینت تکه‌ای از محصول ساخته می‌شود. در انتهای اسپرینت در جلسه‌ی مرور اسپرینت، محصول توسعه یافته به ذی‌نفعان نمایش داده می‌شود و ذی‌نفعان می‌توانند هم بازخوردهای خود را برای حداقل رساندن ریسک اعلام کنند و هم اینکه تصمیم‌گیری کنند آیا ادامه پروژه سودآوری خواهد داشت یا خیر. از فعالیت‌هایی که برای مینیمم کردن ریسک استفاده می‌شود که در اسکرام موجود هستند می‌توان به موارد زیر اشاره نمود:

نمونه‌سازی به وسیله ساخت نمونه اولیه برای امکان‌سنجی راه‌حل - برنامه‌ریزی مبتنی بر ریسک به وسیله مشخص‌سازی اولویت کارها در بک‌لاگ محصول - فرایند *iterative incremental* به خاطر وجود مفهوم اسپرینت در متن این متدولوژی - مشارکت فعالانه کاربر به دلیل جلسه مرور اسپرینت در انتهای هر اسپرینت - انتشار سریع محصول در انتهای هر اسپرینت - اعتبارسنجی مستمر محصول توسط اعضای پروژه - مرور به صورت مداوم محصول در جلسه مرور اسپرینت و ادغام و یکپارچگی محصول در انتهای اسپرینت.

مدیریت پروژه در این متدولوژی در بالاترین سطح ممکن انجام می‌گیرد. به وسیله برنامه‌ریزی دقیق هر اسپرینت در شروع آن یک برنامه منسجم برای انجام کارها تا انتهای اسپرینت مشخص می‌شود. سپس به صورت مداوم در زمان اسپرینت به وسیله اسکرام مستر، تطابق فرایند و فعالیت‌ها با قوانین و اصول اسکرام سنجد می‌شود. همچنین در جلسه انتهای هر اسپرینت به نام مرور اسپرینت، کنترل پروژه به صورت دقیقی صورت می‌گیرد. همچنین فعالیت جلسه استندآپ در هر روز اسکرام بسیار به تعاملات درون تیمی کمک می‌کند که خود این کار باعث می‌شود مدیریت پروژه و آگاهی افراد مختلف از پروژه بسیار افزایش یابد.

کیفیت در متدولوژی‌های چابک در بالاترین سطح قرار دارد و چیزی است که به هیچ وجه ذره‌ای از آن کم نمی‌گردد. این مورد درباره متدولوژی اسکرام نیز صادق است. در متدولوژی اسکرام به صورت مداوم کنترل کیفیت پروژه چه به صورت فنی و چه با استفاده از نظرات ذی‌نفعان صورت می‌گیرد. جلسه مرور اسپرینت و *retrospective* دقیقاً به همین منظور قرار داده شده‌اند تا به صورت بسیار سریع موارد تاثیرگذار بر کیفیت شناسایی شده و برطرف گردند. اعتبارسنجی مداوم و پیوسته یکی دیگر از تکنیک‌های مورد استفاده در اسکرام است. از طرفی دیگر نیز بک‌لاگ محصول و اسپرینت قابلیت ردیابی به نیازمندی‌ها را برای تیم فراهم می‌نمایند. بدین صورت در انتهای انجام یک وظیفه یا مورد از بک‌لاگ در صورتی

که آن ویژگی پیاده‌سازی شده انتظارات را برآورده نکرده باشد، به سرعت آن مورد کشف شده و در جهت بهبود آن اقدام می‌شود.

متدولوژی کریستال کلیر

فعالیت های چتری در این متدولوژی به صورت دقیقی بیان شده اند. برای مدیریت ریسک، فعالیت های تحلیل امکان سنجی (در فعالیت Exploratory 360 degrees)، برنامه‌ریزی(در ابتدای هر iteration)، دخیل کردن مشتری(در انتهای هر iteration نمایش به مشتری صورت می گیرد)، وجود ذات iterative incremental و یکپارچه‌سازی پیوسته انجام می‌گیرند. همچنین اعتبارسنجی سیستم با نمایش به کاربر به صورت مداوم انجام می‌گیرد تا سیستم مدنظر وی ساخته شود.

برای مدیریت پروژه در این متدولوژی تمهیدات خوبی اندیشیده شده است. در ابتدای هر iteration یک جلسه برای برنامه ریزی برقرار می‌شود. همچنین در هر روز در جلسه استندآپ به مرور فعالیت های پرداخته می‌شود. در انتهای هر اسپرینت نیز به صورت دقیق فرایندها و نحوه انجام کارها بررسی می‌شود تا بهبود در متدولوژی به صورت پیوسته صورت گیرد. از نظر ارتباطات افراد درون تیم نیز این متدولوژی به صورت بسیار خوب این کار را انجام می‌دهد تا راندمان افراد به حداکثر خود برسد. برای این کار تیم بایستی در نزدیکی هم قرار گیرد و در جلسات استندآپ به صورت مداوم با یکدیگر در ارتباط باشند.

تضمین کیفیت به وسیله یکپارچه سازی پیوسته و محوری بودن نیازمندی ها انجام می‌گیرد. در انتهای هر iteration اعضای تیم و ذی نفعان کنار یکدیگر جمع می‌گردند تا کیفیت محصول بررسی شود.

