



دانشکده فنی و مهندسی
کارشناسی ارشد مهندسی کامپیوتر نرم افزار
گروه مهندسی کامپیوتر و فناوری اطلاعات
گزارش تحقیق درسی

موضوع:

بررسی پایان نامه آقای دیندرا گیمایر (Dipendra Ghimire)

دانشجوی دکترای دانشگاه لینکلن

با موضوع:

تأثیر رابطه بین مشتری و تیم توسعه نرم افزار بر نتیجه پروژه توسعه نرم افزار

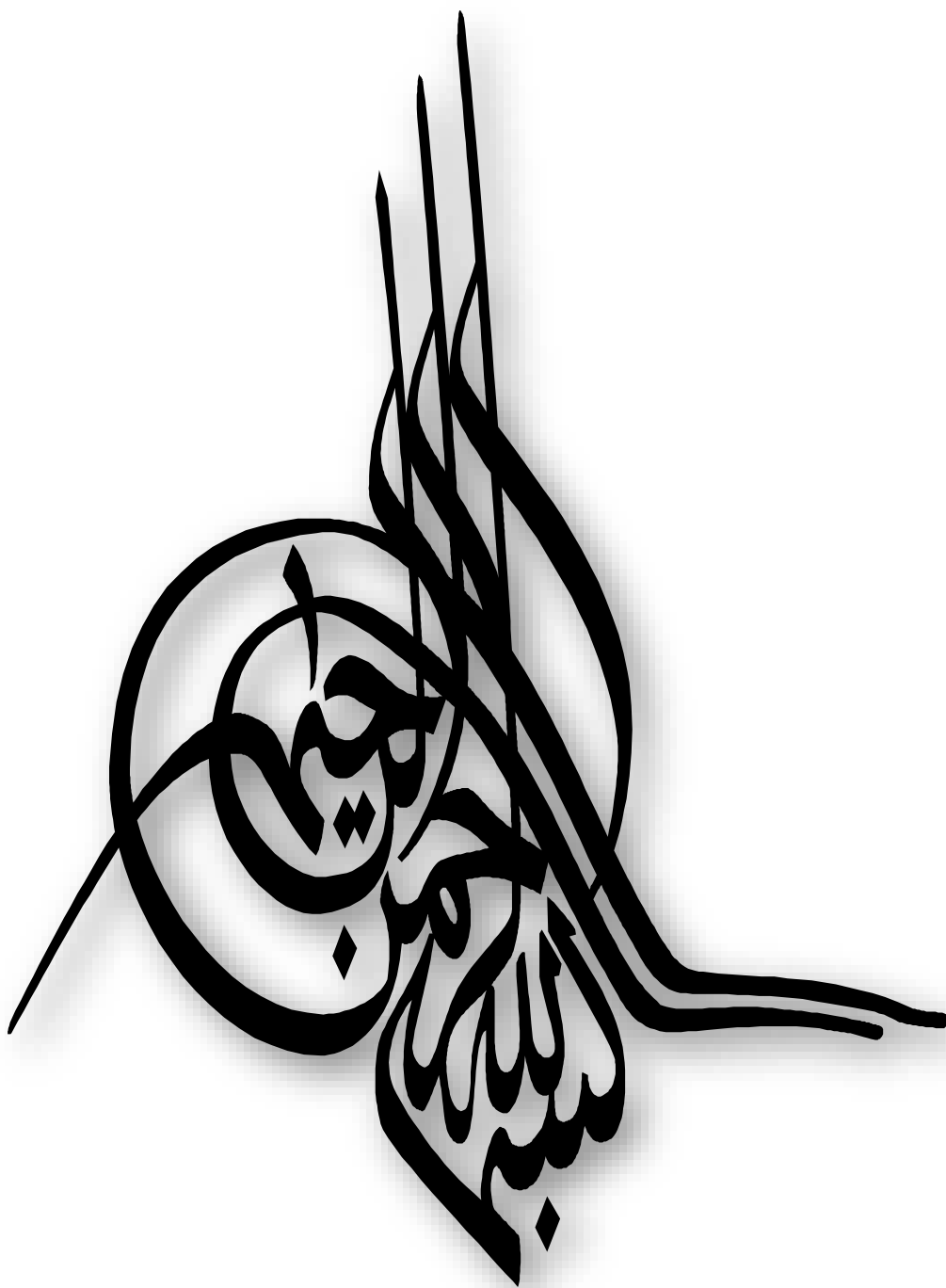
نگارش:

فهیمه ابوحمزه

استاد راهنما:

دکتر علی رضوی

شهریور ۱۴۰۰



چکیده

در این گزارش پایان‌نامه آقای گیمایر را بررسی می‌کنیم. ایشان در خصوص رابطه بین تیم توسعه نرم‌افزار و صاحب محصول، در توسعه نرم‌افزار و تاثیر این رابطه بر میزان موفقیت پروژه با استفاده از روش Agile می‌پردازد. در این پایان‌نامه با استفاده از شواهد تجربی و نظرسنجی و بررسی میزان موفقیت سازمان‌هایی که از روش چابک استفاده می‌کنند یا کرده‌اند تاثیر مثبت رابطه دائم تیم توسعه دهند نرم‌افزار و صاحب محصول را در اتمام موفقیت آمیز پروژه نشان می‌دهد. از مدل کار گروهی می‌توان برای درک روابط در تیم با استفاده از توسعه نرم‌افزار Agile استفاده کرد. توسعه نرم‌افزار چابک مجموعه‌ای از شیوه‌ها و نقش‌ها را برای سازماندهی کار در طول یک پروژه توسعه نرم‌افزار اتخاذ می‌کند. با این حال، برخی از چالش‌ها در توسعه نرم‌افزار چابک وجود دارد که بر روابط بین تیم توسعه و مالک محصول تأثیر می‌گذارد. این تحقیق این چالش‌ها را با Agile مشخص کرده و با استفاده از مدل کار تیمی تأثیر آنها را بر روابط توضیح می‌دهد.

فهرست منابع

فصل اول.....	۵
مقدمه.....	۵
مقدمه.....	۶
۱-۱ تعریف مسئله و بیان سوال‌های اصلی تحقیق.....	۶
۲-۱ روش انجام تحقیق.....	۸
۳-۱ مراحل انجام تحقیق.....	۹
۴-۱ ساختار گزارش سمینار.....	۲۰
فصل دوم.....	۲۱
مفاهیم عمومی و روش‌های بدست‌آوری داده‌ها.....	۲۱
مقدمه.....	۲۲
روش بدست آوردن داده‌های تحقیق.....	۲۲
فصل سوم.....	۲۷
جمع‌بندی و پیشنهادها.....	۲۷
مقدمه.....	۲۸
نتایج حاصل چه بوده‌اند و چگونه این نتایج ارزیابی شده‌اند.....	۲۸
نتیجه‌گیری و جمع‌بندی پژوهش.....	۴۸
چه پیشنهاداتی را برای ادامه کار ارائه نموده است.....	۵۲
منابع.....	۵۴

فصل اول

مقدمه

مقدمه

تفکر چابک یا Agile، مجموعه‌ای از اصول و ارزش‌ها است که استفاده از آن‌ها در محیط توسعه محصولات نرم افزاری، به نتایجی مانند محصولات مفید و کارآمد، مشتریان راضی و نیروی کار باانگیزه می‌انجامد. در واقع استفاده از این متدولوژی باعث می‌شود تولید نرم افزارها کاملاً مطابق با نیاز مشتریان انجام شود. چرا که طبق این دیدگاه، مشتری از مهم‌ترین افراد در تولید پروژه بوده و تنها کسی است که از نیازهای واقعی نرم افزار مطلع است. خوب است بدانید، در روش چابک محصولات به صورت فازبندی به مشتری تحویل داده می‌شود و مشتری با تیم پروژه کاملاً در ارتباط است. بنابراین، محصول نهایی همان چیزی خواهد شد که مدنظر اوست.

۱-۱ تعریف مسئله و بیان سوال‌های اصلی تحقیق

برای جواب این سوال سراغ مقدمه این پایان‌نامه می‌رویم. در مقدمه این مقاله نویسنده سعی کرده است مختصری در خصوص روند توسعه نرم‌افزار و متدهایی که تا کنون استفاده شده صحبت کند و نقاط ضعف متدهای قبلی را ذکر می‌کند. در ابتدا محقق به بیان مختصری از تاریخچه روش‌های توسعه نرم‌افزار در پروژه‌های نرم‌افزاری پرداخته است. که می‌توان به شرح زیر اشاره کرد.

توسعه نرم‌افزاری اولیه از مفهوم "کد نویسی و رفع خطا، کد نویسی بیشتر، رفع مشکل بیشتر" بدون برنامه‌ریزی استفاده کرد. روش‌های برنامه‌محور، مبتنی بر فعالیت‌های متوالی، معرفی شدند. روش‌های برنامه‌محور مورد استفاده برای توسعه نرم‌افزار عبارتند از: Waterfall یا آبشاری، Spiral یا مارپیچی و Prototype یا نمونه سازی. این روش‌ها شامل چندین مرحله متوالی مختلف است که عبارتند از: مشخصات مورد نیاز، تجزیه و تحلیل نیازها، طراحی، کدنویسی، آزمایش و نگهداری.

