

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه پیام نور

مرکز تهران غرب

پروژه کارشناسی ارشد مدیریت کسب و کار گرایش استراتژی

عنوان:

تحلیل مقایسه ای روش های چابک و سنتی در مدیریت پروژه

فناوری اطلاعات

نگارش: سحر صبور - شماره دانشجویی: 990183396

استاد راهنما: دکتر علیرضا آبرود

پاییز 1402

فهرست مطالب

عنوان صفحه

فصل اول: کلیات پژوهش

- 1.1 مقدمه.....5
- 1.2 شرح و بیان مسئله.....6
- 1.3 اهداف انجام تحقیق.....6

فصل دوم: مبانی نظری و پیشینه پژوهش

- 2.1 مقدمه.....7
- 2.2 مبانی نظری.....8
- 2.3 روش شناسی سنتی.....8
- 2.3.1 روش آبشاری.....8
- 2.3.2 Prince2.....8
- 2.3.3 روش مسیر بحرانی.....8
- 2.4 روش شناسی چابک.....9
- 2.4.1 اسکرام.....9
- 2.4.2 کانبان.....11
- 2.4.3 تفاوت اسکرام و کانبان.....12

2.4.4 برنامه ریزی شدید XP.....13

فصل سوم : تجزیه و تحلیل داده ها و ارائه یافته ها

3.1 مقدمه.....15

3.2 ارائه یافته ها.....16

فصل چهارم : نتیجه گیری و ارائه پیشنهادها

4.1 مقدمه.....22

4.2 بحث و نتیجه گیری درباره یافته های پژوهش.....23

4.3 منابع و مآخذ.....24

چکیده

مشخص است که در ابتدا، در طول توسعه نرم افزار، روش سنتی برای همه پروژه های فناوری اطلاعات مورد استفاده قرار گرفته است و پروژه ها اغلب به دلیل رشد سریع صنعت فناوری اطلاعات ناموفق بوده اند و سپس متدولوژی های چابک شروع به توسعه کردند. روش های سنتی نسبت به روش های چابک مزایایی دارند و متداول ترین روش سنتی، روش آبشاری می باشد. با توجه به اینکه این روش محدودیت هایی در رسیدگی به مشکلاتی مانند کد بدون ساختار، پایین آوردن روحیه تیم، دید ضعیف، عدم ارتباط بین ذینفعان و اولویت بندی مکرر نیازهای کاربر دارد، بد نیست نگاهی به متدولوژی های چابک بیاندازیم که بر روی کار با کاربران، آزمایش به طور مستمر، بازسازی و توسعه تدریجی تمرکز دارد. این مقاله به مقایسه روش شناسی در مدیریت پروژه فناوری اطلاعات بر اساس سایر تحقیقات علمی می پردازد. برخی از متدولوژی های ذکر شده عبارتند از Scrum، Kanban، Waterfall و غیره. به این نتیجه رسیدیم که متدولوژی اسکرام چابک بیشتر در شرکت های فناوری اطلاعات مورد استفاده قرار می گیرد که با ترکیبی از چندین متدولوژی اغلب به دلیل نیاز هریک از پروژه ها، به منظور حذف مشکلاتی که ظاهر می شوند، کاستی های هر روش را مرتفع می سازد.

کلمات کلیدی: چابک؛ سنتی؛ پروژه فناوری اطلاعات؛ مدیریت؛ اسکرام؛ روش آبشاری؛ کانبان

فصل اول

کلیات پژوهش

1.1 مقدمه

برای موفقیت پروژه‌ها در مهندسی نرم افزار، سازماندهی خوبی لازم است. پروژه‌ها پیچیده تر هستند، بنابراین لازم است برخی از فرآیندهای رسمی مدیریت پروژه را بدانید. مدیریت پروژه به منظور استفاده بهینه از منابع، داشتن تقسیم کار یکنواخت، تشکیل برنامه ای که به موقع اجرا می شود، با روش اجرا قابل پیش بینی و نتایج خوب و مورد انتظار بسیار مهم است.

پروژه‌ها در صنایع مختلف به عنوان مثال در ساخت و ساز، فناوری، مواد و موارد مشابه استفاده می شوند. مدیریت پروژه در صنعت فناوری اطلاعات که نیازهای کاربران جدید خیلی سریع تغییر می کند، مهم است. برای موفقیت و مدیریت صحیح پروژه‌ها، مهم‌ترین نکته این است که توجه داشته باشید که پروژه‌ها تا حد امکان کارآمد باشند و در بازه زمانی معینی با منابع موجود و با بودجه مشخص اجرا شوند. . روش شناسی در مدیریت پروژه فناوری اطلاعات می تواند سنتی و چابک باشد و انتخاب برخی از آنها به ویژگی ها و ماهیت یک پروژه خاص بستگی دارد (ورسک، 2020). روش های مدیریت پروژه چابک چندین سال است که بیشتر در حرفه فناوری اطلاعات استفاده می شود. هنگام تصمیم گیری برای استفاده از روش مدیریت پروژه، نیازهای ذینفعان، ریسک های مرتبط با پروژه، اندازه پروژه، هزینه ها و البته پیچیدگی پروژه باید در نظر گرفته شود (ولادیمیرویچ اورلوف، و همکاران

همکاران، 2021). این مقاله به مقایسه رویکردهای سنتی و چابک بر اساس ادبیات علمی و سایر مطالعات می پردازد و همچنین به این نتیجه می رسد که کدام روش در عمل بیشترین استفاده را دارد.

1.2 شرح و بیان مسئله

از آنجایی که متدولوژی ها در مدیریت پروژه مهم هستند، تصمیم گیری در مورد متدولوژی مناسب بسیار مهم است و از این رو موضوع تحقیق، در نظر گرفتن متدولوژی ها در مدیریت پروژه فناوری اطلاعات و انتخاب مناسب ترین آنها می باشد. مسئله در این تحقیق انتخاب روشی است که به بهترین وجه با ماهیت یک پروژه خاص مطابقت داشته باشد. دلایل زیادی وجود دارد که باید در هنگام انتخاب به خاطر داشت، مانند مدت زمان پروژه، نیازهای مشتری، بودجه و غیره. مدیران باید متدولوژی ها و مزایای هر روش را بدانند. انتخاب متدولوژی، موفقیت پروژه را تضمین می کند. بررسی سیستماتیک می تواند به مدیران پروژه در این انتخاب کمک کند.

1.3 اهداف انجام تحقیق

هدف از این تحقیق تعیین این است که کدام روش مدیریت پروژه در پروژه های فناوری اطلاعات، در عمل بیشترین کاربرد را دارد، دلایل آن چیست و از کدام ترکیبی از روش شناسی های موجود استفاده می شود. می توان آن را با سؤالات تحقیق زیر ارائه کرد:

- سوال اول: دلایل انتخاب روش های چابک و سنتی کدام است؟
- سوال دوم: کدام روش بیشتر مورد استفاده قرار می گیرد؟
- سوال سوم: اگر ترکیبی از متدولوژی های متعدد استفاده شود، آن ترکیب ها کدامند و دلایل آن چیست؟
- سوال چهارم: اسکرام، کانبان و XP چه تفاوت هایی با هم دارند؟

فصل دوم

مبانی نظری و پیشینه پژوهش

2.1 مقدمه

"مدیریت پروژه استفاده از ابزارها، مهارت‌ها، تکنیک‌ها و دانش در فعالیتهای پروژه برای برآوردن اهداف پروژه یا نیازهای مشتری و محدوده پیش‌فرض آنها است" (ورسک، 2020). چندین متدولوژی مدیریت پروژه وجود دارد که برخی از آنها در این بخش مقاله ارائه می‌شوند.

2.2 مبانی نظری

2.3 روش شناسی سنتی

روش های سنتی در سازمان هایی اعمال می شوند که در آن پروژه های کاملاً تعریف شده اجرا می شوند و برنامه هایی از قبل آماده می شوند که هدف آنها برآورده کردن زمان، بودجه و اهداف پروژه است (ورسک، 2020؛ شنهار و دویر، 2007). مدیریت خود مبتنی بر دستورات، کنترل و ارتباطات رسمی است. مدل توسعه مبتنی بر مدل چرخه زندگی است. این موضوع یک برنامه ریزی کامل است و بررسی نتایج در پایان می آید (ورسک، 2020؛ انگلهارت، 2019). در روش سنتی، الزامات کاربر در ابتدای پروژه به وضوح تعریف می شود و مشتریان در همان ابتدای پروژه، نیازهای خود را بیان می کنند. تیم ها بزرگ هستند و پروژه ها زمان زیادی می برند (ورسک، 2020).