مقایسه متدولوژی اسکرام با کریستال کلیر

هر دو متدولوژی به صورت دقیق و با تاکید بسیار بر هر سه فعالیت مدیریت ریسک، مدیریت پروژه و تضمین کیفیت تمرکز کرده اند. متدولوژی اسکرام اما در مدیریت ریسک به دلیل نداشتن تحلیل امکان سنجی از کریستال ضعیف تر است. اما در تضمین کیفیت و مدیریت پروژه هر دو به صورت بسیار مشابهی این دو فعالیت را انجام می‌دهند. هر دو متدولوژی در iteration های مختلفی توسعه سیستم را انجام می‌دهند. همچنین وجود جلسات استندآپ روزانه و نمایش سیستم به کاربر در انتهای هر اسپرینت از جمله موارد شباهت هر دو متدولوژی می‌باشد. در مجموع می‌توان گفت هر دو متدولوژی بسیار مشابه یکدیگر هستند و نمی‌توان یکی را به صورت برتر از دیگری در نظر گرفت.

معیار چهارم) بی درزی و همواری

متدولوژی اسکرام

متدولوژی اسکرام لزوماً یک متدولوژی بی درز نمی‌باشد. برای مدلسازی و ثبت نیازمندی‌ها از بک‌لاگ محصول استفاده می‌شود. در این بک‌لاگ مواردی اعم از ویژگی‌ها، قابلیت‌ها، فیکس‌ها، خطاها، نیازمندی‌های جدید، بهبودها و ... می‌باشد. سپس تیم توسعه در ابتدای هر اسپرینت جلسه برنامه‌ریزی برای اسپرینت دارد. در این جلسه موارد با اهمیت بالا برداشته شده و در بک‌لاگ اسپرینت قرار داده می‌شود. در طی زمان اسپرینت، بر مبنای موارد طراحی شده و قرارداده شده در بک‌لاگ کارها و برنامه‌نویسی انجام می‌گیرد. همچنین نکته مهم دیگر وجود مفهوم اسپرینت است که توسعه محصول به صورت آرام و پیوسته صورت می‌گیرد. در نتیجه می‌توان گفت بی‌درزی در این متدولوژی به دلیل پرش از نیازمندی‌های ذکر شده در بک‌لاگ به پیاده‌سازی مستقیم کد، تا حدی از بین می‌رود. در این بین نیز مدلسازی صورت نمی‌گیرد و درز به همین دلیل افزایش می‌یابد. اما به دلیل ذات *iterative* این متدولوژی، همواری تا حد بسیار خوب و بالایی وجود دارد. وجود همواری در این متدولوژی بسیار نکته مثبت و قابل توجهی می‌باشد. اما وجود درز در متدولوژی باعث می‌شود که برنامه‌نویسان کمی دشواری در تولید سیستم نهایی داشته باشند.

متدولوژی کریستال کلیر

این متدولوژی دارای خاصیت همواری میان فعالیت‌ها می‌باشد. دلیل آن نیز وجود ذات و ماهیت *iterative incremental* در آن است. فعالیت‌های به صورت آرام و هموار انجام می‌گیرند تا به مرور سیستم ساخته شود. اما بی‌درزی در این متدولوژی به دلیل دره میان نیازمندی‌ها و کدزنی وجود دارد. بعد از مدلسازی نیازمندی‌ها و فضای مسئله سپس ساخت و کدزنی سیستم انجام می‌شود. این باعث می‌شود که دچار شیفت در پارادایم شویم.

مقایسه متدولوژی اسکرام با کریستال کلیر

همانگونه که می‌دانیم، متدولوژی‌های چابک از نظر همواری بسیار خوب هستند. دلیل نیز آن است که توسعه سیستم را به صورت *iterative incremental* و آرام انجام می‌دهند. اما به دلیل آنکه استخراج نیازمندی‌ها در قالب *user story* یا آیتم‌های بک‌لاگ انجام می‌شود و سپس به صورت مستقیم به کدزنی پرداخته می‌شود، بی‌درزی لطمه و صدمه می‌خورد. در نتیجه از نظر بی‌درزی ضعیف می‌باشند. این دو متدولوژی نیز از این موضوع مستثنی نیستند و این معایب را دارند.

در مجموع می توان گفت در این معیار هردو قوتها و ضعفهایی دارند و با یکدیگر در یک مرتبه قرار می گیرند.

معیار پنجم) محوریت بر نیازمندی های عملکردی و غیرعملکردی

متدولوژی اسکرام

نیازمندی ها در متدولوژی اسکرام توسط صاحب محصول مشخص شده و در بک لاگ محصول قرار داده می شود. این بک لاگ شامل ویژگی ها و قابلیت های مدنظر برای پیاده سازی محصول می باشد. سپس در طی زمان توسعه محصول به مرور این بک لاگ کامل تر می شود. ویژگی های جدید به آن افزوده می گردد، برخی ویژگی های کنونی دستخوش تغییرات می شود و همچنین بهبودهای فنی و باگ فیکس ها به آن اضافه می شود. مدیر محصول وظیفه نگهداری و تکمیل این موارد را برای بک لاگ محصول دارد. در این بک لاگ موارد با اهمیت بالاتر در بالای بک لاگ قرار می گیرد و موارد با درجه ضرورت پایین تر در پایین لیست قرار داده می شود. فعالیت مهم برای سازمان دهی بک لاگ به نام grooming می باشد. در این فعالیت آیتم های درون بک لاگ بر اساس ارزش، هزینه و ریسک مرتب سازی می شود و تخمینی از هزینه و سایر انجام آن ها قرار داده می شود. بدین صورت در آغاز هر اسپرینت می توان وظایف مهم تر که در قالب یک اسپرینت قابل انجام هستند را انتخاب کرده و با برنامه ریزی آن ها را انجام داد.