در ادامه به مطرح کردن مشکلات روش‌های قدیمی و علت استفاده از روش دیگر می‌پردازد :

مشکلات روش‌های فوق این است که اغلب اشکالات نرم‌افزاری تا مراحل آخر فرآیند توسعه، زمانی که رفع آنها دشوار است، یافت نمی‌شوند. ممکن است از توسعه دهندگان خواسته شود زمان قابل توجهی را صرف رفع خطاهایی کنند که بر بهره‌وری و کارایی کلی پروژه تأثیر می‌گذارد. مدل توسعه نرم‌افزار آبشاری به دلیل ماهیت مستقیم و

ساخت یافته یکی از پرکاربردترین آنهاست. رویکرد آبشاری یک روش پیش‌بینی برای روند توسعه نرم‌افزار بر اساس مستندات است. فعالیت‌های موجود در این رویکرد شامل تجزیه و تحلیل الزامات، طراحی و آزمایش کد و مستندات دقیق است. فرآیند تعریف شده باید گام به گام دنبال شود. اگرچه مدل آبشار و سایر روش‌های سنتی هنوز در توسعه نرم‌افزار استفاده می‌شود نگرانی‌هایی در مورد سازگاری آهسته آن با تغییر مداوم نیازمندی‌های تجاری و تمایل به فراتر رفتن از بودجه یا عقب افتادن از برنامه وجود دارد.

نتایج و آمارهای برآمده از رویکرد Agile و همچنین مزایای آن را ذکر می‌کند که شامل موارد زیر است:

روشهای توسعه نرم‌افزار چابک به عنوان جایگزینی برای روش‌های سنتی پیشنهاد شد. گزارش شده است که پروژه‌های توسعه نرم‌افزار چابک نسبت به پروژه‌های توسعه نرم‌افزاری سنتی موفق‌تر بوده‌اند. در رویکرد توسعه نرم‌افزار Agile، پروژه‌های نرم‌افزاری با هماهنگی بین مشتری و تیم توسعه، جلو می‌رود. تیم‌های توسعه، خود-سازماندهی‌کننده‌ی کارهای خود هستند، حجم کار را مدیریت و برنامه‌ریزی می‌کنند که پروژه در یک بازه زمانی مشخص تکمیل شود.

این تیم‌های خود-سازماندهی‌کننده چندکاره هستند و باید به یکدیگر اعتماد و درک مشترک از پروژه و توانایی سازگاری با تغییرات پروژه را داشته باشند. در طول پروژه بین اعضای تیم توسعه نرم‌افزار و نماینده مشتری، که اغلب صاحب محصول نامیده می‌شود، تماس مستقیم وجود دارد. به همین دلیل، بین صاحب محصول و مشتری رابطه وجود دارد.

صنعت توسعه نرم‌افزار نمونه‌های زیادی را ارائه کرده است که در پروژه‌های توسعه نرم‌افزار شکست خورده است. بررسی عوامل متعددی در شکست پروژه‌های توسعه نرم‌افزار نقش دارند. شکست یک پروژه توسعه نرم‌افزار، که در آن مقدار زیادی پول و منابع دیگر نیاز به سرمایه‌گذاری بود، باعث شد که محقق این را به عنوان یک منطقه مناسب برای بررسی در نظر بگیرد. محققان مهندسی نرم‌افزار قبلاً رفتار تیم توسعه نرم‌افزار Agile را بررسی کرده‌اند اما تا کنون رابطه با مشتری را بررسی نکرده‌اند.

هدف از این تحقیق درک رابطه بین تیم توسعه نرم‌افزار و صاحب محصول در پروژه‌های توسعه نرم‌افزار Agile و تأثیر آن رابطه بر نتایج پروژه است.

سوالات تحقیق و یا فرضیه ها

سوال تحقیقی اصلی سطح بالا که باید مورد بررسی قرار گیرد عبارت است از:

"رابطه بین تیم توسعه نرم افزار و صاحب محصول چه تاثیری بر نتایج پروژه نرم افزار Agile دارد؟"

سوالات فرعی تحقیق مورد بحث در این پایان نامه شامل موارد زیر است:

- ۱- آثار و یافته ها حاکی از آن است که بین تیم توسعه و نماینده مشتری در پروژه هایی که به سبک Agile هستند، چالشی وجود دارد. این چالشها و مباحث تا چه حدی هستند؟
 - ۲- چه نوع چالشهایی در توسعه نرم افزارها به سبک Agile وجود دارد؟
 - ۳- تاثیر این چالشها بر روابط بین مشتری و تیم توسعه نرم افزار به سبک Agile چیست؟
 - ۴- آیا ماهیت روابط بین مشتری و تیم توسعه تاثیری بر نتیجه پروژه دارد؟
 - ۵- آیا مدل کار گروهی می تواند به طور کافی ماهیت رابطه بین مشتری و تیم توسعه را در زمینه توسعه نرم افزار به سبک Agile توضیح دهد؟
- برای درک این رابطه، چالش های توسعه نرم افزار Agile بررسی شده است. پس از شناسایی چالش ها و درک تاثیر آنها بر روابط، بازنگری در مدل کار گروهی، برای شرح روابط پیشنهاد می شود.

۱-۲ روش انجام تحقیق

این سوال سطح بالایی است که این تحقیق به آن می پردازد، بررسی تاثیر روابط بین تیم توسعه نرم افزار و نماینده مشتری بر نتایج پروژه است. هدف از تحقیق اغلب آزمایش یک نظریه یا استفاده از یک نظریه برای توضیح نتایج تحقیق با ارائه یک پایه مناسب است که بر اساس آن توضیحات را ارائه می دهد. مطالعات تحقیقی در مورد روابط بین گروه های افراد ممکن است تحت تاثیر شرایط مختلف، هم موقعیتی و هم شناختی قرار گیرد. برای انجام این کار، از پارادایم های متعددی استفاده می شود، مانند اثبات گرایی، تفسیری، دگرگون کننده و عملگرا).

(positivist, interpretivist, transformative pragmatic)

در این تحقیق از الگوی عمل گرا و واقع بینانه (pragmatic) برای پاسخ به سوالات تحقیق استفاده شده است. الگوی عمل گرا شامل ابزارهای اثبات گرایی و تفسیری است و به فرایند تولید دانش در تحقیقات کمک می کند، که شامل تولید و آزمایش فرضیه ها می شود.

برای پاسخ به سوالات تحقیق، بررسی دیدگاه ها و تجربه شرکت کنندگان در محیط توسعه نرم افزار مورد نیاز است. بنابراین، لازم است هم تجارب شرکت کنندگان در شرکت های توسعه نرم افزار را کاوش کرده و هم توضیح دهیم، نه اینکه فقط به بررسی یا توضیح آنها بپردازیم. برای این تحقیق مهم است که از اعضای تیم توسعه دریابیم که چرا آنها معتقدند چالش ها رخ می دهند. به منظور دستیابی به پاسخهای دقیق و عمیق برای سوالات تحقیق، روش ترکیبی پیشنهاد شده است.

در روش ترکیبی، داده های کمی و کیفی بر اساس سوالات تحقیق جمع آوری می شود. یک روش ترکیبی به عنوان یک استراتژی استفاده می شود که به موجب آن می توان قدرت پرسشنامه را با درک جزئیات اطلاعات از طریق مصاحبه های نیمه ساخت یافته افزایش داد. این مطالعه شامل پرسشنامه ای است که هدف آن درک فرآیندهای مبتنی بر چابک و چالش های پیش آمده در پروژه های توسعه نرم افزار چابک، همراه با مصاحبه نیمه ساخت یافته است. پرسشنامه ای انتخاب شد که وسیله ای سریع، ارزان و دقیق برای دسترسی به اطلاعات ارائه می دهد. از روشهای ترکیبی می توان برای بهبود قدرت تجزیه و تحلیل با ترکیب تکنیک های کمی و کیفی استفاده کرد. به عنوان مثال، روشهای ترکیبی به توسعه ابزارهای خاص زمینه کمک می کند و درک کامل و گسترده ای از مشکلات تحقیق ارائه می دهد.

۱-۳ مراحل انجام تحقیق

طرح پژوهش

این مطالعه برای درک رابطه بین تیم های توسعه نرم افزار و صاحب محصول/مشتري طراحی شده است. از این رو، این مطالعه به عنوان یک پروژه چند مطالعه ای طراحی شده است. بخش ۱ این تحقیق شامل شرکت کنندگان در نقش های توسعه نرم افزار بود که در آنها رویکردهای چابک اتخاذ شد. بخش دوم این مطالعه شامل شرکت کنندگانی بود که در نقش صاحب محصول کار کرده بودند.