2.3.1 روش آبشاری

مدل آبشار توسط رویس در سال 1970 معرفی شد (کوکو و همکاران، 2011). رایج ترین روش روش های سنتی است. مراحل به شرح زیر است: بررسی نیازمندی، طراحی، پیاده سازی / توسعه، آزمایش و نگهداری (مک کورمیک، 2012). تیم فناوری اطلاعات پروژه زمان زیادی را بر روی فاز بررسی نیازمندی ها، یعنی برنامه ریزی و طراحی، صرف می کند و زمانی که مرحله پیاده سازی / توسعه شروع می شود، هیچ نیاز اضافی، تردید یا قسمت های تعریف نشده در پروژه یا نتیجه نهایی وجود ندارد. به همین دلیل، پروژه های بلندمدت برای استفاده از روش آبشاری مناسب هستند (ورسک، 2020؛ مک کورمیک، 2012؛ بهاوسار، و همکاران، 2020؛ آندری، و همکاران، 2019).

Prince2 2.3.2

Prince2 یک روش سنتی به نام روش «روند محور» است که به سؤالات «چه» و «چرا» و در مقادیر نادری به سؤال «چگونه» پاسخ می دهد. Prince2 که توسط دولت انگلستان ساخته شده است و اخیراً حقوق این روش در دست یک شرکت دولتی و خصوصی است (ورسک، 2020).

2.3.3 روش مسیر بحرانی (CPM)

"روش مسیر بحرانی (CPM) مبتنی بر یک مدل منطقی و ریاضی از پروژه است که اساس آن در زمان بهینه مورد نیاز برای یک فرآیند فردی در پروژه و بهره برداری همزمان از اقتصادی ترین منابع موجود است." (وودهد، 1991). بزرگترین

مزیت آن در تعیین دنباله ای است که ذخیره خط زمانی ندارد (مسیر بحرانی) و در تشخیص توالی فعالیت هایی که آنها را دارند (ورسک، 2020).

2.4 روش شناسی چابک

روش های مدیریت پروژه نرم افزار چابک، راهنمای برنامه ریزی و کنترل هستند (پارکر و دل مونت، 2014). برخی از ویژگی های پروژه های فناوری اطلاعات، نیاز به ایجاد متدولوژی های جدید برای کار کارآمد در برنامه نویسی و توسعه نرم افزار را مشروط کرده است. رویکردهای سنتی همیشه مناسب نبوده اند، اغلب به این دلیل که این پروژه ها معمولاً پروژه هایی هستند که بدون مشخصات ثابت و غیر قابل تغییر شروع می شوند، زیرا معمولاً تغییرات مکرر مورد نیاز است که مطابقاً به رویکرد متفاوت و انعطاف پذیرتری نیاز دارد. مشتری اغلب از آنچه دقیقاً در نتیجه کار انتظار دارد مطمئن نیست و از طریق درخواست تغییر به سمت هدف خود می رود. مشتری گاهی اوقات می خواهد که فرآیند به مرحله قبل برگردد و تغییرات خاصی ایجاد شود. تیمی که روی توسعه نرم افزار کار می کند باید آماده تغییرات مداوم و همکاری نزدیک با مشتری باشد (اسلام، 2013). همکاری تیمی، ارتباط خوب بین اعضای تیم و همچنین ارتباط خوب با مشتری به منظور هماهنگ کردن خواسته های مشتریان و امکان دستیابی به نتایج مهم است (احمد و همکاران، 2016). روش های توسعه نرم افزار چابک در اواخر قرن بیستم ظهور کردند، جایی که کار گروهی کلیدی است (احمد، و همکاران، 2016؛ میلو شویچ، 2018؛ اسلام، 2013). اصطلاح "چابک" به معنای توانایی یک روش برای پاسخگویی به تغییرات مکرر نیازها است. از آنجایی که مشتری نسبت به نمونه اولیه بینشی دارد، می تواند الزامات را تعریف کند و به تیم توضیح دهد که واقعاً چه نیازهایی دارد. با روش چابک، تاکید بر تغییراتی است که می تواند ثابت باشد و در نتیجه موفقیت پروژه را افزایش دهد. بهتر است در صورت نیاز به صورت دوره ای تغییرات ایجاد شود زیرا ارزان تر از زمانی است که پروژه از قبل تکمیل شده است (احمد، و همکاران، 2016؛ میلو شویچ، 2018؛ اسلام، 2013؛ ورسک، 2020). در روش شناسی چابک، پروژه می تواند بر اساس یک ایده حرکت کند و در نهایت به چیزی متفاوت از ایده اصلی منجر شود (ورسک، 2020؛ اسلام، 2013). روش های چابک/اسکرام و برنامه ریزی شدید (Agile/Scrum) و (Extreme Programming (XP) به طور گسترده در شرکت ها برای انجام پروژه های توسعه نرم افزار استفاده می شوند (باراش، 2013؛ گارسیا، و همکاران، 2020؛ ونکاتاچلام، و همکاران، 2017).

2.4.1 اسکرام

اسکرام یک فرآیند تکرار شونده-افزاینده و رایج ترین روش رویکرد چابک است (Mahnič، 2015؛ شوابر، 2004؛ Cocco، و همکاران، 2011)، (Stellman & Greene، 2017؛ Lei، et al.، 2017؛ Granulo، و تانوویچ، 2019؛ اینگاسون، و همکاران، 2013). این فرآیند شامل مجموعه ای از توصیه های مدیریتی است، اما فعالیت های خود فرآیند

توسعه را تعریف نمی کند. اغلب در ترکیب با سایر فرآیندهای توسعه نرم افزار استفاده می شود. اسکرام خروجی سیستم آینده را پس از هر بار تکرار اندازه گیری می کند.

نقش های موجود در روش اسکرام، مالک محصول، تیم توسعه، اسکرام مستر، مدیر و مشتری است (پانتلیچ، و همکاران، 2020؛ شوابر، و بیدل 2001؛ لانتی، 2013؛ البرقی و قریشی، 2018؛ فوستیک، 2017). صاحب محصول (Terlecka، 2012؛ Venkatachalam، و همکاران، 2017)، ورودی ها را از مشتریان، کاربران نهایی و اعضای تیم توسعه جمع آوری می کند و سپس آنها را به نیازمندی ها تبدیل می کند و آنها را از نظر اولویت های توسعه ارزیابی می کند. او مسئول توسعه و تحویل محصول با توجه به نیاز مشتری است. تیم توسعه معمولاً از پنج تا ده عضو تشکیل می شود که برخی از آنها تحلیلگر، توسعه دهنده، طراح و آزمایش کننده هستند. این تیم دارای استقلال در تصمیم گیری و همچنین آزادی ارائه ایده هایی برای بهبود محصول به صاحب محصول است. اسکرام مستر باید میانجیگری بین صاحبان محصول و اعضای تیم توسعه باشد و مسئولیت توسعه موفقیت آمیز محصول نهایی و اجرای موفقیت آمیز روش اسکرام در پروژه، ارائه کمک و پشتیبانی مستمر به اعضای تیم توسعه را بر عهده دارد. مدیر، مسئول تصمیم گیری نهایی است و در فرآیند تعیین اهداف و تعیین الزامات شرکت می کند. مشتری در فرآیند ایجاد درخواست ها و تعریف عملکردی که سیستم آینده باید داشته باشد شرکت می کند و فرآیند بررسی نتایج و عملکرد به دست آمده را به عهده دارد و بازخورد خود را به تیم توسعه از طریق مالک محصول می دهد.

کار توسعه نرم افزار در چرخه های کوتاه تری به نام اسپرینت انجام می شود. پس از آن باید به طور مستمر با مشتری مشورت گردد و پس از یک چرخه مشخص، تحلیل و بررسی و هرگونه تغییر دلخواه و لازم انجام می شود. جلسات قبل و بعد از هر اسپرینت برای تعیین اینکه آیا همه چیز طبق درخواست مشتری انجام شده است و برای تعیین اینکه آیا چیزی نیاز به تغییر دارد یا خیر، اجباری است. هر اسپرینت شامل جلسات روزانه اسکرام، دوره مسایل اسپرینت و برنامه ریزی برای اسپرینت بعدی می باشد. اسپرینت می تواند تا یک ماه ادامه داشته باشد (Ferrão & Canedo، 2015؛ Brezočnik & Majer، 2016)، و نتیجه یک محصول اجرایی است (Pantelić، و همکاران، 2020؛ Schwaber & Sutherland، 2020؛ Fustik، 2017). هنگام برنامه ریزی یک اسپرینت، مشخص می شود که چه کاری باید در اسپرینت انجام شود (Bhavsar, et al., 2020)، یعنی نتیجه محصول چه خواهد بود و چگونه آن نتیجه حاصل می شود. جلسات روزانه باید برگزار شود و می تواند حدود پانزده دقیقه طول بکشد. هر یک از اعضای تیم موظف است تا جلسه روزانه فردا گزارشی را در مورد مشکلاتی که متوجه شده است، در مورد کار انجام شده در روز قبل و نتایجی که قصد دارد به دست آورد، ارائه دهد (پانتلیچ، و همکاران، 2020؛ آبراهامسون، و همکاران، 2002). عبارت بکلاگ بخشی ضروری از اسکرام است. در روش اسکرام بکلاگ محصول و بکلاگ اسپرینت وجود دارد. بکلاگ محصول فهرستی از تمام ویژگی های نتیجه نهایی است که تیم باید برای رضایت بخش بودن نتیجه ایجاد کند. بکلاگ اسپرینت مجموعه ای از تمام

ویژگی‌های ارائه شده توسط مالک محصول به همراه تیمی است که تصمیم به توسعه در اسپرینت بعدی دارد. ویژگی‌ها از بکالگ محصول گرفته شده اند که از آن کارکردها با اولویت انجام می شود (ورسک، 2020؛ استلمن و گرین، 2017). برخی از افراد بر این باورند که اسکرام بسیار سفت و سخت است، مقیاس پذیر نیست و برای کارهای تعمیر و نگهداری نامناسب است (Sjoberg و همکاران، 2012).