همان گونه که بیان شد، تنها مرحله اصلی قبل از پیاده سازی نرم افزار، مشخص سازی نیازمندی ها و قابلیت ها در قالب بک لاگ محصول و اسپرینت می باشد. سپس مرحله پیاده سازی و کدزنی براساس همین موارد مشخص شده انجام می گیرد. و از طرفی چون هم نیازمندی های عملکردی و هم نیازمندی های غیرعملکردی مانند سرعت، قابل تست بودن، امنیت و ... به خوبی لحاظ می شود، این متدولوژی بر محور نیازمندی ها جلو می رود. همچنین در آخر هر اسپرینت محصول تولید شده به ذی نفعان و مشتریان عرضه می شود تا آنها نیز بازخورد خود را ارائه کنند و اعتبارسنجی موارد صورت بگیرد. در مجموع می توان گفت نمره این متدولوژی در قابلیت محوریت بر نیازمندی ها نمره بالایی می باشد.

متدولوژی کریستال کلیر

این متدولوژی همانند اکثر متدولوژی های چابک دیگر، تمرکز و تاکید ویژه ای بر استخراج و مدلسازی نیازمندی ها دارد. در فعالیت Exploratory 360 degrees نیازمندی های سطح بالای سیستم مشخص می گردد. سپس در فاز دوم پروژه، در فعالیت Recalibrate the release plan به صورت جزئی و دقیق کارها و فعالیت های متناسب با هر نیازمندی مشخص می گردد. بدین صورت تیم توسعه دهنده محصول بر مبنای این برنامه مشخص شده، شروع به ساخت محصول می کند.

این متدولوژی همانند دیگر متدولوژی‌های چابک دقت ویژه‌ای بر نیازمندی‌های غیرعملکردی سیستم دارد. دلیل نیز آن است که هم در فعالیت مشخص سازی تکنولوژی‌ها و هم در هنگام نمایش سیستم به کاربر سفیر، این موارد را لحاظ کرده و هنگام ساخت سیستم رسیدن به آن را الزامی می‌کند.

مقایسه متدولوژی اسکرام با کریستال کلیر

متدولوژی اسکرام در این مورد قوی‌تر عمل می‌کند. دلیل آن است که یک نقش به تیم اضافه شده است به نام صاحب محصول که فعالیت اصلی وی مشخص کردن و اولویت بندی نیازمندی‌ها می‌باشد. این شخص وظیفه دارد به صورت مستمر چک کند که آیا سیستم ساخته شده بر طبق نیازمندی‌های عملکردی و غیرعملکردی مشخص شده عمل می‌کند یا خیر. همچنین در ابتدای هر اسپرینت به صورت مفصل به مرتب کردن و بیان جزئیات هر نیازمندی که به صورت یک آیتم در بک‌لاگ بیان شده است، می‌پردازد.

متدولوژی کریستال کلیر نیز این فعالیت‌ها را به خوبی انجام می‌دهد ولی نبود یک نقش جداگانه برای بررسی و تشریح نیازمندی‌ها، باعث شده است که از متدولوژی اسکرام عقب بیافتد.

معيار ششم) قابل تست بودن، ملموس بودن و قابل ردیابی به نیازمندی ها

متدولوژی اسکرام

معيار قابل تست بودن محصولات به سادگی، تعداد کم و قابل فهم بودن مصنوعات اعم از مدل ها و کدهای تولید شده برمی گردد. متدولوژی اسکرام تا حد زیادی نسبت به مدل سازی مقاوم است و اجازه مدل سازی را تا جای ممکن نمی دهد. به همین دلیل تعداد مصنوعات و مدل های تولید شده بسیار کم می باشد. همچنین مدل ها تنها برای موارد دشوار و پیچیدگی های بالا به کار می روند. این باعث می گردد که مصنوعات قابلیت تست بالا داشته باشند. از جهتی نوشتن تست برای سیستم که یکی از ارکان مهم کیفیت در متدولوژی های چابک می باشد، در اسکرام بسیار توصیه می گردد. پس کد تولید شده برای نرم افزار نیز قابلیت تست بالایی دارد.

محصولات تولید شده در این متدولوژی بسیار کم می باشند. از طرفی در انتهای هر اسپرینت سیستم و نرم افزار تولید شده به کاربر نهایی و ذی نفعان نمایش داده می شود تا بازخوردها به صورت سریع و دقیق گرفته شوند تا با استفاده از آن بک لاگ سیستم تکمیل گردد. این مورد که در انتهای هر اسپرینت محصولی قابل عرضه تولید شده است، بسیار به ملموس بودن برای کاربر و تیم کمک می کند. در نتیجه هم اعضای تیم خوشحال تر در توسعه سیستم خواهند بود و هم اینکه مشتری ها اطلاع خوبی از وضعیت پروژه خواهند داشت. همچنین کم بودن تعداد نمودارها و مدل های تولید شده نیز کمک به بالاتر بودن امتیاز این متدولوژی در این معیار خواهد بود.