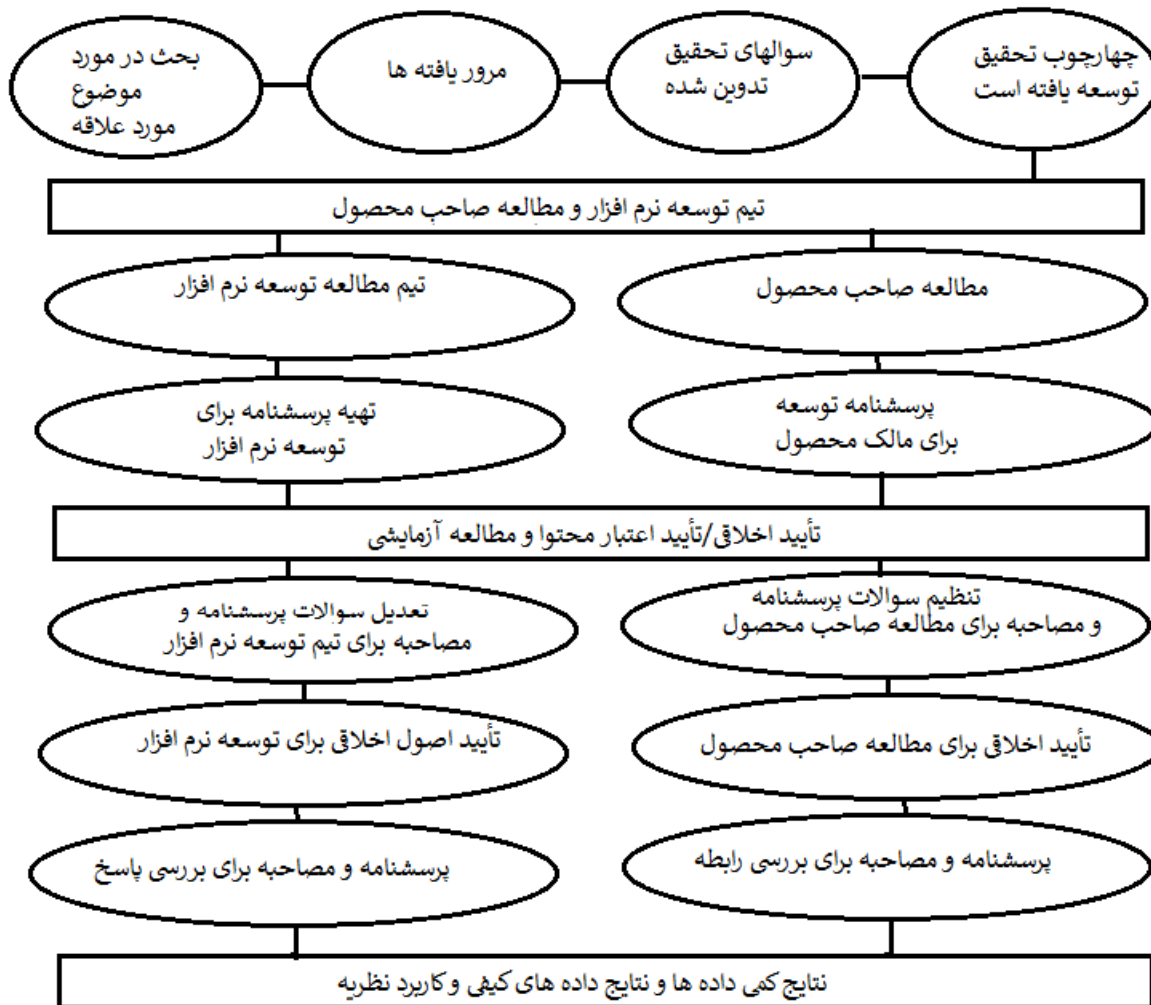
ایجاد یک ماتریس به مشاهده چارچوب نظری مورد استفاده برای بررسی رایج ترین عواملی که در ایجاد روابط کاری در تیم های توسعه نرم افزار Agile، از جمله مالک محصول دخیل بوده است، کمک کرد. مدل کار گروهی برای این مطالعه انتخاب شد زیرا چارچوب این مدل می تواند برای بررسی اجزای بیشتری که در روابط بین تیم های توسعه نرم افزار و صاحب محصول تأثیر دارند، مورد استفاده قرار گیرد.

یک پرسشنامه با استفاده از چارچوب مدل کار گروهی برای اطلاع از توسعه سوالات ساخته شد. این پرسشنامه شامل اطلاعات پیش زمینه، مانند نقش آنها، زمانی که در توسعه نرم افزار کار کرده اند و اندازه یک سازمان است. این پرسشنامه همچنین شامل سوالاتی برای جمع آوری اطلاعات در مورد چالش ها و عواملی است که در توسعه نرم افزار چابک موثر بوده است.

برای اولین مرحله از روش جمع آوری داده ها، پرسشنامه ای تهیه شد. اعتبار محتوا و مطالعه آزمایشی پرسشنامه انجام شد، مرحله دوم بخش ۱ این مطالعه شامل مصاحبه نیمه ساخت یافته با زیر مجموعه افرادی که پرسشنامه آنلاین را تکمیل کرده بودند، بوده است.

مراحل مربوط به تحقیق

در شکل ۳-۱ نشان داده شده است و در قسمت زیر به تفصیل توضیح داده شده است.



شکل ۳-۱ مروری بر طرح تحقیق

این تحقیق شامل دو مطالعه است: مطالعه تیم‌های توسعه نرم‌افزار و مطالعه صاحبان محصولات. یک ابزار جداگانه برای هر مطالعه برای درک دیدگاه‌های تیم‌های توسعه نرم‌افزار Agile و نماینده مشتری/مالک محصول، برای کشف چالش‌ها و توضیح روابط در پروژه‌های توسعه نرم‌افزار Agile توضیح داده شد. اعتبار سنجی و جمع‌آوری داده‌ها برای هر مطالعه در بخش‌های زیر توضیح می‌دهد:

توسعه ابزار

قبل از جمع‌آوری داده‌ها، ابزار تحقیق طراحی شد. بدین منظور، لیستی از سوالات متمرکز بر تحقیق تهیه شد. این ابزارهای تحقیق پس از در نظر گرفتن موضوع تحقیق، مرور یافته‌ها و مطابق با آن نوشته شده است. همانطور که ذکر شد، چارچوب نظری برای این مطالعه انتخاب شده است. پرسشنامه اولیه شامل ۶۰ مورد و پنج بخش بود:

- داده‌های پیشین تیم‌های توسعه نرم‌افزار/مالک محصول

- نوع سازمان

- فرآیندی که از شیوه Agile استفاده می‌کند.

- اطلاعات پروژه

- مشکلات پیش روی پروژه

به منظور اعتبار بخشیدن به ابزار تحقیق توسعه یافته مراحل زیر انجام شد.

اعتبارسنجی ابزار

اعتبارسنجی ابزار برای این انجام می‌شود تا مشخص شود آیا ابزار اندازه‌گیری مورد نظر، اندازه‌گیری را انجام می‌دهد یا خیر. اعتبار ابزار در این تحقیق این بود که ببینیم آیا پرسشنامه‌ای که برای تحقیق تهیه شده است برای پاسخ دهندگان هدف مناسب بود. فرآیندهای مختلف اعتبارسنجی ابزار وجود دارد، مانند پیش آزمون یا مطالعات آزمایشی، اعتبار محتوا، اعتبار سازه و قابلیت اطمینان. فرایند اعتبار سنجی مورد استفاده برای این تحقیق در زیر مورد بحث قرار گرفته است.

تأیید اعتبار محتوا

اعتبار سنجی محتوا به اندازه پرسشنامه تهیه شده برای این مطالعه وابسته است و نشان دهنده محتوای مورد سنجش است. فرآیند اعتبارسنجی محتوا به یک محقق امکان می‌دهد تا دریابد که آیا این معیار شامل سوالاتی در تمام بخشهای مربوط، به منظور آزمایش مشکل است. تأیید محتوا را می‌توان از طریق قضاوت افرادی با تجربه در یک زمینه خاص، که به آنها هیئت متخصصان نیز گفته می‌شود، انجام داد. در این مطالعه، پنل‌هایی با استفاده از افراد با تجربه در توسعه نرم‌افزار تشکیل شد. به طور معمول، افرادی که برای تأیید اعتبار محتوا انتخاب شده‌اند، افرادی حرفه‌ای بوده‌اند که در زمینه مرتبط نتایج مطالعات خود را چاپ و منتشر کرده‌اند. در این مطالعه، متخصصان اعتبارسنجی محتوا افرادی بودند که در توسعه نرم‌افزار مطالبی را چاپ کرده یا در توسعه نرم‌افزار مشارکت داشته‌اند. برای این فرآیند، کارشناسان محتوا از طریق شبکه‌های مختلف، از جمله فارغ التحصیلان دانشگاهی محاسبات و توسعه نرم‌افزار و متخصصان Agile Professional Network2 در Christchurch حضور داشتند. پس از شناسایی نامزدهای احتمالی، یک درخواست ایمیل برای مشارکت ارسال

شد. در مجموع، از چهار متخصص برای تأیید اعتبار محتوا استفاده شد. کارشناسان از طریق پرسشنامه کار کردند و بازخورد ارائه دادند.

رویکرد اعتبارسنجی محتوا در این مطالعه به عنوان یکی از روش‌های کیفیت سنجی داده‌ها با استفاده از نظر افراد متخصص، استفاده از شاخص اعتبارسنجی محتوا (CVI) است. از شاخص اعتبارسنجی محتوا برای محاسبه ارتباط سطوح مورد استفاده قرار می‌گیرد که توسط گروهی از متخصصان ارزیابی شده است که حداقل سه متخصص توصیه می‌شود، اما بهترین تعداد بین سه تا ده نفر خواهد بود. رتبه بندی اقلام معمولاً در مقیاس ترتیبی ۴ درجه‌ای است. و پیشنهاد می‌شود که مقیاس‌های رتبه بندی ۳ یا ۵ درجه‌ای مورد استفاده قرار بگیرد، اما ترجیح می‌دهد از مقیاس ۴ درجه‌ای برای جلوگیری از داشتن نقطه خنثی استفاده کند. برای هر مورد، CVI به عنوان تعداد متخصصانی که دارای رتبه ۳ یا ۴ هستند تقسیم بر تعداد کل متخصصان محاسبه شود. به عنوان مثال، موردی که توسط ۳ از ۴ داور به طور کامل یا بسیار مرتبط ارزیابی شده است، CVI 0.75 دارد.

یکی از نگرانی‌های این رویکرد این بود که CVI بر توافق و ارتباط متمرکز بود بدون در نظر گرفتن اینکه آیا این ابزار شامل مواردی است که آنچه را که برای اندازه‌گیری در نظر گرفته شده است، اندازه‌گیری کند. نگرانی دیگر در مورد این رویکرد این بود که هیچ ملاحظه‌ای برای توافق اتفاقی در نظر گرفته نشده بود. به منظور تعیین توافقنامه تصادفی، CVI را با ضریب کاپا (*k) تعریف کرد. این ضریب کاپا این بود که اطمینان حاصل شود که اقلام موجود در ابزار به همان چیزی که برای آن ساخته شده بود، وفادار بماند. برای تأیید اعتبار محتوا از CVI با توافق اتفاقی استفاده شد

محاسبه توافق احتمالی شانس به شرح زیر است:

$$PC = ((N! / (A! (N-A)!)) / 5N$$

جایی که N = تعداد کارشناسان و A = تعداد متخصصان در مورد اهمیت توافق دارند.