2.4.2 کانبان

کانبان در سال 2004 وارد توسعه نرم افزار شد. کانبان دومین روش معروف چابک است (Moonden, 2011). کانبان روشی مبتنی بر اصطلاح «در زمان» است (احمد، و همکاران، 2013). ایده این روش از سه مهندس تویوتا (احمد، و همکاران، 2018؛ آندری، و همکاران، 2019)، کیچیرو تویوتا، ایچی تویوتا و تایچی اوهنو در دهه 1940 گرفته شد (Womack، و همکاران، 1990؛ Stellman و گرین، 2017؛ موندن، 2011). "کانبان دارای پنج اصل اساسی است: تجسم گردش کار، محدود کردن گردش کار، اندازه گیری و مدیریت گردش کار، مشخص کردن خط مشی های فرآیند و استفاده از مدل ها برای شناسایی پیشرفت ها و فرصت ها" (احمد و همکاران، 2016). Johnsen, Sjoberg (2012) Solberg استفاده از اسکرام و کانبان را در یک شرکت متوسط در یک دوره دو ساله مقایسه کردند. یافته های آنها نشان می دهد که با کانبان تعداد خطاهای وزنی به ترتیب 10% و 11% کاهش یافته و بهره وری 21% برای عملکردهای اجرا شده بهبود یافته است. کانبان بر ارتباط، همکاری و یکپارچگی بین توسعه دهندگان نرم افزار، آزمایش کنندگان و تیم های پشتیبانی تمرکز دارد که در نتیجه توسعه سریع نرم افزار و تحویل مداوم به مشتری ایجاد می شود. کانبان در توسعه نرم افزار روشن بودن مسیر را به ارمغان می آورد و کارایی و توان عملیاتی کار را بهبود می بخشد. روبه صنعتی نشان داده است که کانبان می تواند هزینه های نگهداری و توسعه را کاهش دهد (احمد، و همکاران، 2016). ایده اصلی این است که هر سازمانی از طریق یادگیری مستمر از تجربیات خود، اصول اولیه کانبان را به روش خود پیاده می کند. به همین دلیل، دستورالعمل های مستدلی برای اجرای موثر آنها وجود ندارد. فلسفه کانبان بر کاهش هزینه های کلی، بهبود کیفیت و کیفیت کلی محصولات تحویل شده به مشتریان، کوتاه کردن زمان تحویل محصول و افزایش رضایت مشتری متمرکز است. یکی از مهم ترین ایده های روش کانبان، حذف زیاده روی است. این امر با استفاده از کارت های کانبان و تابلوهای کانبان برای تجسم نحوه حرکت منابع در چرخه تولید به دست می آید. این کار به همه شرکت کنندگان اجازه می دهد تا به طور کامل در فرآیند مشارکت داشته باشند و به مدیران کمک می کند تا مازاد یا کسری تولید را مدیریت کنند. کانبان به سازمان ها اجازه می دهد تا با گردش کار موجود خود شروع کنند و به آرامی تغییرات را در طول زمان برای رسیدن به سطح مورد نظر و دستیابی به نتایج مورد نظر معرفی کنند. این را می توان با محدود کردن کار در حال انجام (Work In Progress – WIP) نیز به دست آورد. فعالیت هایی که تشویق می شوند عبارتند از تجسم، کار در حال پیشرفت، مدیریت گردش کار، قوانین به وضوح تعریف شده و بازخورد. سیستم های کانبان از مکانیسم هایی مانند

برد کانبان برای تجسم کار و فرآیندی که کار طی می کند استفاده می کند. چندین شاخص برای تشخیص تجسم مؤثر تعریف شده است. مانند نقطه تعهد و محل تحویل. نقطه تعهد زمانی به وجود می آید که تیم کاری با انجام یک وظیفه کاری خاص موافقت کند (پانتلیج، و همکاران، 2020). نقطه تحویل زمانی است که تیم کار را به مشتری تحویل می دهد. سپس، تجسم مؤثر در سیاست‌هایی که تعیین می‌کنند کدام کار باید در یک مرحله خاص باشد، و همچنین در محدودیت‌های کار در حال پیشرفت، منعکس می‌شود. زمانی که محدودیت‌های مربوط به میزان کار در حال انجام سیستم تعیین می‌شود و از آن محدودیت‌ها استفاده می‌کنید تا بدانید چه زمانی باید فرآیند جدید را شروع کنید، میتوان گردش کار و زمان را کاهش داد، کیفیت را بهبود بخشید و اغلب کیفیت بیشتری ارائه داد. گردش کار باید تحویل ارزش را به حداکثر برساند، زمان تلف شده را به حداقل برساند و تا حد امکان قابل پیش بینی باشد. تیم‌ها از کنترل تجربی از طریق شفافیت، بازرسی و انطباق برای ایجاد تعادل بین اهداف بالقوه متضاد و عدم تحقق آنها استفاده می‌کنند. قوانین واضح به توضیح کل فرآیند و مراحل مختلف در طول فرآیند کمک می‌کند. قوانین باید ساده، به خوبی تعریف شده، قابل مشاهده، همیشه قابل اجرا و به گونه‌ای باشند که به راحتی قابل تغییر باشند. در نهایت، بازخورد یک عنصر اساسی در هر سیستمی است که به دنبال تغییر تکاملی است (Cocco, et al., 2011; Senapathi ;Kniberg & Skarin, 2010). (& Drury-Grogan, 2021; Santos, et al., 2018).

2.4.3 تفاوت اسکرام و کانبان

هر دو اسکرام و کانبان به روش‌های چابک و ناب تعلق دارند. جایی که تمرکز بر پاسخگویی سریع به درخواست‌های مشتری است. هر دو بسیار سازگار هستند و بر اساس تیم‌های بسیار مشارکت‌کننده و خودمدیریتی هستند. کنیبرگ و اسکارین (2010) گفتند که اسکرام بیشتر از کانبان تجویز می‌شود. اسکرام استفاده از تکرارهای زمانی را تجویز می‌کند، کانبان اینطور نیست. بین صفحات اسکرام و کانبان تفاوت وجود دارد. اسکرام دو حالت دارد: حالت زمان بندی و حالت عملیات. بورد اسکرام توسط تیم‌ها برای برنامه‌ریزی کار قبل از شروع واقعی استفاده می‌شود. در حالت برنامه‌ریزی، تیم‌ها کار خود را در اسپرینت به اشتراک می‌گذارند و نقاط داستانی را به داستان‌های کاربر اختصاص می‌دهند تا برنامه‌ریزی کنند که کدام داستان به کدام سرعت می‌رود. در مقابل، کانبان دستور برنامه‌ای ندارد. کانبان از همان رابط کاربری ستونی اسکرام، برای ردیابی وضعیت آیتم بدون سازماندهی کار به صورت اسپرینت استفاده می‌کند (احمد، و همکاران، 2016). ساهوتا توضیح می‌دهد که کانبان می‌تواند با تعداد زیادی وقفه در پروژه، کارکنان پشتیبانی با نقش‌های تخصصی و مجموعه‌ای از مهارت‌ها مقابله کند. کانبان برای تیم‌های بزرگتر خوب عمل می‌کند زیرا هزینه‌های ارتباط و برنامه‌ریزی کمتر است. اسکرام در پروژه‌هایی که نیاز به همکاری و نوآوری عمیق دارند بهتر عمل می‌کند و با تیم‌های کوچک متقابل کار می‌کند و کارشناسان عمومی را تشویق می‌کند (لی و همکاران، 2017؛ اینگاسون و همکاران، 2013؛ احمد و همکاران، 2014؛ نیکیتینا. Kajko-Mattsson & Miller, 2011; Al-Baik & Sjøberg, 2015).

همکاران، 2012؛ احمد، و همکاران، 2016؛ Fagarasan، 2021). بسیاری از شرکت‌هایی که قبلاً از اسکرام استفاده می‌کردند، کانبان را به دلیل ویژگی‌های آن از جمله شفافیت و کار محدود در حال انجام، اتخاذ کردند (احمد و همکاران، 2014؛ نیکیتینا و کاجکو-ماتسون، 2011؛ آل بایک، و میلر، 2015؛ اسجوبرگ و همکاران، 2012؛ احمد و همکاران، 2016؛ شفیق و عنایت، 2017).