از طرفی دیگر به دلیل پایه بودن بک لاگ در توسعه نرم افزار، نمودارها و کدهای تولید شده برای سیستم قابلیت ردیابی مستقیم به نیازمندی ها را خواهد داشت. همانطور که در قبل بیان شد، ویژگی ها و قابلیت های در اولویت بالا برای تولید در بک لاگ محصول و اسپرینت قرار می گیرند و تمام پیاده سازی ها بر اساس بک لاگ صورت می گیرد. موارد موجود در بک لاگ نیز به صورت مستقیم نیازمندی های سیستم را بیان می کنند. در نتیجه ردیابی به نیازمندی ها در این متدولوژی در سطح بالایی برقرار می باشد.

متدولوژی کریستال کلیر

قابل تست بودن محصولات و مصنوعات متدولوژی در کریستال کلیر برقرار است. دلیل نیز آن است که در این متدولوژی تعداد نمودارها بسیار اندک می باشد و تمرکز اصلی بر توسعه کد و ساخت سیستم می باشد. به همین دلیل تا جای ممکن محصولات جانبی اندک هستند. قابلیت ملموس بودن این محصولات نیز هم برای کاربر و هم برای توسعه دهنده بسیار بالاست. دلیل آن است که مهم ترین محصول تولید شده در این متدولوژی سیستم تحت ساخت است و این محصول به صورت مرتب و منظم در انتهای هر iteration به کاربر نمایش داده می شود. همچنین توسعه دهنده سیستم نیز لازم نیست درگیر ساخت نمودارهای پر حجم و کم فایده گردد.

به دلیل آنکه در ابتدای هر iteration نیازمندی‌ها تکمیل و جزئی سازی می شود، قابلیت ردیابی کدهای زده شده به نیازمندی‌ها بسیار افزایش می‌یابد. خوبی این مورد آن است که در هنگام تولید کد و سپس در جلسه مرور محصول می‌توان بررسی کرد که هر قابلیت نشات گرفته از کدام نیازمندی است.

مقایسه متدولوژی اسکرام با کریستال کلیر

به دلیل آنکه هر دو متدولوژی چابک هستند، بسیار در تولید محصولات شبیه یکدیگر می‌باشند. دلیل نیز اصول و فلسفه موجود در مانیفست چابک است. به همین دلیل نمی‌توان یکی از این دو متدولوژی را در این معیار به صورت مطلق بالاتر دانست و امتیاز بیشتری به آن داد.

هر دو متدولوژی هر سه قابلیت قابل تست بودن، ملموس بودن و قابلیت ردیابی به نیازمندی را در حد بالایی برقرار می‌کنند.

معيار هفتم) مشاركت فعال مشتری

متدولوژی اسكرام

متدولوژی اسكرام اهميت بسيار زيادی به مشتری و نظرات وی می‌دهد. در هر اسپرینت یک جلسه مرور اسپرینت برگزار می‌شود تا محصول پياده‌سازی شده به کاربر و ذی‌نفعان دمو داده شود. در نتیجه اعتبارسنجی برای تولید سیستم مدنظر مشتری به صورت مداوم و از ابتدا صورت می‌گیرد. همچنین نقش‌های مالک محصول و اسكرام مستر در جهت اتصال بين مشتری و تیم توسعه در متدولوژی اسكرام تعريف شده اند. مالک محصول شخصی است که از طرف مشتری موضع گیری می‌کند و به صورت جدی محصول تولید شده را ارزیابی می‌نماید تا نقص و خلل در آن کمینه باشد. همچنین اسكرام مستر شخصی است که پل ارتباطی بين تیم و سازمان می‌باشد تا بازخوردها را گرفته و در اصلاح فرایندها بکوشد. بدین ترتیب نظرات مشتری‌ها و کاربران تا حد زيادی به تیم منعکس می‌گردد تا محصول تولید شده از سطح کیفی بالایی برخوردار گردد.

در انتهای هر اسپرینت لازم است محصول تولید شده به سطح خوبی از **stable** بودن رسیده باشد که کاربر قابلیت استفاده از آن را داشته باشد. این باعث می‌گردد تا محصول مدنظر مشتری در حال ساخت و تکمیل شدن باشد. همچنین متدولوژی اسكرام شبیه دیگر متدولوژی‌های چابک، به تغییرات با روی باز واکنش نشان می‌دهد تا در صورت تعریف نیازمندی جدید، آن را پياده‌سازی نماید. این فعالیت‌ها باعث می‌گردد تا مدیریت ریسک تا حد بسیار خوبی انجام گیرد و در صورتی که قابلیت درجه اهميت بالاتری داشته باشد، اولویت را بتوان عوض کرده و آن را سریع تر پياده سازی نمود و در نتیجه در بازار بسیار رقابتی از رقبای عقب نیفتاد.