معادله مورد استفاده برای محاسبه توافقنامه شانس عبارت است از:

$$K^* = (I - CVI - PC) / (1 - PC)$$

ارزیابی برای محاسبه هر مورد K^* مقدار در * ارائه شده است

ارزش k^* برای هر مورد	ارزیابی
>0.74	عالی
$0.60 \Rightarrow 0.74$	خوب
$0.40 \Rightarrow 0.60$	منصفانه (برای تجدید نظر در نظر گرفته شده است)
<0.40	برای حذف در نظر بگیرید

مقدار K^* برای هر مورد محاسبه شد. (برای مشاهده جدول K^* به پیوست A مراجعه کنید). از ۶۰ مورد، ۵۴ مورد عالی و ۶ مورد نیز خوب ارزیابی شد. هیچ موردی برای حذف یا بازبینی ارزیابی نشده است. پس از تکمیل فرایند اعتبار سنجی محتوا، مطالعه آزمایشی انجام شد.

این فرایند در بخش زیر توضیح داده شده است. با تکمیل اعتبار محتوا، یک مطالعه آزمایشی انجام شد تا مشخص شود آیا این ابزار برای پاسخ دهنده واضح است یا خیر.

مطالعه آزمایشی

مطالعه آزمایشی یک فرایند تحقیقاتی است که روش‌های تحقیق انتخاب شده را در نمونه کوچکتر از پیش آزمایش می‌کند. انجام یک مطالعه آزمایشی یک مرحله مهم است زیرا، زمانی که محقق یک روش آزمایشی از روش انتخاب شده را انجام می‌دهد، محقق از هر گونه نقص آگاه می‌شود و می‌تواند تغییرات مناسب را ایجاد کند. یک پرسشنامه مطالعه آزمایشی پس از تأیید اعتبار محتوا تهیه شد. این پرسشنامه با افرادی که در توسعه نرم‌افزار مشارکت داشتند به صورت آزمایشی اجرا شد. پرسشنامه آنلاین و سوالات مصاحبه با شرکت کنندگان در مطالعه آزمایشی مورد آزمایش قرار گرفت تا آیا سوالاتی که مورد بررسی قرار گرفته است واضح، قابل فهم یا مبهم بودند یا خیر. هر سوالی که باعث سردرگمی شود برای مطالعه اصلی بازنویسی شد. مطالعات آزمایشی به قضاوت در مورد مدت زمان انجام مصاحبه و همچنین برآورد زمان مورد نیاز برای تکمیل پرسشنامه توسط شرکت کنندگان کمک می‌کند.

یافته‌ها

ده شرکت کننده دارای طیف وسیعی از تجربه، از یک سال تا ۱۷ سال بودند. میانگین تجربه شرکت کنندگان آزمایشی ۱۳ سال بود. شرکت کنندگان در نقش‌های مختلفی از جمله: اسکرام مستر، توسعه دهنده، آزمایش کننده، تحلیلگر کسب و کار به کار گرفته شدند.

جدول زیر تعداد شرکت کنندگان بر اساس عنوان شغل را نشان می دهد.

عنوان شغلی	تعداد شرکت کنندگان
مدیر عملیات	۱
مدیر توسعه نرم افزار	۱
رهبر تیم توسعه	۱
تحلیل گر کسب و کار	۱
رهبر طراحی نرم افزار	۱
برنامه نویس	۳
رهبر فنی پروژه	۱
مدیران راه حل پروژه و کسب و کار	۱

جدول ۳-۲ پاسخ دهندگان مطالعه آزمایشی

از بین رویکردهای چابک که از شرکت کنندگان درخواست شد ، بیشترین استفاده از Scrum (۵۰) بود.

Scrum همچنین در ترکیب با سایر روشها استفاده شد. جدول ۳-۳ ، رویکرد مورد استفاده در پروژه را نشان می دهد. برخی از روشها در ترکیب با سایر روشها مورد استفاده قرار گرفت.

روش های مختلف مورد استفاده	مجموع	درصد(٪)
Scrum	۵	۵۰٪
Scrum and eXtreme programming (XP)	۱	۱۰٪
Scrum and Feature Driven Development (FDD)	۲	۲۰٪
Scrum and Kanban	۲	۲۰٪
Scrum and Lean software development	۱	۱۰٪
Scrum and Scrum-ban	۲	۲۰٪
Test Driven Development (TDD)	۳	۳۰٪

Behaviour Driven Development (BDD)	۲	۲۰٪
Scrum and Waterfall development	۱	۱۰٪
Waterfall	۱	۱۰٪
Prototyping development	۱	۲۰٪
Scrum and Prototyping development	۱	۱۰٪
Code and fix	۱	۲۰٪

جدول ۳-۳ روش‌های مطالعه آزمایشی استفاده شده

این پرسشنامه شامل ۳۸ سوال بود. به طور متوسط، شرکت کنندگان به ۸۵ درصد از سوالات پاسخ دادند. برخی از سوالات همانطور که انتظار می‌رفت پاسخ داده نشد، این ممکن است به این دلیل باشد که سوالها یا مربوط به نقش آنها نیست یا روشن نیست. سوالی در پایان پرسشنامه گنجانده شد که از شرکت کننده می‌پرسید آیا به نظر آنها پاسخ به سوالها آسان است و مربوط به نقش آنهاست. هشتاد درصد شرکت کنندگان پاسخ دادند که پرسشنامه قابل درک است. بیست درصد از شرکت کنندگان نظرات ارزشمندی در مورد ارتباط سوالها با توجه به نقش آنها ارائه کردند. بازخورد، هنگام ایجاد ابزار برای مطالعه با تیم‌های توسعه نرم‌افزار و همچنین مطالعه صاحب محصول، در نظر گرفته شد.

نتایج مطالعه آزمایشی

بازخورد از مطالعه آزمایشی در تجدید ساختار پرسشنامه آنلاین و سوالات مصاحبه مفید بود همچنین برای تعیین اینکه آیا ابزار قادر است اطلاعات مورد نیاز برای پرداختن به سوال تحقیق را بدست آورد، مورد استفاده قرار گرفت. مطالعه آزمایشی همچنین به برآورد زمان مورد نیاز برای تکمیل هر بخش از نظرسنجی کمک کرد.

انتخاب نمونه و انجام مصاحبه‌ها

مصاحبه حضوری با سوالات نیمه ساخت یافته با زیرمجموعه‌ای از شرکت کنندگان انجام شد که پس از تکمیل پرسشنامه و موافقت با ارائه اطلاعات عمیق‌تر در مورد نقش خود در فرایند توسعه نرم‌افزار، به طور تصادفی انتخاب شدند. شرکت کنندگان بر اساس فعالیت شرکتی که در آن کار می‌کردند به سه گروه مختلف (سفارشی/قراردادی، داخلی، توسعه محصول) طبقه بندی شدند. شرکت کنندگان از هر گروه به صورت تصادفی انتخاب شدند. به شرکت کنندگان منتخب یک ایمیل دعوت نامه برای شرکت در مصاحبه ارسال شد. مصاحبه‌هایی با پاسخ دهندگان Auckland, Wellington, Christchurch انجام شد.

برای این مطالعه و مطالعه آزمایشی، جمع‌آوری اطلاعات شامل جلب اطلاعات افراد در مورد جنبه‌های شغلی آنها بود. در تحقیقات کمی و کیفی یک انتقاد این است که ماهیت تحقیق امکان تماس نزدیک با مردم را فراهم می‌کند و این ممکن است منجر به مشکلات اخلاقی شود. پرسشنامه و مصاحبه‌ای که برای این تحقیق استفاده شد، برای افراد بود. بنابراین، ملاحظات اخلاقی در مراحل اولیه تحقیق مورد توجه قرار گرفت. ملاحظات اخلاقی نیز برای جلوگیری از مزاحمت برای شرکت کنندگان و اجتناب از هر شرایطی که ممکن است شرکت کننده در تحقیق را در معرض خطر قرار دهد، در نظر گرفته شد. به عنوان مثال، داده‌ها فقط برای اهداف تحقیق مورد استفاده قرار می‌گیرند، هویت فردی در نتایج یا هر نشریه ذکر نمی‌شود. بر اساس سیاست اخلاق انسانی دانشگاه لینکلن، کلیه تحقیقات مربوط به افراد قبل از شروع تحقیق نیاز به تأیید کمیته اخلاق انسانی دانشگاه لینکلن دارد. در طی این فرایند، یک فرم درخواست، یک کپی از پرسشنامه، یک کپی از سوالات مصاحبه، برگه اطلاعات تحقیق و ایمیل دعوت نامه‌ای که برای جذب شرکت کنندگان استفاده می‌شد، قبل از شروع جمع‌آوری داده‌ها، برای اخذ مجوز ارسال شد.

برنامه‌های جداگانه برای موارد زیر ساخته شده است:

- اعتبار سنجی محتوا و مطالعه آزمایشی
- مطالعه شامل تیم‌های توسعه نرم‌افزار و تأیید برای ادامه تحقیقات از LUHEC برای مطالعه آزمایشی در ۲۵ ژوئن ۲۰۱۴ دریافت شد.
- تأیید پروژه: تأثیر رابطه بین مشتری و تیم توسعه نرم‌افزار در نتیجه پروژه‌های توسعه نرم‌افزار در ۲۷ مارس ۲۰۱۵ ارائه شد.