2.4.4 برنامه ریزی شدید (XP)

با روش برنامه ریزی شدید، تمرکز بر تولید نرم افزار با کیفیت بهتر و کار سازنده تر تیم توسعه است. این روش در اواسط دهه 1980 توسط کنت بک و وارد چانینگ توسعه یافت. بک با به کارگیری مهمترین موارد، یعنی ارتباط، سادگی، بازخورد و شجاعت، اصول و روشهای جدیدی را برای بکارگیری در پروژه ها اضافه کرد (پاپ، 2008). این روش منحصر به فرد است زیرا به طور خاص برای توسعه نرم افزار استفاده می شود (Beck & Andres, 2004). اصل این روش این است که مشتری عضوی از تیم است و اهداف و اولویت‌ها را در واحدهای کاربری با همکاری مستمر اعضای تیم تعریف می‌کند تا تیم توسعه دهنده نیازهای کاربر را بهتر درک کند. توسعه از طریق چرخه‌های کوتاهی انجام می‌شود که شامل طرح دوره فعلی (current iteration) می‌شود. موفقیت فرآیند توسعه با مشاهده پیشرفت سنجیده می‌شود. تمام جزئیات واحدهای کاربر در قالب آزمایش مستند شده است تا نظارت بر پیشرفت و الزامات اجرا شده را تسهیل کند. کد تولیدی باید توسط دو برنامه نویس در یک واحد توسعه نوشته شود. یکی از اعضا کد را می‌نویسد در حالی که دیگران آن را بررسی می‌کنند و بنابراین روند پیاده سازی را دنبال می‌کنند. مطلوب است که نقش‌ها به طور مکرر تغییر کنند تا یک رابطه کیفی بین اعضا حفظ شود. کد تولیدی برای برآورده کردن کد تست نوشته شده است. نرم افزار توسعه یافته دارای جمعی اعضای تیم است. ماهیت برنامه ریزی در تقسیم مسئولیت‌ها بین کاربر و تیم توسعه است. کاربر تصمیم می‌گیرد که نرم افزار کدام قابلیت‌ها را اجرا کند و هزینه آن چقدر است. سیستم باید تا حد امکان ساده و با بازسازی مکرر طراحی شود. بزرگترین مزیت XP این است که به شرکت‌های توسعه نرم افزار اجازه می‌دهد تا با حذف فعالیت‌های غیرمولد در هزینه‌ها، ناامیدی‌ها و زمان صرفه جویی کنند. در درجه اول به دنبال کاهش خطرات مرتبط با شکست پروژه است و این به توسعه دهندگان اجازه می‌دهد تا روی کننویسی تمرکز کنند. ساخت فرآیند راه حل نرم افزار XP از طریق چندین مرحله توسعه محقق می‌شود: تحقیق، برنامه ریزی، پیاده سازی دوره، تولید، نگهداری و تکمیل (Marić، 2015؛ Tadić، 2005؛ Malik، و همکاران، 2019؛ Pap، 2008؛ Subih، و همکاران، 2019).

به گفته نویسنده (Malik, et al., 2019)، انعطاف پذیری روش‌های چابک مهم‌ترین ویژگی آن‌ها است، چه اسکرام، چه برنامه ریزی شدید و یا دیگر روش‌ها. یکی از مشکلات روش‌های چابک این است که در پروژه‌های پیچیده، بخش‌هایی از روش‌های چابک مانند جلسات یا تماس‌های تلفنی که ضروری هستند ممکن است در تیم‌های بین‌المللی که

دارای مناطق زمانی متفاوت هستند مشکل ساز باشند. برنامه نویسی شدید بیشتر تمرکز خود را بر توسعه نرم افزار یا بهترین شیوه ها برای توسعه آن می گذارد، در حالی که بهترین شیوه ها را در نحوه اجرای کل پروژه در یک بودجه معین و در مهلت های تعیین شده کنار می گذارد یا اولویت کمتری در این روش دارد (Javanmard & Alian, 2015). برای تسهیل این کاربرد در پروژه های بزرگتر با پیچیدگی بیشتر، توصیه می شود از نرم افزار مدیریت پروژه ای استفاده شود که، به سه دسته نرم افزار مدیریت پروژه، نرم افزار مدیریت فرآیند و نرم افزار ردیابی زمان تقسیم می شود. (ورسک، 2020)

فصل سوم

تجزیه و تحلیل داده ها و ارائه یافته ها

3.1 مقدمه

بر اساس تحقیقات در یک شرکت فناوری اطلاعات، مشخص شد که پاسخ دهندگان زمانی روش های سنتی را انتخاب می کنند که دقیقاً بدانند در طول پروژه چه کاری باید انجام شود، در حالی که روش های چابک اغلب زمانی انتخاب می شوند که فقط نیازمندی ها و اهداف «حدودی» شناخته شده باشند. روش های سنتی حتی زمانی انتخاب می شوند که پروژه را نمی توان به بخش های کوچک تر تقسیم کرد، در حالی که روش های چابک زمانی انتخاب می شوند که انعطاف پذیری لازم باشد. تاکید بر این نکته حائز اهمیت است که روش های سنتی زمانی انتخاب می شوند که کارفرمای پروژه افراد فنی لازم برای بررسی قسمت های خاص را نداشته باشد و فقط به محصول نهایی علاقه مند باشد، در حالی که روش های چابک زمانی انتخاب می شوند که تغییراتی در طول پروژه انتظار می روند و این تغییرات متداول باشند.

(ورسک، 2020). تحقیقات لیورمور (لیورمور، 2008)، نشان می‌دهد که بین اندازه تیم و موفقیت اجرای روش‌ها در رابطه با آنها همبستگی معناداری وجود ندارد. وی بیان می‌کند که این نتیجه، با توجه به اینکه تعداد زیادی از تحقیقات در مورد موضوع مذکور به این نتیجه رسیده‌اند که روش‌های چابک نسبت به تیم‌های بزرگتر موفقیت کمتری دارند، غیرمنتظره است. این نتایج از این واقعیت حمایت می‌کند که روش‌ها، به ویژه چابک، با تیم‌ها و موقعیت‌های مختلف سازگار هستند. انعطاف‌پذیری آنها نه تنها از نظر پاسخ سریع به درخواست‌های اضافی در اینجا قابل استفاده است، بلکه آنها در انطباق با یک تیم یا پروژه نیز انعطاف‌پذیر هستند - اندازه تیم، نقش تیم، طول سرعت (ورسک، 2020). روش‌های چابک بهتر از روش‌های سنتی هستند زیرا می‌توانند در هزینه و زمان صرفه‌جویی کنند و محصولات باکیفیت را به موقع تحویل دهند (Venkatachalam، و همکاران، 2017). به گفته نویسندگان (Merzouk و همکاران، 2018) و مقایسه آنها از روش‌های چابک، هنگام انتخاب روش چابک صحیح برای یک پروژه، باید اندازه تیم و پروژه را مشاهده کرد (Vresk، 2020). در متدولوژی‌های چابک، نیازهای کاربر به وضوح تعریف نمی‌شود، فقط نتیجه نهایی تعریف می‌شود. تیم‌ها نسبت به روش سنتی کوچکتر هستند. مشتری از ابتدا تا انتها در کل پروژه درگیر است. کلید این است که تیمی داشته باشید که با هم کار کنند. ارتباط غیررسمی است. روش‌شناسی چابک یک مدل تکرارشونده است (ورسک، 2020). در سال 2010، فورستر (Womack، و همکاران، 1990)، نتایج نظرسنجی جهانی توسعه دهندگان خود را گزارش کرد، که نشان داد 35٪ از پاسخ دهندگان از یک فرآیند چابک، 21٪ از یک فرآیند تکرارشونده و 13٪ از یک فرآیند توسعه آشناری استفاده کردند، در حالی که 31٪ از آنها از یک روش فرآیند رسمی مدیریت پروژه استفاده نکردند (Mahnič، 2014).

3.2.3 ارائه یافته‌ها

طبق نظر Rising و Janoff (2000) و Schwaber (2004)، رایج‌ترین روش چابک اسکرام است. گزارش شده است که اسکرام خالص توسط 54٪، Scrum/XP هیبریدی توسط 11٪ و اسکرام‌بان توسط 7٪ از پاسخ دهندگان استفاده می‌شود. با این حال، یک نظرسنجی همچنین رشد سریع تعداد کاربران کانبان را نشان داد. در مقایسه با سال 2011، کانبان و انواع آن (اندرسون، 2010؛ لاداس، 2008)، در سال 2012 تقریباً دو برابر شده است که بیشتر به دلیل افزایش استفاده از اسکرام‌بان بود (Mahnič، 2014).