متدولوژی کریستال کلیر

متدولوژی کریستال کلیر اهميت فوق العاده‌ای به وجود کاربر سفیر در تیم می‌دهد. از همان ابتدای چرخه تولید نرم افزار و در فعالیت انتخاب نقش های تیم، یک نقش به نام کاربر سفیر تعیین می‌گردد که به نوعی خبره استفاده از سیستم است و به صورت مداوم وظیفه دارد سیستم توسعه داده شده را ارزیابی نماید.

همچنین در **iteration** های تولید محصول، در فعالیت‌های تحویل سیستم به مشتری و ورکشاپ انتهای هر **iteration** به صورت کامل و مفصل سیستم فعلی دمو داده می‌شود تا فیدبک‌های لازمه دریافت شود. سپس نیازمندی‌ها اصلاح شده و اگر ویژگی‌ای از سیستم بود که طبق نظر مشتری پياده‌سازی نشده بود، اصلاح گردد.

مقایسه متدولوژی اسکرام با کریستال کلیر

هر دو متدولوژی سعی کرده اند در انتهای هر iteration از تولید محصول، کاربر را در ماجرا دخیل کرده و به صورت early از نظرات وی مطلع گردند. این فعالیت را نیز در انتهای هر دوره از تولید محصول قرار داده‌اند.

همچنین هر دو متدولوژی با اضافه کردن نقش کاربر سفیر (البته با نام‌های ظاهراً متفاوت) در پی این هدف بوده‌اند تا نظرات وی اعمال شده و اگر اعتبارسنجی سیستم به درستی پاس نشد، سریعاً نیازمندی‌ها و قابلیت‌های پیاده‌سازی شده تغییر کنند تا درصد موفقیت سیستم افزایش یابد.

در نتیجه به دلیل فوق‌العاده بودن هر دو متدولوژی در این ویژگی نمی‌توان یکی را بر دیگری ترجیح داد و هر دو در یکی سطح به این مسئله مهم پرداخته‌اند.

معیار هشتم) practical و practicable بودن

متدولوژی اسکرام

متدولوژی اسکرام بسیار بر کیفیت انجام کار و خروجی تاکید دارد و در این میان از هرگونه فعالیت اضافی و دست و پاگیر اجتناب می کند. به همین دلیل پیچیدگی فعالیت ها و جریان کار بسیار پایین می باشد. همچنین در جلسه retrospective انتهای هر اسپرینت بررسی می شود تا موانع و مشکلات شناسایی و برطرف گردد. همچنین به دلیل مدیریت پروژه قوی در این متدولوژی هر دو معیار practical و practicable بودن تا حد زیادی برقرار می گردد.

جلسات استنداپ روزانه و وجود بک لاگ برای انجام کارها باعث تمرکز بیشتر اعضای تیم و در جریان قرار گرفتن ایشان می گردد. همچنین فعالیت های اضافی و مدلسازی های وقت گیر و کم اثر تا حد زیادی حذف شده است. در مجموع با توأصیف ذکر شده، این دو معیار بسیار امتیاز بالایی در این متدولوژی می گیرند و همین گونه که از گزارشات تیم های مختلف برداشت می شود، امروزه این متدولوژی یکی از پر استفاده ترین و پر اقبال ترین متدولوژی های مورد استفاده در شرکت های نرم افزاری می باشد.

متدولوژی کریستال کلیر

همانگونه که در اولین بخش ذکر کردیم، اعضای متدولوژی کریستال برای پوشش پروژه های با سطح بحرانی متفاوت و با تیم های با سایزهای گوناگون درست شده اند. این موارد به صورت انتزاعی و سطح بالا در متدولوژی بیان شده اند و برخی از این متدولوژی ها (همانند کریستال قرمز و آبی) در واقعیت به وجود نیامده و استفاده نشده اند. به همین دلیل می توان گفت تمامی اعضای این خانواده این متدولوژی دارای ویژگی های practicality و practicability نیستند و تنها سه عضو کریستال کلیر، اورنج و اورنج وب از این موضوع مستثنی هستند.

اما متدولوژی کریستال کلیر به دلیل استفاده زیاد از آن در پروژه های شش نفره با سایز کوچک، قوت و میزان استفاده از خود را اثبات کرده است. این متدولوژی دارای فرایندهای ساده، توجه به شرایط جاری در پروژه و حاوی اصول چابکی است. از این رو به یکی از انتخاب های متداول برای توسعه سیستم های نرم افزاری بدل شده است.

مقایسه متدولوژی اسکرام با کریستال کلیر

متدولوژی اسکرام یکی از پرکاربردترین متدولوژی‌های امروزه است. این متدولوژی با دخیل کردن کاربر در فرایندهای توسعه نرم‌افزار و دقت بر توسعه سریع و با راندمان بالای سیستم توانسته است به اولین انتخاب در میان تیم‌های تولید کننده نرم‌افزار مبدل شود.

همچنین این متدولوژی قابلیت پشتیبانی از تیم‌های با تعداد نفرات بالا را (برخلاف کریستال کلیر) داراست. فعالیت‌های مدیریت پروژه و نظارت بر فرایند نیز به صورت بسیار مناسبی در این متدولوژی صورت می‌گیرند. در انتها می‌توان گفت این متدولوژی در قابلیت استفاده و practicality به نسبت کریستال کلیر در مرتبه بالاتری قرار دارد و انتخاب شدن آن به اولین گزینه نیز شاهدهی بر این مدعاست.