مطالعه صاحب محصول

هدف اصلی این تحقیق درک رابطه بین تیم توسعه نرم‌افزار و صاحب محصول بود. بنابراین، یک مطالعه جداگانه با صاحبان محصول علاوه بر مطالعه تیم توسعه نرم‌افزار انجام شد. یک ابزار تحقیقاتی برای مطالعه صاحب محصول قبل از اداره کردن مجموعه داده‌ها که حاوی لیستی از سوالات بوده است که بر روی پرسش‌های تحقیق متمرکز بود، طراحی شد. هنگام نوشتن سوالات تحقیق، موضوع تحقیق، مرور یافته‌ها و چارچوب نظری مورد توجه قرار گرفت. پرسشنامه اولیه شامل ۱۵ سال بود.

این سوالات شامل مواردی بود که به صاحب محصول مربوط می‌شد:

- نوع سازمان
- اطلاعات پروژه
- مشکلات پیش روی پروژه

اعتبارسنجی ابزار مطالعه صاحب محصول

هر دو ابزار برای تیم‌های توسعه نرم‌افزار و مطالعه مالک محصول با استفاده از اعتبار سنجی محتوا، که قبلاً توضیح داده شده است، معتبر هستند. پس از تأیید محتوا، برخی از موارد مربوط به مطالعه صاحبان محصولات نیز با توجه

به بازخورد ارائه شده توسط کارشناسان اصلاح شد. مطالعه آزمایشی پس از تکمیل اعتبارسنجی محتوا انجام شد. پرسشنامه برای صاحبان محصول و تیم‌های توسعه نرم‌افزار با شرکت‌کنندگان در مطالعه آزمایشی برای ارزیابی اینکه آیا سوالات واضح، قابل فهم یا مبهم هستند مورد آزمایش قرار گرفتند. هر سوالی که باعث سردرگمی می‌شد، بازنویسی شد. سوالات مصاحبه برای صاحب محصول برای به دست آوردن اطلاعات با جزئیات نوشته شده است. برنامه‌های جداگانه‌ای برای تأیید اخلاق انسانی برای مطالعه‌ای که صاحبان محصول را شامل می‌شود، ساخته شد. تأییدیه برای این مطالعه که شامل نمایندگان مشتری است در ۲۷ اکتبر ۲۰۱۵ داده شد.

انتخاب نمونه و اجرای پرسشنامه آنلاین

تعداد کل شرکت‌های توسعه نرم‌افزار در نیوزلند ۱۱۲۰ تخمین زده می‌شود. این عدد از جمعیت‌شناسی کسب و کار در گزارش بخش فناوری اطلاعات که از Statistics New Zealand³، گزارش فناوری و نوآوری Canterbury از Flux4 و موسسه ملی نوآوری ICT5 بدست آمده است، محاسبه شده است. تعداد شرکت‌کنندگان در این مطالعه ۲۸۵ نفر انتخاب شد. این عدد از محاسبه حجم نمونه از جمعیت شرکت‌های توسعه نرم‌افزار در نیوزلند با استفاده از فرمول اندازه نمونه Cochran برای داده‌های دسته‌ای بدست آمده است:

$$n = (t)^2 * (p)(q) / (d)^2$$

که در آن t در ۹۵٪ عدد 1.96 هست

سطح Alpha یا سطح ریسک عدد 0.05 است

$$(p)(q) = \text{برآورد واریانس} = 0.25$$

حاشیه خطای قابل قبول $(d) = 0.25$

$$n_0 = \frac{(1.96)^2 * (0.5)(0.5)}{(0.05)^2}$$

بنابراین، برای جمعیت ۱۱۲۰ نفر، حجم نمونه مورد نیاز ۳۸۴ است. با این حال، از آنجا که این حجم نمونه بیش از ۵٪ از جمعیت (۵۶۲ = $0.5 * 1120$) بود، از فرمول تصحیح Cochran (۱۹۷۷) برای محاسبه حجم نمونه نهایی استفاده شد.

این محاسبات به شرح زیر است:

$$\begin{aligned} N1 &= n_0 / (1 + \frac{n_0}{\text{Population}}) \\ &= 384 / (1 + 384/1120) \\ &= 384 (1 + 0.343) \\ &= 384 / 1.343 \\ &= 285 \end{aligned}$$

شرکت‌کنندگان برای تشکیل نمونه‌ای از سازمان‌های توسعه نرم‌افزار در سراسر نیوزلند به کار گرفته شدند. شرکت‌کنندگان با استفاده از یکی از این موارد استخدام شدند: رویکردها:

- از طریق ایمیل،
- درخواست ارسال دعوتنامه به مردم،
- آدرس ایمیل به دست آمده از کارت ویزیت،
- از طریق شبکه حرفه ای Agile،
- از طریق Test Professional Network 6
- آدرس ایمیل تماس یک شرکت.

برای جمع‌آوری داده‌های افرادی که در حال حاضر در توسعه نرم‌افزار نقش دارند، از پرسشنامه آنلاین استفاده شد. شرکت‌کنندگان بر اساس تکنیک‌های نمونه‌گیری غیر احتمالی نمونه‌گیری هدفمند و "گلوله برفی"^۱ انتخاب شدند. این تکنیک‌ها برای انتخاب شرکت‌کنندگان از مناطق تحقیق به دلیل ماهیت تحقیق انتخاب شدند. نمونه‌گیری هدفمند، که به عنوان نمونه‌گیری قضاوتی یا انتخابی نیز شناخته می‌شود، نوعی نمونه‌گیری غیر احتمالی است. هدف اصلی این نوع نمونه‌گیری غیر احتمالی الزام تکیه بر قضاوت محقق است. رمز نمونه‌گیری هدفمند این نیست که به طور تصادفی یک نمونه را از یک جمعیت به منظور عمومی سازی در مورد آن نمونه انتخاب کنید بلکه برای یک هدف خاص انتخاب کنید. نمونه‌گیری هدفمند متکی بر شرکت‌کنندگانی است که قادر به ارائه اطلاعات غنی در مورد حوزه تحقیق هستند. بنابراین، روش نمونه‌گیری هدفمند برای این مطالعه انتخاب شد زیرا امکان انتخاب نمونه بر اساس هدف مطالعه - تحقیق را فراهم می‌آورد. علاوه بر نمونه‌گیری هدفمند، از عنصر نمونه‌گیری گلوله برفی نیز استفاده شد. نمونه‌گیری گلوله برفی یک روش نمونه‌گیری غیراحتمالی است که در آن داده‌ها از پاسخ دهندگان شناخته شده جمع‌آوری می‌شود و به نوبه خود از آنها خواسته می‌شود تا اطلاعاتی را برای یافتن سایر پاسخ دهندگان ارائه دهند.

در این مطالعه از نمونه‌گیری گلوله برفی برای دستیابی به جمعیت پنهان افرادی که در توسعه نرم‌افزار چابک دخیل هستند استفاده شد. جمعیت صاحبان محصول که در توسعه نرم‌افزار Agile کار می‌کنند پنهان بود. تعداد شرکت‌کنندگان مورد نیاز برای این مطالعه ناشناخته بود و برآورد می‌شود که بیشتر از تعداد سازمانهایی باشد که از رویکرد توسعه نرم‌افزار Agile پیروی می‌کردند، که بیش از ۱۱۲۰ نفر تخمین زده می‌شد. در صورت امکان، نمونه‌گیری تصادفی در انجام تحقیقات توصیه می‌شود، زیرا تصادفی تعصبات را کاهش می‌دهد و نتایج را می‌توان تعمیم داد. با این حال، نمونه‌گیری تصادفی ممکن است همیشه امکان پذیر نباشد. این ممکن است به این دلیل

¹ Snowball

رخ دهد که همه افراد حاضر به مشارکت نیستند. روش نمونه‌گیری هدفمند با وجود سوگیری در نمونه‌گیری، همچنان می‌تواند داده‌های معتبر و قوی مورد نیاز برای تحقیقات را ارائه دهد. برای این مطالعه، شرکت‌کنندگان از چندین سازمان توسعه نرم‌افزار مختلف انتخاب شدند. معیار انتخاب پاسخ‌دهندگان افرادی بودند که در تیم‌های توسعه نرم‌افزار Agile شرکت داشتند. شرکت‌کنندگانی که در سازمانهای توسعه نرم‌افزار کار می‌کردند اما در حال حاضر در توسعه نرم‌افزار مشارکت نداشتند در این تحقیق گنجانده نشدند. یک پرسش در پرسشنامه آنلاین برای انتخاب شرکت‌کنندگانی که در حال حاضر در توسعه نرم‌افزار Agile کار می‌کنند، استفاده شد. این مورد در ایمیل نیازمندی در مورد نقش‌هایی که در مطالعه گنجانده می‌شوند و افرادی که باید برای شرکت در آن شرکت کنند، مشخص شده است.

۴-۱ ساختار گزارش سمینار

فصل اول : در فصل اول به سوال‌های:

۱- چگونه موضوع و مساله را تعریف و تبیین کرده است؟

۲- سوالات تحقیق و یا فرضیه‌ها چه هستند؟

۳- چه روش پژوهشی را استفاده کرده است؟

۴- مراحل تحقیق چه بوده‌اند

فصل دوم : در این فصل به سوال پنجم که عبارتست از: داده‌های تحقیق را چگونه بدست آورده است؟ جواب خواهیم داد.