از آنجایی که هر روش دارای ویژگی‌ها و معایب خاص خود است، یک قالب مستقل از اسکرام یا کانبان یا واترفال نمی‌تواند راه حل‌های کاملی برای همه چالش‌ها ارائه دهد. فریم‌ورک‌های چابک برای پروژه‌های نرم افزاری برد بزرگ یا

متوسط بسیار مناسب هستند. جایی که نیاز پروژه نیز این است که امکان سنجی آن اغلب در طول زمان تغییر کند، در حالی که آبشار برای پروژه های کوتاه مدت مناسب است و نیاز واضح در فاز اولیه پروژه را می توان پیش بینی کرد و الزامات پروژه ممکن است در طول چرخه عمر آن تا تحویل نهایی پروژه تغییر نکند. ادغام اسکرام و کانبان با وافرال قدرت بزرگی را برای شیوه های مدیریت مهندسی نرم افزار (SEM) به شکل اسکرامبان فال (Scrumbanfall) فراهم می کند. مصنوعات اسکرامبان فال اساس چارچوب در فرم ارزش مشتری، اسناد، مدیریت گردش کار، شفافیت، افزایش و فرصت برای تیم اسکرامبان فال و سایر ذینفعان است (Bhavsar, et al., 2020). کوکو و همکاران (2011)، یک مدل دینامیک سیستم برای مقایسه ابزارهای تکنیک های شبیه سازی برای رویکرد تجویزی با استفاده از اسکرام و کانبان در روش آبشاری در سال 2011 توسعه دادند و ارزیابی کردند که هر یک از آنها نقاط قوت و ضعف خود را دارند و کار تحقیقاتی بیشتری را در مورد ترکیبی از همه آنها برای حل مشکل توسعه نرم افزار چابک پیشنهاد کردند (Bhavsar, et al., 2020). در سال 2019، موهان، دویزری و کومار قوانین اسکرام را با انعطاف پذیری کانبان در قالب اسکرامبان ترکیب کردند و در توسعه نرم افزار بخش دولتی آن را پیاده سازی کردند که استرس توسعه سربار را کاهش و کارایی نرم افزار را افزایش داد. تبدیل اسکرام، کانبان و روش آبشاری به اسکرامبان فال در مقایسه با چارچوب مستقل و تکی هرکدام و قابلیت پاسخگویی به چالش های توسعه نرم افزار و شیوه های مدیریتی مانند دخالت مستقیم ذینفعان خارجی در نیازمندی های پروژه و مستندات تجزیه و تحلیل، قدرت زیادی دارد. برنامه ریزی پروژه، برآورد درست تر و ردیابی دقیق تر از مرایای این روش می باشد. برآورد نادرست خطرات زیادی را برای پروژه به وجود می آورد و دید نامشخص در مورد محصول نرم افزاری درست می کند که منجر به هزینه غیرمنتظره در پروژه برای صاحبان پروژه نرم افزاری یا سازمان های توسعه نرم افزاری می شود و هزینه توسعه پروژه به مراتب بالاتر می رود. (Bhavsar, et al., 2020). در یک مقاله، نویسندگان رفتار پویای کانبان و اسکرام را در مقابل فرآیند توسعه نرم افزار سنتی مانند رویکرد آبشاری تحلیل کردند. آنها از یک مدل دینامیک سیستم، بر اساس روابط بین متغیرهای سیستم، برای ارزیابی مزایای نسبی رویکردهای مورد مطالعه استفاده کردند. گردش کار کانبان از طریق یک مکانیسم کنترل موثر باعث محدود کردن کار در حال انجام و به حداقل رساندن زمان انجام مدیریت شد. یکی از مزایای این روش این است که کار بهتر کنترل می شود، به طوری که اثرات خطاهای محدود می شود. برعکس، در مورد روش آبشاری، اغلب پروژه ها ممکن است به دلیل مشکل در تصحیح خطاها، از جمله خطا در الزامات، تکمیل نشوند (کوکو، و همکاران، 2011). در مطالعه دوم، پروژه دیوار کانبان می تواند بررسی و اجرای پروژه را تسهیل کند و ارتباط بین ذینفعان را کارآمدتر و موثرتر کند. استفاده ترکیبی از اسکرام و کانبان در این پروژه بی عیب بوده است و روش جدید از آغاز تا کنون موفق بوده است (اینگاسون و همکاران، 2013). یک مطالعه نشان داد که اندازه گروه در کانبان و XP نسبت به اسکرام سازگارتر است، در حالی که اندازه کار در حال انجام (WIP) اسکرام یک، دو یا سه ماه در اسپرینت است، اما ارزیابی کار در حال انجام کانبان و XP نسبتاً کم است (می تواند به اندازه یک ساعت مجموعه باشد) (صالح و

همکاران، 2019؛ کوهن، 2007). در این راستا، کانبان پیش‌شرط‌ها را هر روز سازمان‌دهی می‌کند و XP به طور مداوم نیازهای سازمان‌دهی شده را اجازه می‌دهد که انجام شوند. در حالی که پیش‌شرط‌ها باید بر اساس طول اجرا در اسکرام سازماندهی شوند (کانوال، و همکاران، 2010؛ بک و فاولر، 2000). علاوه بر این، در کامپوننت کانبان حجم در مقایسه با اسکرام و XP کم است، اما یک صنعت زمانی که از اسکرام به کانبان منتقل می‌شود، در بازه‌های زمانی بین پیشنهاد عنصر دوم و ایجاد درخواست‌ها و ارسال آنها به وب‌سایت کاربر، نصف می‌شود (شالووی، 2011؛ رامن، 2014). کانبان و XP نیز بیشتر بر نرم افزارهای بهبود کیفیت متمرکز هستند (Verweij & Maassen, 2011; Sjøberg, et al., 2012; Raman, 2014).

تحقیقات نشان می‌دهد که حدود نیمی از کسب و کارها هنوز از مدل آبخاری استفاده می‌کنند، در حالی که نیمی دیگر از رویکردهای چابک و تکرارشونده استفاده می‌کنند (هولز، 2019). شرکت‌هایی که از روش‌های چابک استفاده می‌کنند، طبق داده‌های دهمین نظرسنجی سالانه "ورژن یک"، اغلب اسکرام و اسکرام + XP (70%)، اسکرام‌بان (7%) و کانبان (5%) (ساترلند، 2010) را انتخاب می‌کنند. از انتخاب روش‌های چابک، برنامه ریزی شدید (XP) را حذف کردیم، زیرا اصول آن اغلب در ترکیب با روش‌های دیگر (اسکرام و کانبان) استفاده می‌شود (Majer و Brezočnik, 2016). یک مطالعه نشان داد که اسکرام مطمئناً در شرکت‌های بالغی که تیم‌های باتجربه‌ای دارند که بیش از یک سال روی محصول یا پروژه کار می‌کنند، بهترین عملکرد را دارد. برای شرکت‌هایی با تولید مستمر که نیاز به واکنش سریع به تغییرات دارند و تیم‌های محصولی که در پشتیبانی و نگهداری محصول فعالیت می‌کنند، استفاده از کانبان را پیشنهاد می‌کنیم. اسکرام‌بان برای شرکت‌های جوان و کوچک بهترین است، زیرا دارای انعطاف‌پذیری کانبان و ویژگی‌های اساسی اسکرام است. اما مطمئناً روش‌های چابک یک جزء قوی از انعطاف‌پذیری را شامل می‌شوند. تیم‌ها می‌توانند، صرف نظر از روش انتخاب شده، آن را به گونه‌ای تطبیق دهند که در خدمت هدف آنها باشد - سازماندهی کار موثر و توسعه محصولات با کیفیت (Majer و Brezočnik, 2016).

به گفته آلایداروس و همکاران. (2018)، اسکرام بیشترین را دنبال کننده را دارد و 58 درصد از پاسخ دهندگان آن را در بین سایر روش‌های چابک انجام می‌دهند، در حالی که بیش از 39 درصد از پاسخ دهندگان روش کانبان را در سازمان خود تمرین می‌کنند. اسکرام محبوب‌ترین روش چابک است. فعالیت‌های روزانه مبتنی بر تجربیات گذشته است و نه تنها بر جنبه‌های نظری (Mircea, 2019). محبوب‌ترین روش مورد استفاده داوطلبان در تحقیق دوم اسکرام بود که 62/5 درصد از آنها حداقل یک بار از اسکرام استفاده کردند. 28.1 درصد از آنها از روش آبخاری و تنها 25 درصد از کانبان