معيار نهم) قابليت مديریت پيچيدگي

متدولوژی اسكرام

در تيم اسكرام نقش های مختلفي وجود دارد. اين موضوع باعث می گردد كه فعاليت ها جدای از هم انجام گیرند و تا حد امکان شكست كار در حوزه های مختلف به خوبی صورت پذیرد. صاحب محصول وظيفه مديریت كارها و اولويت شان را برعهده دارد. اسكرام مستر وظيفه مديریت و تطابق با اسكرام را برعهده دارد و تيم توسعه نیز به ساخت و تکميل محصول می پردازد.

از طرفی ديگر ساخت پروژه به چندین اسپرینت شكسته می شود. با برنامه ریزی ابتدای اسپرینت تسك ها شكسته شده و وظيفه هر شخص به صورت دقيق تعیین می گردد. همچنين در جلسات روزانه اسپرینت، مشكلات و پيچيدگي ها بيان می شود تا اعضای تيم با كمك يكدیگر آن را برطرف نمایند.

وجود بک لاگ برای محصول و هر اسپرینت یکی از مهم ترین مفاهيم برای مديریت پيچيدگي می باشد. كارها و ویژگی های مد نظر برای پیاده سازی ابتدا به صورت درشت دانه هستند اما به مرور شكسته شده و ملموس تر می گردند. در انتها كارهایی كه به اشخاص محول می گردد به صورت كاملاً شفاف و ریزدانه هستند كه در مدت كمی قابل انجام هستند. صاحب محصول نیز در مديریت میزان پيچيدگي نقش مهمی ایفا می کنند.

متدولوژی اسكرام به صورت بسیار خوبی مديریت پيچيدگي كارها را انجام میدهد و نمره بسیار قابل قبولی در اين معيار كسب می نماید.

متدولوژی كريستال كلير

متدولوژی كريستال كلير برای استفاده در تيم های با نهايت سايز ۶ نفره مناسب می باشد. از اين رو قابليت مديریت پيچيدگي در آن به صورت محدود موجود می باشد. اما در همین تعداد افراد نیز با شكندن وظايف تيم به افراد مختلف، باعث می شود وظايف به خوبی انجام گیرند. همچنين اين متدولوژی توسعه محصول را به iteration هایی می شكند. اين كار باعث می شود بخشی از سيستم در هر گام توليد شده و بر مبنای آن فعاليت های آتی صورت گیرد. اين كار به صورت سلسله مراتبی دوباره صورت می گیرد تا هر iteration نیز خود به روزهای مختلف تقسيم می شود كه در آن وظيفه ساخت محصول و تست آن به افراد محول شده است.

مقایسه متدولوژی اسکرام با کریستال کلیر

هر دو متدولوژی به خوبی پیچیدگی را مدیریت می‌کنند. این کار به وسیله دو نکته اصلی صورت می‌گیرد. اول شکستن وظایف اعضای تیم به نقش‌های مختلف. دوم به شکستن بازه‌های تولید سیستم به iteration ها و اسپرینت‌های متفاوت. اگر بخواهیم یکی از این دو متدولوژی را برتر بدانیم، انتخاب اسکرام خواهد بود. این متدولوژی قابلیت پشتیبانی از پروژه‌های با سایزهای بزرگ‌تر را داراست. وجود بک‌لاگ‌های مختلف اعم از دو نوع بک‌لاگ محصول و بک‌لاگ اسپرینت نیز باعث می‌شود سلسله مراتبی ویژگی‌های سیستم به خوبی تعریف شود و پیاده‌سازی هرکدام به نحو بسیار خوبی انجام شود. حتی از اسکرام برای مدیریت تیم‌های با سایز ۳۰۰ نفر نیز (اما با در نظر گرفتن ملاحظات) استفاده شده است که خود نشانگر قدرت بالاتری این متدولوژی در مدیریت پیچیدگی پروژه می‌باشد.

معیار دهم) قابلیت گسترش / مقیاس پذیر / قابلیت شخصی سازی / انعطاف پذیری

متدولوژی اسکرام

قابلیت گسترش در این متدولوژی وجود دارد. اسکرام به صورت کلی تشکیل شده است از یک سری نقش، فعالیت، مصنوعات و قوانین. هر تیمی در استفاده از اسکرام با توجه به نوع و ذات پروژه تحت ساخت، می تواند شخصی سازی انجام دهد و مطابق با نیازهای خود قوانینی اضافه کند یا فرایندها را تغییر دهد.

اما این متدولوژی قابلیت مقیاس پذیری را پشتیبانی نمی کند. تیم های اسکرام از حدود ده نفر تشکیل می شوند و اگر تعداد اعضای تیم افزایش یابد، بایستی اعضا را به تیم های ده نفره شکاند. در این صورت با صرف هزینه بسیار زیادتر و اضافه سازی افرادی برای نظارت و پایش هماهنگی بین تیم ها امکان استفاده از اسکرام را به وجود آورد. پس بسیار منطقی است از متدولوژی هایی مانند RUP و OPM در این گونه از پروژه ها که نیازمند تعداد اعضای زیاد است، استفاده نمود.