فصل سوم: در این فصل به سوال‌های

۶- نتایج حاصل چه بوده‌اند

۷- چگونه نتایج ارزیابی شده‌اند

۸- نتیجه‌گیری و جمع‌بندی پژوهش انجام شده چه بوده

۹- چه پیشنهاداتی را برای ادامه کار ارائه نموده است

در فصل منابع تمامی منابع استفاده شده در این تحقیق را نام خواهیم برد

فصل دوم

مفاهیم عمومی و

روش‌های بدست‌آوری داده‌ها

مقدمه

یکی از مهم‌ترین مراحل در روش تحقیق، جمع‌آوری داده‌ها یا Data Collecting است بطوری که نویسندگان هر مقاله باید برای رسیدن به اهداف و بررسی فرضیات و سوالات خود داده‌ها را جمع‌آوری کرده و سپس به آنالیز و تحلیل آن‌ها بپردازند. در این قسمت قصد داریم بطور مفصل به بررسی این موضوع بپردازیم.

روش بدست آوردن داده‌های تحقیق

جمع‌آوری داده‌ها از شرکت‌کنندگان به دو شیوه انجام شده است:

۱. پرسشنامه آنلاین (داده‌های کمی و داده‌های کیفی)

۲ مصاحبه نیمه ساخت‌یافته (داده‌های کیفی)

پرسشنامه این مطالعه شامل سوالاتی از جمله نقش پاسخ دهندگان، مدت زمان کار آنها در توسعه نرم‌افزار و اندازه سازمان بود. این پرسشنامه همچنین شامل سوالاتی برای جمع‌آوری اطلاعات در مورد شیوه‌های چابک مورد استفاده در پروژه‌های توسعه نرم‌افزار بود. به منظور کسب درک بیشتر از تجربیات شرکت‌کنندگان در محیط تیم چابک، از پیگیری مصاحبه استفاده شد. مصاحبه‌های نیمه ساخت‌یافته روش مناسبی برای به دست آوردن درک عمیق از پروژه‌های چابک در نظر گرفته شد.

درک عمیق از کار شرکت‌کنندگان در تیم‌های چابک از الزامات این مطالعه است. مصاحبه‌های نیمه ساخت‌یافته به مصاحبه کننده این فرصت را می‌دهد که سوالات را بر اساس مفاد مصاحبه، از جمله تجربیات پاسخ دهنده، که به سهولت ارتباط کمک می‌کند، ساده کند. مصاحبه‌های نیمه ساخت‌یافته به مصاحبه‌کننده اجازه می‌دهد در صورتی که پاسخ داده شده به اندازه کافی به جواب سوالها را مشخص نکند، موضوعی را بیشتر مورد بررسی قرار دهد. بنابراین، مصاحبه نیمه ساخت‌یافته بر مصاحبه ساختارمند ترجیح داده شد. برای کشف اطلاعات از روش‌های کمی استفاده می‌شود و از روش کیفی برای توضیح دقیق چالش‌ها استفاده می‌شود. در گردآوری داده‌های کمی، داده‌هایی که جمع‌آوری شدند، می‌توانند به صورت عددی بیان شوند. از آنجا که داده‌ها به صورت عددی هستند، می‌توان آزمون‌های آماری توصیفی و استنباطی را بر روی داده‌ها انجام داد. آمار توصیفی از میانگین، میانه و انحراف استاندارد برای توصیف آنچه داده‌ها نشان می‌دهد، استفاده می‌کند. از آمار استنباطی، مانند ضریب

همبستگی محصول-لحظه پیرسون، برای بررسی رابطه بین مجموعه‌های مختلف داده‌ها، بر اساس همزمانی داده‌ها استفاده می‌شود.

داده‌های کیفی جزئیات مربوط به رفتار یا ویژگی‌های شخصی انسان را ارائه می‌دهد و شامل اطلاعاتی در مورد تجربیات، نیازها، روال، خواسته‌های شرکت‌کنندگان، موارد استفاده و سایر اطلاعات مفصل که برای تحقیقات ضروری است می‌باشد که مطالعات کمی نمی‌توانند ارائه دهند. نمونه‌هایی از داده‌های کیفی شامل پرسشنامه‌های باز، مصاحبه‌های بدون ساختار و مشاهدات بدون ساختار است. از پرسشنامه به عنوان ابزار تحقیق استفاده می‌شود که در آن مجموعه‌ای از سوالات به منظور جمع‌آوری اطلاعات از پاسخ‌دهندگان مطرح می‌شود. مصاحبه گفتگویی بین مصاحبه کننده و پاسخ‌دهندگان است که برای جمع‌آوری اطلاعات از پاسخ‌دهندگان استفاده می‌شود.

یک روش کیفی برای به دست آوردن نظرات و بینش دقیق‌تر از زیرمجموعه افرادی که پرسشنامه را تکمیل کرده بودند، استفاده شد. در این مطالعه، داده‌های کیفی به تو ضیح بیشتر چالش‌های مختلف مردم در طول توسعه نرم‌افزار کمک کرد. یک مطالعه کیفی همچنین برای تعریف، نظریه‌پردازی، توضیح، کاوش و ترسیم تحلیل‌های اساسی برای درک روابط ضروری است. داده‌های کیفی در این مطالعه از طریق مصاحبه‌های نیمه ساختار یافته جمع‌آوری شد. بنابراین، برای دستیابی به اهداف این مطالعه، یک روش ترکیبی انتخاب شده است.

جمع‌آوری داده‌ها از مطالعه آزمایشی

هدف از مطالعه آزمایشی، آزمایش ابزارهای توسعه یافته بود. فرآیند اتخاذ شده برای انجام مطالعه آزمایشی در زیر مورد بحث قرار گرفته است.

مشارکت‌کنندگان بالقوه از سازمان‌هایی که فعالیت اصلی آنها توسعه نرم‌افزار بود یا به عنوان بخشی از تجارت خود، نرم‌افزاری را برای استفاده داخلی تهیه کردند، مشخص شدند. این سازمانها در سه دسته قراردادی/سفارشی، توسعه داخلی و توسعه محصول طبقه‌بندی شدند. سازمانهای توسعه نرم‌افزاری قراردادی/سفارشی نرم‌افزاری را برای یک سازمان یا کاربر خاص توسعه می‌دهند.

در حالی که سازمان‌های توسعه داخلی از تیم‌های داخلی برای توسعه نرم‌افزار برای استفاده شرکت استفاده می‌کنند، شرکت‌های توسعه محصول، محصولات یا خدمات جدیدی را برای مشتری طراحی، ایجاد و به بازار عرضه می‌کنند.

برای جمع آوری داده‌ها از شرکت کنندگان از روش ترکیبی استفاده شد. یک پرسشنامه و سوالات مصاحبه تهیه شد. داده‌های کمی از طریق پرسشنامه آنلاین و داده‌های کیفی از مصاحبه‌های نیمه ساخت‌یافته جمع‌آوری شد. شرکت کنندگانی که آدرس ایمیل آنها از چندین منبع مختلف، از جمله شبکه‌های حرفه‌ای که متخصصان نرم‌افزار به آن تعلق دارند، به دست آمده است. هنگامی که ارتباطات برقرار شد، در دسته بندی قراردادهای/سفارشی، توسعه داخلی و توسعه محصول طبقه بندی شد. انتخاب تصادفی از هر گروه انجام شد و یک ایمیل دعوت برای شرکت ارسال شد. اگر پاسخی در مدت دو هفته دریافت نشد، دعوت نامه یادآوری ارسال می شود. این روند تا تکمیل ده پرسشنامه ادامه یافت. در کل، ۳۰ ایمیل ارسال شد و ۱۵ نفر پاسخ دادند. از بین افرادی که پاسخ دادند پنج نفر معیارهای انتخاب را ندارند، بنابراین از مطالعه حذف شدند. آنها معیارها را برآورده نمی کردند زیرا مستقیماً در توسعه نرم‌افزار مشارکت نداشتند، حتی اگر از یک سازمان توسعه نرم‌افزار بودند. میزان پاسخ برای مطالعه آزمایشی ۳۳ درصد بود. داده‌های مطالعه آزمایشی در یک دوره دو ماهه جمع‌آوری شد. برای ارائه پرسشنامه آنلاین از بستر Qualtrics استفاده شد. داده‌های پرسشنامه مستقیماً از Qualtrics به Microsoft Excel وارد شد. مصاحبه‌های ضبط شده به صورت کلمه به کلمه در مایکروسافت ورد بازنویسی شد. در سیستم فایل شبکه‌ای دانشگاه که به محقق اختصاص داده شده ذخیره می شود.

پرسشنامه آزمایشی شامل سوالاتی که وابسته به آمار نفوس و تجربیات آنها بود تا میزان مشکلاتی را که مشارکت کنندگان را در تیم توسعه یا نماینده مشتری تجربه کرده باشد را کشف کند. سه شرکت کننده از شرکت‌های توسعه دهنده قرارداد / توسعه سفارشی و شرکت‌های توسعه داخلی و چهار شرکت کننده از شرکت‌های توسعه محصول حضور داشتند. با تمام شرکت کنندگانی که پرسشنامه را تکمیل کردند، شخصاً تماس گرفته شد تا در یک مصاحبه نیمه ساخت‌یافته شرکت کنند. زمان و مکان، رضایت آنها از ضبط صدا و مدت زمان مورد انتظار مصاحبه نیز در یک ایمیل ارسال شده در طول تماس اولیه مشخص شد. یکی از پاسخگویان آزمایشی به دلیل تعهدات دیگر نتوانست مصاحبه را در بازه زمانی لازم تکمیل کند. نه مصاحبه با شرکت کنندگان همراه با ضبط صدای مصاحبه شونده انجام شد. مصاحبه‌ها تایپ شده و از روش موضوعی برای تجزیه و تحلیل داده‌ها استفاده شد. داده‌های بدست آمده از مصاحبه‌ها به بازنگری و اصلاح سوالات مصاحبه به منظور وضوح و اطمینان از اینکه اطلاعات مورد نیاز به دست آمده است، کمک کرد.