استفاده می کردند (Andrei, et al., 2019). پاسخ دهندگانی که از اسکرام استفاده کردند گزارش دادند که در طول توسعه بسیار انگیزه داشتند و از این واقعیت خوششان می آمد که با بقیه اعضای تیم هماهنگی خوبی داشتند. با این حال، برخی از آنها از تعداد زیاد جلسات و اینکه آنها یک روتین را القا می کردند، لذت نمی بردند. کاربران کانبان از سادگی و انعطاف پذیری روش در حین گزارش مشکلات مربوط به اولویت بندی کارها و مدت زمانی که برای به روزرسانی کارتها صرف کردند، لذت بردند. کاربران روش آبشاری از توسعه برنامه محوری که این روش شناسی تشویق می کند و الزامات سختگیرانه ای که از ابتدا تحمیل شده بود لذت بردند، اما مشکلاتی در اندازه گیری پیشرفت داشتند (Andrei, et al., 2019). بر اساس آمار ارائه شده در گزارش سالانه وضعیت چابک (Engelhardt, 2019) که توسط Digital.ai تهیه شده است، 58 درصد از پروژه های نرم افزاری که با استفاده از متدولوژی های چابک پیاده سازی شده اند نیز اسکرام را به عنوان روش پیاده سازی انتخاب کرده اند. در مقابل، تنها 7 درصد از کل پروژه های نرم افزاری با استفاده از روش توسعه کانبان خالص پیاده سازی شدند، این آمار نباید بر تصمیم مدیران پروژه در هنگام انتخاب روش اجرای مناسب تأثیر بگذارد (Fagarasan, et al., 2021). مطالعات مختلف گزارش کرده اند که روش کانبان در حال حاضر در میان روش های چابک مدعی است زیرا دارای مزایای متعددی است که باعث می شود از نظر تجربه سازگاری بیشتر در مدیریت پروژه های مهندسی نرم افزار (SE) بهتر از اسکرام و سایر روش های چابک عمل کند (Chande, 2014؛ لی، و همکاران، 2017؛ Karunanithi, 2016). با این حال، روش کانبان کمبود قابل توجهی در وظیفه نظارت بر پیشرفت در طول فرآیند توسعه پروژه های نرم افزاری دارد. این مشکل به دلیل تاخیر در زمان بندی پروژه ها که منجر به تحویل دیر هنگام می شود، بر موفقیت پروژه های نرم افزاری تأثیر منفی می گذارد (اسکینر، و همکاران، 2015؛ آل بایک و میلر، 2015؛ کیرووسکا و کوچسکی، 2015).

همانطور که در جدول 1 مشاهده می شود، متدولوژی های چابک تنها زمانی استفاده می شوند که الزامات و هدفی کلی وجود داشته باشد، زمانی که تغییرات در طول پروژه ضروری باشد و در جایی که تغییرات در کل پروژه امکان پذیر باشد. این به صرفه جویی در هزینه و ارائه محصول با کیفیت به موقع کمک می کند. روش های سنتی زمانی استفاده می شوند که تمام نیازهای مشتری دقیقاً از ابتدا مشخص باشد. مورد استفاده مشتری پروژه دقیقاً می داند که می خواهد محصول نهایی او باشد. طبق ادبیات علمی، چابک ترین روش مورد استفاده اسکرام و روش سنتی آن روش آبشاری است.

جدول 1. مروری بر دلایل استفاده از روش شناسی چابک و سنتی و بررسی متداول ترین روش شناسی

روش شناسی در مدیریت پروژه فناوری اطلاعات	روش چابک	روش شناسی سنتی
انتخاب روش	آنها زمانی استفاده می شوند که فقط نیازمندی ها و اهداف اولیه شناخته شده باشند و انعطاف پذیری و تغییرات لازم در طول پروژه مورد انتظار باشد زیرا می توانند در هزینه، زمان و ارائه محصولات به موقع صرفه جویی کنند.	زمانی استفاده می شود که دقیقاً مشخص باشد که در طول پروژه چه کاری باید انجام شود و زمانی که پروژه را نمی توان به بخش های کوچکتر تقسیم کرد و زمانی که مشتری پروژه فقط به محصول نهایی علاقه مند است و ارتباطات در تیم غیررسمی است.
شایع ترین روش	اسکرام (Mircea، 2019)	آبشار (Mircea، 2019؛ Andrei، و همکاران، 2019).

با توجه به اینکه هر متدولوژی دارای کاستی هایی است، لازم است در برخی موارد که ماهیت پروژه ایجاب می کند چندین متدولوژی ترکیب شود. اسکرام و کانبان برای پروژه های نرم افزاری در مقیاس بزرگ یا متوسط مناسب هستند که نیازها در طول عمر پروژه تغییر می کنند روش آبشاری برای پروژه های کوتاه مدت مناسب است و الزامات آن از ابتدای پروژه مشخص است. اسکرام مطمئناً در شرکت های بالغی که تیم های باتجربه ای دارند که بیش از یک سال روی محصول یا پروژه کار می کنند، بهترین کار را انجام می دهد. برای شرکت هایی با تولید مستمر که نیاز به واکنش سریع به تغییرات دارند و تیم هایی که در زمینه پشتیبانی و نگهداری محصول فعالیت می کنند، استفاده از کانبان ضروری است. اسکرامبان برای شرکت های جوان و کوچک بهترین است، زیرا دارای انعطاف پذیری کانبان و ویژگی های اساسی اسکرام است. اسکرامبان را می توان به صورت ترکیبی از اسکرام و کانبان استفاده کرد و مزیت آن این است که کار کنترل بهتری دارد و می تواند بهتر از استفاده از روش آبشاری باشد زیرا در روش آبشاری اغلب پروژه ها به دلیل دشواری تصحیح اشتباهات تکمیل نمی شوند. ترکیب اسکرام، کانبان و روش آبشاری می تواند ارتباط بین ذینفعان را کارآمدتر و موثرتر کند. ترکیبی از این سه روش ذکر شده اسکرامبان فال نامیده می شود. اسکرامبان فال قدرت زیادی برای پاسخگویی به چالش های توسعه نرم افزار در شیوه های مدیریتی مانند مشارکت مستقیم سهامداران خارجی در الزامات پروژه و گنجاندن اسناد تجزیه و تحلیل و همچنین برنامه ریزی، ارزیابی و نظارت پروژه دارد. با توجه به اینکه امروزه متدولوژی های چابک رایج است، بد نیست ببینیم چه زمانی لازم است روش مناسب را انتخاب کنیم. با توجه به ادبیات علمی، می توان دریافت که اسکرام

بیشترین استفاده را دارد، با این حال اندازه گروه در کانبان و XP نسبت به اسکرام سازگارتر است، در حالی که اندازه کار در حال انجام اسکرام یک، دو یا سه ماه در اسپرینت است، اما برآورد کار در حال انجام کانبان و XP نسبتاً کم است. کانبان پیش شرطهای روزانه را سازمان‌دهی می‌کند و XP به طور مداوم نیازهای سازمان‌دهی شده را مجاز می‌کند، در حالی که پیش شرطها باید بر اساس طول اجرا در طول پذیرش اسکرام سازماندهی شوند. کانبان و XP نیز بیشتر بر نرم افزارهای بهبود کیفیت تمرکز دارند.

فصل چهارم

نتیجه گیری و ارائه پیشنهاد ها

4.1 مقدمه

در هر پژوهش، نتایج به دست آمده اهمیت به سزایی دارد؛ زیرا میتواند برای رفع موانع یا بهبود بخشیدن به وضعیت موجود، مبنایی فراهم کند. در فصلهای گذشته مباحث کلی پژوهش شامل، موضوع پژوهش، مبنای نظری، پیشینه پژوهش و یافته های پژوهش بیان گردید. تجزیه و تحلیل و نتیجه گیری از یافته های پژوهش، استفاده کنندگان را قادر میسازد تا با بهره گیری از نتایج به دست آمده، تصمیم های مطلوبی اتخاذ کنند.

4.2 بحث و نتیجه گیری درباره یافته های پژوهش

در توسعه نرم افزار نقش مشتری و ارتباط مستمر با مشتری بسیار مهم است تا بتواند نیازهای او را برآورده کند و به نتیجه نهایی مطلوب برسد. ارائه و مقایسه این روش ها هرگز به دلیلی برای یافتن بهترین روش انجام نمی شود، اما تفاوت های بین آن ها مورد بررسی قرار گرفت و دلایل بالقوه ای یافت شد درباره چرایی انتخاب یک روش و اینکه در چه شرایطی روشی خاص یا ترکیبی از آن ها را میتوان انتخاب کرد. به دلیل گستره وسیعی از مشکلات و الزامات پیش آمده، دستورالعمل واحدی برای مدیریت و رهبری پروژه وجود ندارد. در واقع بهترین روش برای استفاده وجود ندارد. بسته به ماهیت مشکل، روش شناسی مناسب انتخاب خواهد شد. اگر پروژه الزامات و اهداف مشخصی داشته باشد، برخی از روش های سنتی مورد استفاده قرار می گیرند، اما برای پروژه هایی که نیازمندی های ناپایدار هستند، در برخی از آنها همیشه از متدولوژی ها یا ترکیبات چابک استفاده می شوند، زیرا قابل انطباق هستند.