همانگونه که اشاره شد قابلیت شخصی سازی و انعطاف پذیری نیز به صورت بسیار خوبی در این متدولوژی وجود دارد. در جلسات استندآپ روزانه و جلسه retrospective در انتهای هر اسپرینت نظرات افراد گرفته می شود تا مشکلات در اولین زمان ممکن برطرف گردند. بعد از شناسایی مشکلات و نقاط قوت احتمالی، فرایندها و فعالیت ها به نحوی تغییر می یابند که سرعت انجام کار و پیشرفت پروژه تا حد ممکن زیاد گردد.

متدولوژی کریستال کلیور

خانواده متدولوژی های کریستال دارای قابلیت مقیاس پذیری می باشد. از این جهت که توانایی استفاده در پروژه های با سطح بحرانی متفاوت را داراست. برای این کار نیز لازم است مدلسازی به متدولوژی اضافه گردد. همچنین قابلیت گسترش و مقیاس پذیری برای استفاده در تیم های با سایزهای مختلف نیز وجود دارد (رنگ های مختلف متدولوژی ها). پس در کل می توان گفت خانواده این متدولوژی ها در دو قابلیت قابل گسترش بودن و مقیاس پذیری بسیار مناسب هستند. اما خود متدولوژی کریستال کلیور به صورت خاص از این دو ویژگی برخوردار نیست. این متدولوژی فقط برای تیم های ۶ نفره با سطح بحرانی C و D قابل استفاده است و حتی E را نیز پشتیبانی نمی کند.

اما دو قابلیت شخصی سازی و انعطاف پذیری بسیار خوب در این متدولوژی لحاظ شده اند. در ابتدای شروع پروژه می توان متدولوژی را شخصی سازی کرده و مطابق با نیازها تعریف نمود (در فاز chartering به صورت مفصل این کار انجام می شود و در آخر بر روی آن توافق صورت می گیرد). همچنین پس از اتمام هر iteration تیم یک جلسه ورکشاپ برگزار می کند تا در آن نحوه انجام کارها مشاهده شود و نقایص برطرف شود. همچنین اگر نیاز است تغییراتی در متدولوژی و فرایندهای آن صورت گیرد تا سرعت توسعه بسیار افزایش یابد.

مقایسه متدولوژی اسکرام با کریستال کلیر

متدولوژی کریستال در این چهار مورد از اسکرام برتر است. در متدولوژی‌های کریستال می‌توان قابلیت مقیاس‌پذیری و گسترش‌پذیری را به دلیل وجود رنگ‌های مختلف و متدولوژی‌های با تیم‌های گوناگون به آن دست یافت. اما در اسکرام سائیز تیم محدود است.

همچنین قابلیت گسترش‌پذیری، انعطاف‌پذیری و شخصی‌سازی نیز در کریستال بهتر بیان شده است. در متدولوژی کریستال می‌توان در ابتدای شروع پروژه بر جزئیات نحوه استفاده توافق کرده و به یک مورد مدنظر همه افراد رسید. همچنین در انتهای هر iteration نیز جلسه ورکشاپ اجازه تغییر در جزئیات را با دست باز به افراد استفاده‌کننده از آن می‌دهد.

معیار یازدهم) حوزه کاربرد (Application scope)

متدولوژی اسکرام

متدولوژی اسکرام یک متدولوژی چابک می باشد. این دسته از متدولوژی ها برای پروژه های با **criticality** بالا اصلا مناسب نیستند و فقط قابلیت استفاده در سطوح C ، D ، و نهایتا E را دارند. دلیل نیز آن است که از مدلسازی تا حد زیادی پرهیز می کنند.

اما امروزه این متدولوژی برای استفاده در سیستم های تجاری روزمره گزینه اول می باشد. این متدولوژی به دلیل فعالیت ها و قوانینی که دارد و در بخش های قبل به تفصیل بیان شد، بسیار محبوب و رایج است. کتابها و مقالات زیادی نیز درباره تجربیات کسب شده از استفاده از این متدولوژی چاپ شده اند. این متدولوژی در سیستم های **data intensive** به خوبی قابلیت استفاده و به کارگیری را دارد.

متدولوژی کریستال کلیر

متدولوژی کریستال تنها قابلیت استفاده در سطوح C و D را داراست. همچنین تیم های حداکثر ۶ نفره را پشتیبانی می کند. مورد اصلی استفاده از این متدولوژی در سیستم های تجاری مرسوم می باشد. از این رو از یک طرف دارای محدودیت است و از طرفی دیگر سیستم های تجاری مرسوم بازه وسیعی از سیستم ها را در برمی گیرد.

مقایسه متدولوژی اسکرام با کریستال کلیر

هر دو متدولوژی دارای نقص در پوشش سیستم های بحرانی که جان انسان در خطر می افتد هستند. برای مثال قابلیت استفاده در توسعه سیستم کامپیوتری هواپیما و ماشین را نخواهند داشت. اما از طرفی دیگر بسیاری از سیستم های مرسوم امروزی را می توانند پشتیبانی کنند.

متدولوژی اسکرام دارای اقبال بیشتری است و پروژه های بیشتری از آن استفاده کرده اند. این باعث می شود که تجربیات انتقال یافته و در آینده حوزه وسیع تری را پوشش دهد. اما کریستال کلیر از منظر سطح بحرانیت و سبزی تیم دارای محدودیت هایی است که بسیار بر حوزه کاربرد آن تاثیرگذار خواهد بود.