جمع آوری داده‌ها از تیم‌های توسعه نرم‌افزار

داده‌های اولیه با استفاده از پرسشنامه آنلاین از افرادی که در حال حاضر در توسعه نرم‌افزار در سراسر نیوزلند دخیل هستند جمع آوری شد. پرسشنامه‌ها برای مطالعه تیم توسعه نرم‌افزار به طور متوسط در مدت ۱۸ دقیقه با محدوده ۱۳ تا ۲۶ دقیقه تکمیل کردند. پاسخ‌ها طی یک دوره شش ماهه بین ماه اوت ۲۰۱۵ و ژانویه ۲۰۱۶ جمع آوری شد. مصاحبه‌های نیمه ساخت‌یافته با نمونه تصادفی طبقه‌ای از پاسخ دهندگانی که پرسشنامه آنلاین را تکمیل کرده بودند، انجام شد تا درک عمیق‌تری از روابط بین مشتری و تیم توسعه نرم‌افزار چابک به دست آید.

جمع آوری اطلاعات از نمایندگان مشتری/صاحبان محصول

داده‌های اولیه با استفاده از پرسشنامه آنلاین از افرادی که در حال حاضر به عنوان نماینده مشتری/صاحب محصول در توسعه نرم‌افزار در سراسر نیوزلند نقش دارند، جمع آوری شد. مصاحبه‌های نیمه ساخت‌یافته با افرادی که پرسشنامه آنلاین را تکمیل کرده بودند انجام شد. شرکت کنندگانی که در سازمان‌های مختلف در سراسر نیوزلند استخدام شده بودند از طریق ایمیلی که برای گروه‌های مختلف توسعه نرم‌افزار، مانند شبکه حرفه ای Agile ارسال شده بود، با آنها تماس گرفته شد. شبکه حرفه ای Agile گروهی از افرادی است که در توسعه نرم‌افزار با استفاده از رویکرد چابک کار می‌کنند یا به روش چابک علاقه‌مند بودند و در Christchurch مستقر بودند. Agile Auckland گروهی از افرادی است که در توسعه نرم‌افزار با استفاده از رویکرد Agile کار می‌کنند یا علاقه‌مند به آن رویکرد بودند و در اوکلند مستقر بودند و 8 Agile Welly یک گروه چابک حرفه‌ای مستقر در ولینگتون است. با برگزارکنندگان رویداد هر یک از این گروه‌ها تماس گرفته شد و از آنها خواسته شد تا ایمیل دعوت نامه را برای اعضای خود ارسال کنند. از برگزارکنندگان نیز خواسته شد که در رویدادهای خود مدتی به محقق فرصت دهند تا در مورد تحقیقاتش صحبت کند و شرکت کنندگان را معرفی کند.

مصاحبه حضوری با همه شرکت کنندگان که پرسشنامه را تکمیل کردند انجام شد. نمونه‌گیری هدفمند، همانطور که قبلاً توضیح داده شد، برای جذب شرکت کنندگان استفاده شد. تعیین اندازه جمعیت نمایندگان مشتری/صاحبان محصول دشوار بود. در نیوزلند، تقریباً ۱۱۲۰ سازمان وجود دارد که فعالیت اصلی آنها توسعه نرم‌افزار است یا در توسعه پروژه‌های نرم‌افزاری مشارکت دارند. در سازمانی که از شیوه‌های چابک استفاده می‌کنند، تیم‌هایی تشکیل شد که حداقل یکی از آنها نقش نماینده مشتری را بر عهده داشت. ممکن است بیش از یک تیم در یک سازمان وجود داشته باشد یا نماینده مشتری ممکن است با بیش از یک تیم کار کند. در این مورد، ایجاد تعدادی پروژه که به صاحب محصول نماینده مشتری نیاز داشته باشد دشوار است. بنابراین، به دلیل فقدان داده در مورد تعداد پروژه‌های انجام شده، نمی‌توان جمعیت کل صاحبان محصول/نمایندگان مشتری را تعیین کرد. با این حال، هدف از پاسخ دهندگان برای مطالعه صاحب محصول این بود که شامل سازمان‌های مختلف از جمله سازمان‌های سفارشی، داخلی و توسعه محصول می‌شود.

تعیین اندازه جمعیت نمایندگان مشتری/صاحبان محصول دشوار بود. در نیوزلند، تقریباً ۱۱۲۰ سازمان وجود دارد که فعالیت اصلی آنها توسعه نرم افزار است یا در توسعه پروژه‌های نرم افزاری مشارکت دارند. در سازمانی که از شیوه‌های چابک استفاده می‌کند، تیم‌هایی تشکیل شد که حداقل یکی از آنها نقش نماینده مشتری را بر عهده داشت. ممکن است بیش از یک تیم در یک سازمان وجود داشته باشد یا نماینده مشتری ممکن است با بیش از یک تیم کار کند. در این مورد، ایجاد تعدادی از پروژه‌هایی که نیاز به نماینده مشتری (صاحب محصول) دارند دشوار است. بنابراین، کل جمعیت صاحبان محصول/مشتری به دلیل کمبود داده‌ها در مورد تعداد پروژه‌های انجام شده، نمی‌توان نمایندگان را تعیین کرد. با این حال، هدف از پاسخ دهندگان برای مطالعه صاحب محصول این بود که شامل سازمان‌های مختلف از جمله سازمان‌های سفارشی، داخلی و توسعه محصول می‌شود.

خلاصه

در این فصل، روش‌های تحقیق با نشان دادن درک تحقیقات کمی و کیفی مورد بحث قرار گرفته است. این فصل همچنین نشان داد که چرا روش ترکیبی برای این تحقیق ترجیح داده شد. روشها و ابزارهای مورد استفاده با توجیهی برای انتخاب‌های انجام شده برای این تحقیق مورد بحث قرار گرفت. در این تحقیق دو دسته مطالعه مختلف انجام شد، یکی با تیم‌های توسعه نرم افزار و دیگری با صاحبان محصول. در قسمت‌های جلوتر، نتایج حاصل از داده‌های کمی را ارائه می‌شود و شامل نتایج و بحث از داده‌های کیفی برای هر دو مطالعه است.

فصل سوم

جمع‌بندی و پیشنهادها

مقدمه

مانند هر روش دیگر، روش چابک نیز برای همه پروژه‌ها مناسب نیست و بهتر است شما برای هر موقعیت و هر پروژه متدولوژی مناسب آن را انتخاب کنید. هر چند روش چابک به توسعه دهندگان و تیم‌های پروژه در برآوردن نیازهای مشتری کمک می‌کند، اما همیشه هم برای دریافت نتیجه مناسب و به پایان رساندن پروژه، مناسب نیست. محدودیت‌های Agile thinking :

- در صورتی که مشتری اهداف و انتظارات خود را به شکل صریح، واضح و شفاف بیان نکند، استفاده از متد چابک چندان موفق نخواهد بود.
- در صورتی که مدیران پروژه یا تیم پروژه بی تجربه باشند و یا به دلیل تحت فشار بودن نتوانند درست کار کنند، نیز این روش کارایی چندانی نخواهد داشت.
- در سازمان‌های سنتی، معمولاً روش‌هایی مانند روش چابک با عدم پذیرش از سوی کارکنان روبرو می‌شود. چرا که این سازمان‌ها مدت‌ها است پروژه‌ها را با طرح‌ها و برنامه‌های سابق خود انجام می‌دهند و پذیرش تغییر این روند با مقاومت از سوی آنان همراه است.
- روش Agile برای مشتریانی که به فرایندهای سازمان خود به سختی پایبند هستند و یا روش‌های عملیاتی خود را دارند نیز چندان مناسب نخواهد بود.

نتایج حاصل چه بوده‌اند و چگونه این نتایج ارزیابی شده‌اند

در این قسمت به یافته‌های در خصوص جواب سوال‌های تحقیق می‌پردازیم.

۱- آثار و یافته‌ها حاکی از آن است که بین تیم توسعه و نماینده مشتری در پروژه‌هایی که به سبک Agile

هستند، چالشی وجود دارد. این چالش‌ها و مباحث تا چه حدی هستند؟

داده‌های ارائه شده در بخش ۴، ۵، ۳، ۲ و چالش‌های ارتباطی، اولویت‌های پروژه، چارچوب زمانی پروژه‌ها و الزامات پروژه و همچنین چالش‌های بین فردی بین اعضای تیم را مشخص کرد. این بدان معناست که در پروژه‌های توسعه نرم‌افزار Agile چالش‌هایی وجود دارد. چنین چالش‌هایی توسط اکثریت (۷۳٪) پاسخ دهندگان تیم توسعه نرم‌افزار و شصت و هفت درصد از دارندگان محصول گزارش شده است.

هم تیم توسعه نرم افزار و هم صاحبان محصول معتقد بودند که اینها می توانند چالش های متداول در توسعه نرم افزار چابک باشند.

اطلاعات به دست آمده از مطالعه توسعه نرم افزار و مطالعه صاحب محصول نشان می دهد که ۵۳ درصد از پاسخ دهندگان حداقل با یک تا سه چالش در مورد اولویت های پروژه، الزامات پروژه، مدت زمان پروژه، چالش های بین فردی یا ارتباطات روبرو هستند. این یافته ها نشان می دهد که چالش های موجود در توسعه نرم افزار چابک قابل مشاهده است و می تواند بر نتیجه پروژه تأثیر بگذارد. مفاهیم این چالش های قابل مشاهده این بود که برخی از این چالش ها بر نتایج پروژه تأثیر داشتند. این امر با محاسبه همبستگی بین چالش ها، مانند ارتباطات، چالش های بین فردی، اولویت های پروژه، به اشتراک گذاری ایده ها و زمان اتمام پروژه، که به طور مفصل در بخش ۴,۳ ارائه شده است، تأیید شد.