4.3 منابع و مأخذ

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods: Review and analysis. VTT Technical Research Centre of Finland. VTT Publications No. 478. <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>
- Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, 137, 96-113. <https://doi.org/10.1016/j.jss.2017.11.045>
- Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in Software Development: A Systematic Literature Review. *Proceedings of the 2013 39th Euromicro Conference Series on Software Engineering and Advanced Applications*, 9-16. <https://doi.org/10.1109/SEAA.2013.28>
- Ahmad, M.O., Kuvaja, P., Oivo, M., & Markkula, J. (2016). Transition of Software Maintenance Teams from Scrum to Kanban. *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 5427-5436. <https://doi.org/10.1109/HICSS.2016.670>
- Ahmad, M. O., Markkula, J., Oivo, M., & Kuvaja, P. (2014). Usage of Kanban in Software Companies: An empirical study on motivation, benefits and challenges. *Proceedings of the Ninth International Conference on Software Engineering Advances (ICSEA 2014)*, 150-155. ISBN: 978-1-61208-367-4
- Alaidaros, H., Omar, M., & Romli, R. (2018). Towards an Improved Software Project Monitoring Task Model of Agile Kanban Method. *International Journal of Supply Chain Management (IJSCM)*, 7(3), 118-125. <https://core.ac.uk/download/pdf/230745993.pdf>

Al-Baik, O., & Miller, J. (2015). The kanban approach, between agility and leanness: a systematic review. *Empirical Software Engineering*, 20(6), 1861-1897. <https://doi.org/10.1007/s10664-014-9340-x>

Albarqi, A. A., & Qureshi, R. (2018). The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development. *International Journal of Information Engineering and Electronic Business*, 3, 23-35. <https://10.5815/ijieeb.2018.03.04>

Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press. ISBN: 0984521402

Andrei, B-A., Casu-Pop, A-C., Gheorghe, S-C., & Boiangiu, C-A. (2019). A study on using waterfall and agile methods in software project management. *Journal Of Information Systems & Operations Management (JISOM)*, 13(1), 125-135. <http://jisom.rau.ro/Vol.13%20No.1%20-%202019/JISOM-SU19-A12.pdf>

Antill, M. J., & Woodhead, W.R. (1991). *Critical Path Methods in Construction Practice* (4th Edition). John Wiley & Sons. ISBN: 978-0-471-62057-0

Barash, I. (2013). Use of Agile with XP and Kanban Methodologies in the Same Project. *PM World Journal*, 11(X). <https://pmworldlibrary.net/wp-content/uploads/2013/10/pmwj15-oct2013-barash-use-of-agile-xp-kanban-same-project-SecondEdition.pdf>

Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change* (2nd Edition). Addison-Wesley Professional. ISBN: 0321278658. <https://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf>

Beck, K., & Fowler, M. (2000). *Planning extreme programming*. Addison-Wesley Longman Publishing Co., Inc. ISBN:978-0-201-71091-5

Bhavsar, K., Shah, V., & Gopalan, S. (2020). Scrumbanfall: An Agile Integration of Scrum and Kanban with Waterfall in Software Engineering. *International*

Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(4), 2075-2084. <https://10.35940/ijitee.D1437.029420>

Brezočnik, L., & Majer, Č. (2016). Comparison of agile methods: Scrum, Kanban, and Scrumban. Proceedings of the 19th International Multiconference Information Society - IS 2016: Collaboration, software and services in information society, 30-33. http://library.ijs.si/Stacks/Proceedings/InformationSociety/2016/IS2016_Volume_C%20-%20CSS.pdf

Cocco L., Mannaro K., Concas G., Marchesi M. (2011). Simulating Kanban and Scrum vs. Waterfall with System Dynamics. Processes in Software Engineering and Extreme Programming. XP 2011. Lecture Notes in Business Information Processing, 117-131. https://doi.org/10.1007/978-3-642-20677-1_9

Cohn, M. (2007, April 6). Differences Between Scrum and Extreme Programming. Mountain Goat Software. <https://www.mountaingoatsoftware.com/blog/differences-between-scrum-and-extreme-programming/comments>

dos Santos, P. S. M., Beltrão, A. C., de Souza, B. P., & Travassos, G. H. (2018). On the benefits and challenges of using kanban in software engineering: a structured synthesis study.

Journal of Software Engineering Research and Development, 6(1). <https://doi.org/10.1186/s40411-018-0057-1>

Engelhardt, N. (2019). Comparison of Agile and Traditional Project Management: Simulation of Process models. Acta academica karviniensia, 19(2), 15-27. <https://10.25142/aak.2019.011>

Fabac, R., Pihir, I., & Radošević, D. (2009). The Application of Modern Methodologies and ICT in the Management of Projects in Croatian Companies. Proceedings of the 1st International Conference on Information Society and Information Technologies, ISIT 2009. <https://ssrn.com/abstract=2505743>

Fagarasan, C., Popa, O., Pislă, A., & Cristea, C. (2021). Agile, waterfall and iterative approach in information technology projects. Proceedings of the IOP Conference Series: Materials Science and Engineering, 1169(1). <https://10.1088/1757-899X/1169/1/012025>

Ferrão, S. É. R., & Canedo, E. D. (2015). A study of the applicability of an agile methodology scrum allied to the Kanban method. Proceedings of the 10th Iberian Conference on Information Systems and Technologies (CISTI). <http://dx.doi.org/10.1109/CISTI.2015.7170382>

Flora, H. K., & Chande, S. V. (2014). A Systematic Study on Agile Software Development Methodologies and Practices. International Journal of Computer Science and Information Technologies (IJCSIT), 5(3), 3626-3637. https://www.researchgate.net/publication/266742256_A_Systematic_Study_on_Agile_Software_Development_Methodologies_and_Practices

Fustik, V. (2017). Scrum Methodology Compared with Other Methodologies Applied in the Software Development Projects. Proceedings of the International Conference on Information Technologies (InfoTech-2017). <http://infotech-bg.com/sites/default/files/2017/A-VF.pdf>

Garcia, A. L., da Rocha Miguel, I., Eugênio, J. B., da Silva Vilela, M., & Marcondes, G. A. B. (2020). Scrum-Based Application for Agile Project Management. Journal of Software, 15(4), 106-113. <https://10.17706/jsw.15.4.106-113>

Granulo, A., & Tanović, A. (2019). Comparison of SCRUM and KANBAN in the Learning Management System implementation process. Proceedings of the 27th Telecommunications forum TELFOR 2019, 636-639. <https://doi.org/10.1109/TELFOR48224.2019.8971201> Holz, B. (2019). Results Summary: Agile in the Enterprise. Gartner. [https://circle.gartner.com/Portals/2/Resources/pdf/Agile%20in%20the%20Enterprise%202019%20-%20Results%20Summary%20\(updated\).pdf](https://circle.gartner.com/Portals/2/Resources/pdf/Agile%20in%20the%20Enterprise%202019%20-%20Results%20Summary%20(updated).pdf) <https://doi.org/10.1002/hfm.4530040310>

Ingason, H. T., Gestsson, E., & Jonasson, H. I. (2013). The Project Kanban Wall: Combining Kanban and Scrum for Coordinating Software Projects. *PM World Journal*, II(VIII). <https://pmworldlibrary.net/wp-content/uploads/2013/08/pmwj13-aug2013-Ingasson-Gestsson-Jonasson-Project-Kanban-Wall-FeaturedPaper.pdf>

Islam, K.A. (2013). *Agile Methodology for Developing & Measuring Learning: Training Development for Today's World*. AuthorHouse. ISBN: 9781491823880

Javanmard, M. & Alian, M. (2015). Comparison between Agile and Traditional software development methodologies. *Cumhuriyet Science Journal*, 36(3), 1386-1394. <https://dergipark.org.tr/en/pub/cumuscij/issue/45132/564438>

Kanwal, F., Junaid, K., & Fehiem, M. A. (2010). A Hybrid Software Architecture Evaluation Method for FDD - An Agile Process Model. *Proceedings of the International Conference on Computational Intelligence and Software Engineering (CiSE)*, 1-5. <https://doi.org/10.1109/CISE.2010.5676863>

Karunanithi, K. (2016). *Metrics in Agile and Kanban , Software Measurement Techniques*. Project: Software Measurement. California State University. https://www.researchgate.net/publication/305613018_Metrics_in_Agile_and_Kanban_Software_Measurement_Techniques

Kirovska, N., & Koceski, S. (2015). Usage of Kanban methodology at software development teams. *Journal of Applied Economics and Business*, 3(3), 25-34. <http://www.aebjournal.org/articles/0303/030302.pdf>

Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum - Making the Most of Both*. C4Media Inc. http://www.agileinnovation.eu/wordpress/wp-content/uploads/2010/09/KanbanAndScrum_MakingTheMostOfBoth.pdf

Laanti, M. (2013). Agile and Wellbeing -- Stress, Empowerment, and Performance in Scrum and Kanban Teams. *Proceedings of the 46th Hawaii International Conference on System Sciences*, 4761-4770. <https://doi.org/10.1109/HICSS.2013.74>

Ladas, C. (2008). Scrumban-Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press.

Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43, 59-67. <https://doi.org/10.1016/j.rcim.2015.12.001>

Livermore, A. J. (2008). Factors that Significantly Impact the Implementation of an Agile Software Development Methodology. *Journal of Software*, 3(4), 31-36. <http://www.jssoftware.us/vol3/jsw0304-04.pdf>

Mahnič, V. (2014). Improving software development through combination of scrum and kanban. *Proceedings of the 8th WSEAS International Conference on Computer Engineering and Applications (CEA '14)*, 281-288. <http://www.wseas.us/e-library/conferences/2014/Tenerife/INFORM/INFORM-40.pdf>

Mahnič, V. (2015). From Scrum to Kanban: introducing Lean principles to a software engineering capstone course. *International Journal of Engineering Education*, 31(4), 1106-1116

Malik, R. S., Ahmad, S.S., & Hussain, M.T.H. (2019). A Review of Agile Methodology in IT Projects. *Proceedings of the 2nd International Conference on Advanced Computing and Software Engineering (ICACSE-2019)*, 655-656. <http://dx.doi.org/10.2139/ssrn.3351064>

Marić, M. (2015). Metodološko-radni okvir za razvoj softverske arhitekture poslovnog softvera u agilnim procesima [A Methodological Framework for Developing the Software Architecture of Business Software in Agile Processes]. [Doctoral dissertation, University of Novi Sad, Faculty of Economy, Subotica]. PQDT Open. https://hdl.handle.net/21.15107/rcub_nardus_5151

McCormick, M. (2012). Waterfall vs. Agile Metodology. MPCs, Inc. http://mccormickpcs.com/images/Waterfall_vs_Agile_Methodology.pdf

Merzouk, S., Elhadi, S., Cherkaoui, A., Marzak, A., & Sael, N. (2018). Agile Software Development: Comparative Study. Proceedings of The second International Conference on Smart Application and Data Analysis for Smart Cities (SADASC'18). <http://dx.doi.org/10.2139/ssrn.3186323>

Milošević, D. (2018 December 2). Agilna metodologija [Agile methodology]. Analitika i vizuelizacija podataka [Data analytics and visualization]. <https://dusanmilosevic.com/agilna-metodologija/>

Mircea, E. (2019). Project Management using Agile Frameworks. Economy Informatics, 19(1), 34-44. <https://10.12948/ei2019.01.04>

Mohan, A., Devisree, A. S., & Kumar, C. V. P. (2019). Rationalized Scrumban Based Methodology for Improving the Slackening Pension Scheme of Employee Sectors in India. International Journal of Recent Technology and Engineering (IJRTE), 8(1), 1525-1550. <https://www.ijrte.org/wp-content/uploads/papers/v8i1/A9236058119.pdf>

Moonden, Y. (2011). Toyota Production System: An Integrated Approach to Just-In-Time (4th Edition). CRC Press, Taylor & Francis Group. <https://doi.org/10.1201/b11731>

Nikitina, N., & Kajko-Mattsson, M. (2011). Developer-driven big-bang process transition from Scrum to Kanban. Proceedings of the 2nd Workshop on Software Engineering for Sensor Network Applications - SESENA '11, 159-168. <https://doi.org/10.1145/1987875.1987901>

Pantelić, O., Nikolić, U., & Krstović, S. (2020). Razvoj softvera za upravljanje projektima zasnovan na različitim agilnim metodologijama [Development of project management software based on diferent agile methodologies]. Časopis za informacionu tehnologiju i multimedijalne sisteme, (71), 38-43 [Journal for information technology and multimedia systems, (71), 38-43].

Pap, J. (2008). Ekstremno programiranje kao metoda agilnog razvoja softvera [Extreme programming as a method of agile software development]. [Graduate thesis, University of Novi Sad, Faculty of Science]. PQDT Open. http://www.robertpapp.info/docs/theses/PappRobert_XP_BSc-thesis.pdf

- Parker, M. E. F., & del Monte, Y. F. (2014). The Agile Management of Development Projects of Software Combining Scrum, Kanban and Expert Consultation. Proceedings of the 10th IFIP International Conference on Open Source Systems (OSS), 176-180. https://10.1007/978-3-642-55128-4_25
- Raman, S. (2014, April 2). eXtreme Programming The Methodology. InfoQ. <https://www.infoq.com/articles/implementing-xp-methodology/>
- Rising, L., Janoff, N.S. (2000). The Scrum software development process for small teams. IEEE Software, 17(4), 26-32. <https://10.1109/52.854065>
- Saleh, S. M., Huq, S. M., & Rahman, M. A. (2019). Comparative Study within Scrum, Kanban, XP Focused on Their Practices. Proceedings of the International Conference on Electrical, Computer and Communication Engineering (ECCE), 1-6. <https://doi.org/10.1109/ECACE.2019.8679334>
- Schwaber, K. (2004). Agile Project Management with Scrum. Microsoft Press. ISBN:978-0-7356-1993-7
- Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. Scrum.org. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>
- Schwaber, K., Beedle, M. (2002). Agile Software Development with Scrum. Upper Saddle River, New Jersey: Prentice Hall. ISBN: 0130676349
- Senapathi, M., & Drury-Grogan, M. L. (2021). Systems Thinking Approach to Implementing Kanban: A case study. Journal of Software: Evolution and Process, 33(4). <https://doi.org/10.1002/smr.2322>
- Shafiq, S., & Inayat, I. (2017). Towards Studying the Communication Patterns of Kanban Teams: A Research Design. Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), 303-306. <https://doi.org/10.1109/REW.2017.34>
- Shalloway, A. (2011). Demystifying Kanban. Cutter IT Journal, 24(3), 12-17.

Shenhar, J. A., & Dvir, D. (2007). Reinventing Project Management: The Diamond Approach To Successful Growth And Innovation. Harvard Business Publishing

Sjøberg, D. I. K., Johnsen, A., Solberg, J. (2012). Quantifying the Effect of Using Kanban versus Scrum: A Case Study. IEEE Software, 29(5), 47-53. <https://10.1109/MS.2012.110>

Skinner, R., Land, L., Chin, W., & Nelson, R. R. (2015). Reviewing the Past for a Better Future: Reevaluating the IT Project Retrospective. Proceedings of the 10th International Research Workshop on Information Technology Project Management (IRWITPM), 110-119. <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1008&context=irwitpm2015>

Stellman, A., & Greene, J. (2017). Head First Agile: A Brain-Friendly Guide. O'Reilly Media, Inc. ISBN: 9781449314330

Subih, M. A., Malik, B. H., Mazhar, I., ul-Hassan, I., Sabir, U., Wakeel, T., Ali, W., Yousaf, A., bin-Ijaz, B., & Nawaz, H. (2019). Comparison of Agile Method and Scrum Method with Software Quality Affecting Factors. International Journal of Advanced Computer Science and Applications (IJACSA), 10(5), 531-535. https://thesai.org/Downloads/Volume10No5/Paper_69_Comparison_of_Agile_Method_and_Scrum_Method.pdf

Sutherland, J. (2010). Jeff Sutherland's Scrum Handbook. The Scrum Training Institute. https://www.researchgate.net/publication/301685699_Jeff_Sutherland's_Scrum_Handbook

Tadić, B. (2005). Ekstremno programiranje i primjena na Balkanu [Extreme programming and application in the Balkans]. Proceedings of the 4. Naučno-stručni skup sa međunarodnim učešćem "KVALITET 2005" [4th Scientific-professional conference with international participation "QUALITY 2005"], 237-246. <http://www.quality.unze.ba/zbornici/QUALITY%202005/034-Q05-028.pdf>

Terlecka, K. (2012). Combining Kanban and Scrum -- Lessons from a Team of Sysadmins. Proceedings of the Agile Conference (AGILE), 99-102. <http://dx.doi.org/10.1109/Agile.2012.20>

Venkatachalam, A., Rajkumar, S., Selvanathan, N., & Karthick, K. (2017). Agile SCRUM – The Revolution in IT Industry. Proceedings of the International Conference on Intelligent Computing Systems (ICICS 2017 – Dec 15th - 16th 2017), 228-238. <http://dx.doi.org/10.2139/ssrn.3122169>

Verweij, D., & Massen, O. (2011). Kanban at an Insurance Company in the Netherlands. Cutter IT Journal, 24(3), 18-22.

Vladimirovich Orlov, E., Mikhailovna Rogulenko, T., Alexandrovich Smolyakov, O., Vladimirovna Oshovskaya, N., Ivanovna Zvorykina, T., Grigorevich Rostanets, V., & Petrovna Dyundik, E. (2021). Comparative Analysis of the Use of Kanban and Scrum Methodologies in IT Projects. Universal Journal of Accounting and Finance, 9(4), 693-700. <https://10.13189/ujaf.2021.090415>

Vresk, A. (2020). Agilne metode za upravljanje projektima [Graduate thesis, University of Zagreb, Faculty of Organization and Informatics]. PQDT Open. <https://urn.nsk.hr/urn:nbn:hr:211:105099>

Womack, J., Jones, D., & Roos, D. (1990). The machine that changed the world. Rawson Associates.