معيار اول زبان مدلسازی) پشتیبانی از مدلسازی شی گرای سازگار، دقیق و بدون ابهام

متدولوژی اسکرام

متدولوژی اسکرام درباره استفاده از مدلسازی تا حد امکان جلوی این کار را می‌گیرد و از مدلسازی پرهیز می‌کند. حتی لقب model-phobic به این متدولوژی اطلاق شده است. به همین دلیل دیدهای مختلف مدلسازی اعم از رفتاری، ساختاری و وظیفه مندی را پشتیبانی نمی‌نماید. همچنین مدلسازی کسب و کار و دامنه راه حل در این متدولوژی صورت نمی‌گیرد. نبود مدلسازی فرمال در این متدولوژی نیز عیان و واضح است. به همین دلیل برای پروژه‌های دقیق و سطح بحرانییت بالا اصلا مناسب نمی‌باشد.

این متدولوژی تنها در موارد خاص که پیچیدگی کار بالاست و به صورت حداقلی اجازه مدلسازی را به تیم توسعه می‌دهد. در نتیجه نبود مدلسازی در این متدولوژی باعث می‌شود که نمره پایینی در این معیار بگیرد. اما نباید از یاد برد که نبود مدلسازی به دلیل سرعت بالاتر توسعه و پیشرفت کار می‌باشد و این کار به صورت تعمدی از سوی واضعان این متدولوژی صورت گرفته است.

متدولوژی کریستال کلیر

متدولوژی کریستال کلیر از مدلسازی سیستم پرهیز می‌کند. تنها به مشخص سازی معماری سطح بالا و مدلسازی فضای مسئله می‌پردازد. این مورد باعث میشود که سطح بحرانییت پروژه دارای محدودیت گردد. اما جلوی تیم توسعه را نمی‌گیرد. همچنین در سیستم‌های سطح E و L نیز کاربردی ندارد.

در مجموع می‌توان گفت این متدولوژی نه تجویزی درباره مدلسازی می‌کند و نه جلوی تیم توسعه را می‌گیرد.

مقایسه متدولوژی اسکرام با کریستال کلیر

هر دو متدولوژی در این معیار دارای ضعف زیادی هستند. متدولوژی اسکرام تا حد زیادی جلوی تیم را از مدلسازی می‌گیرد و اجازه آن را نمی‌دهد. اما کریستال دستور معماری سیستم و مدلسازی فضای مسئله را صادر کرده است. همچنین اعضای کریستال با اضافه سازی مدلسازی قابلیت پشتیبانی از سیستم‌های با سطح بحرانییت بالاتر را فراهم می‌کنند که از ویژگی‌های خوب آن محسوب می‌شود.

در این معیار می‌توان متدولوژی کریستال را تا حدی برتر از اسکرام دانست اما با توجه به این نکته که خود این متدولوژی در مقابل متدولوژی‌های integrated و حتی متدولوژی‌های نطفه‌ای نسل اول و دوم دارای معایب زیادی می‌باشد.

معيار دوم زبان مدلسازی) تهیه استراتژی ها و تکنیک هایی برای مقابله با ناسازگاری مدل و مدیریت پیچیدگی مدل

متدولوژی اسکرام

همانگونه که در بخش قبل بیان شد، این متدولوژی مدلسازی را پشتیبانی نمی کند و تنها در موارد ضروری کاربرد می تواند مدلسازی را انجام دهد. اگر نیاز به مدلسازی بود، دست تیم باز است و می تواند از مدلسازی های مختلف استفاده کند. گزینه اول و مهم نیز UML است که دارای سطوح مختلف و دارای مکانیزم هایی برای جلوگیری از ناسازگاری می باشد. اما خود متدولوژی در این معیار نمره خوبی کسب نمی کند.

متدولوژی کریستال کلیر

متدولوژی کریستال کلیر به صورت شفاف و صریح مطلبی در این رابطه ذکر نکرده است. تیم توسعه نرم افزار خود موظف هستند برای مقابله با این مشکل یعنی ناسازگاری مدل ها و مدیریت پیچیدگی ایشان به دنبال راهکارهایی بگردند و از آن استفاده کنند. برای مثال استفاده از UML یکی از این راهکارها خواهد بود.

مقایسه متدولوژی اسکرام با کریستال کلیر

متدولوژی اسکرام با منع مدلسازی به نحوی صورت مسئله را پاک کرده و به همین دلیل مدلی وجود نخواهد داشت تا ناسازگاری پدید آید. در نتیجه تیم وقتی برای مقابله با ناسازگاری و پرهیز از آن صرف نخواهد کرد.

از طرفی دیگر کریستال کلیر اجازه مدلسازی را می دهد و در مواردی مانند تولید معماری آن را الزامی می کند اما راهکار مشروح و مبسوطی درباره آن ارائه نمی دهد.

در نتیجه در حالت کلی می توان گفت هیچکدام راهکار دقیق و مشخصی در صورت وجود مدل سازی در سیستم ارائه نداده اند. متدولوژی های سنگین وزن و *integrated* دارای روش ها و فعالیت هایی برای این کار هستند اما متدولوژی های چابک هیچ گزینه ای پیش پای توسعه دهنده نمی گذارند و هر دو متدولوژی اسکرام و کریستال کلیر در این معیار نمره قابل قبولی اخذ نمی کنند.