۲- چه نوع چالش هایی در توسعه نرم افزارها به سبک Agile وجود دارد؟

داده های جمع آوری شده از پرسشنامه آنلاین نشان داد که چالش هایی در تیم های توسعه نرم افزار و همچنین بین مالک محصول و تیم توسعه نرم افزار وجود دارد. پیامدهای چنین چالش هایی می تواند بر نتیجه پروژه تأثیر بگذارد. چنین چالش هایی در راستای چالش هایی بود که توسط Chow & Cao (۲۰۰۸) و Misra و همکاران مشخص شد. (۲۰۰۹). به عنوان بخشی از این مطالعه، علاوه بر پرسشنامه آنلاین، داده ها با استفاده از مصاحبه های نیمه ساخت یافته جمع آوری شده است. هدف از این جمع آوری اطلاعات، بررسی اطلاعات با جزئیات بیشتر از حد مجاز در قالب پرسشنامه بود. تجزیه و تحلیل داده های مصاحبه چالش هایی را نشان داد که قبلاً در مطالعات قبلی در زمینه توسعه نرم افزار Agile شناسایی نشده بود. چالش های اضافی شامل فرهنگ سازمانی بود، چیزی که یک تیم نمی تواند آن را کنترل کند.

چالش های شناسایی شده در این مطالعه عبارتند از:

- ارتباطات
- در دسترس بودن صاحب محصول
- انگیزه تیمی
- فرهنگ سازمانی
- چالش های بین فردی
- عدم توافق با مشتریان در مورد اولویت ها و الزامات پروژه

چالش های ارتباطی که توسط Hoda و همکاران (۲۰۱۱) گزارش شده است، به دلیل عدم دسترسی به صاحب محصول در طول توسعه پروژه بوده است. پیامدهای چنین چالش هایی مشکلاتی در جمع آوری و روشن شدن الزامات بود زیرا صاحب محصول حاضر نیست وقتی تیم نیاز به توضیح بیشتری در مورد الزامات داشت. این امر منجر به تأخیر در تصمیم گیری شد. در این مطالعه حاضر، پاسخ دهندگان گزارش دادند که در دسترس بودن صاحب محصول یک مشکل است و باعث تأخیر در ارتباط می شود. این یافته به طور عمیق مورد بحث قرار گرفت. Chow & Cao (۲۰۰۸) و Neur و همکاران (۲۰۰۵) گزارش دادند که محدوده پروژه به عنوان یکی از چالش های

اصلی در نرم افزار Agile تعریف شده است. این به این دلیل بود که روش‌های توسعه نرم‌افزار Agile بر اساس برنامه‌ریزی متکی بود که برخلاف توسعه سریع پروژه‌ها است. در این مطالعه، از پاسخ دهندگان پرسیده شد که چند بار در یک ارتباط معمولی با چالش خاصی در ارتباطات، اولویت‌های پروژه، الزامات پروژه و مسائل بین فردی روبرو شده‌اند. این اطلاعات به نشان دادن عمق چالش‌های رایج در توسعه نرم‌افزار Agile کمک کرد. اطلاعات به دست آمده از پاسخ دهندگان در مورد تعداد چالش‌های پیش روی در هر دو ارتباط در بخش ۴,۳ ارائه شده است. انگیزه تیم، در زمینه این مطالعه، به تمایل اعضای تیم توسعه برای تکمیل وظایف محوله اشاره کرد.

به طور مشابه، Chow & Cao (۲۰۰۸) یافته‌های مربوط به توسعه چابک را مورد بررسی قرار داد و انگیزه تیم را به عنوان یکی از عوامل موفقیت بالقوه برای پروژه‌های توسعه نرم‌افزار Agile پس از شناسایی این عوامل از منابع، آنها یک نظرسنجی مبتنی بر وب انجام دادند و بازخورد ۱۰۹ پروژه چابک از ۲۵ کشور مختلف را جمع‌آوری کردند. یافته‌های این نظرسنجی به آنها کمک کرد تا لیست عوامل مهم موفقیت خود را بیشتر اصلاح کنند. لیست نهایی شامل انگیزه تیم به عنوان یکی از عوامل موفقیت در توسعه نرم‌افزار چابک نبود. نتایج آنها بر اساس پرسشنامه‌ای بود که در آن از مقیاس هفت درجه ای Likert برای نشان دادن سطح درک پاسخ دهندگان در مورد عوامل موفقیت استفاده شد. مطالعه آنها نمی‌تواند انگیزه تیم را به عنوان یک عامل موفقیت در توسعه نرم‌افزار Agile تأیید کند. این می‌تواند به این دلیل باشد که پاسخ‌دهندگان ممکن است فکر کنند که روش‌های چابک خود باعث ایجاد انگیزه در تیم می‌شود.

Ceschi و همکاران، (۲۰۰۵) پرسشی را در پرسشنامه با عنوان "مهمترین ویژگی توسعه دهنده در توسعه نرم‌افزار چابک چیست؟" گنجانند. سی درصد از پاسخ‌دهندگان در مطالعه خود انگیزه تیم را به عنوان یک ویژگی مهم در توسعه نرم‌افزار Agile ذکر کردند. با این حال، در این مطالعه حاضر از پاسخ‌دهندگان پرسیده شد که چرا هنگام کار در رویکرد توسعه نرم‌افزار چابک انگیزه دارند. این سوال برای یافتن دلیل علاقه افراد به کار بر روی پروژه‌های توسعه نرم‌افزار Agile گنجانده شد. یافته‌های تیم توسعه نرم‌افزار از این مطالعه فعلی نشان می‌دهد، مانند یافته‌هایی که اولین بار توسط Chow & Cao (۲۰۰۸) انجام شد، انگیزه تیم یکی از عوامل موفقیت ضروری در توسعه نرم‌افزار چابک بود.

چالش‌های ناشی از عوامل بیرونی و فرهنگ سازمانی، چالش‌های جدیدی بود که در این مطالعه شناسایی شد و در مطالعه قبلی گزارش نشده بود. عوامل خارجی مشخص شده در این مطالعه شامل عدم تجربه مشتری از یک فرآیند چابک یا تکیه بر تعدادی از ذینفعان و پیمانکاران خارجی برای بخش‌های جدایی ناپذیر پروژه است. چنین چالش‌هایی می‌تواند در تکمیل به موقع کار تأثیر داشته باشد.

چالش‌های موجود در فرهنگ سازمانی شامل مالک محصول دارای نقش‌های متعدد بود. در این مطالعه دو نفر از پاسخ دهندگان گزارش کردند که برخی نقش‌ها در مالکیت محصول در برخی از سازمان‌ها وجود دارد. گاهی اوقات این چالش‌ها بیشتر از دیدگاه صاحب پروژه نسبت به نظر تیم توسعه بوده است. صاحب پروژه با نقش‌های متعدد و ناتوانی در اولویت‌بندی مناسب این نقش‌ها، دچار سردرگمی شد. این مورد تنها توسط دو پاسخ دهنده صاحب محصول گزارش شده است و این ممکن است برای همه پروژه‌ها مشترک باشد یا نباشد. رویکرد چابک از کار تیمی

پشتیبانی می‌کند. با این حال، مواردی در این مطالعه گزارش شده است که یکی از اعضا به دلیل مجموعه مهارت‌هایی که این شخص دارد مجبور به کار به صورت تنها می‌شود. این بدان معناست که فقط یک نفر دارای مهارت مورد نیاز برای آن پروژه در سازمان است. این نشان می‌دهد که در این مورد خاص، آنها از اصول چابک پیروی نمی‌کنند، زیرا Agile پیشنهاد کرد که تیم‌های چند منظوره در طول توسعه پروژه داشته باشند. سه سازمان وجود داشت که در آن تنها یک فرد در یک تیم وجود داشت. در هر یک از این سه مورد، پاسخ دهندگان اظهار کردند که از رویکرد چابک پیروی می‌کنند. این نشان می‌دهد که برخی از سازمان‌ها ممکن است باور داشته باشند که در حین اجرای

یک فرایند چابک هستند. با این حال، ممکن است در پیروی از این روش تنها با یک نفر از اعضای تیم مشکلاتی وجود داشته باشد. این نشان می‌دهد که ممکن است سازمان‌ها فکر کنند چابک هستند اما آنها فقط از یکی یا دو روش چابک در بخش‌هایی از فرآیند مورد نظر خود استفاده می‌کنند یا می‌توانند (به دلیل تعداد کارکنان) از آنها پیروی کنند. در مورد فرآیند چابک، هنگامی که به درستی دنبال نشود، با نتایجی نادرستی مواجه خواهیم بود.

۳- تاثیر این چالش‌ها بر روابط بین مشتری و تیم توسعه نرم‌افزار به سبک Agile چیست؟

پاسخ دهندگان تیم توسعه نرم‌افزار گفتند که آنها در مرحله جمع‌آوری الزامات یک پروژه "آزادی کاوش" را دارند. این ممکن است به این معنا باشد که آنها مجبور نبودند از روش چابک پیروی کنند تا کار را آنطور که باید انجام شود، انجام دهند. هنگامی که صاحب محصول فهرست اولویت‌های پروژه‌ها را تعیین می‌کند، تیم توسعه می‌تواند بین خود بحث و گفتگو کرده و سپس به راه حل برسد. این همچنین می‌تواند به این معنی باشد که صاحب محصول به تیمی که وظیفه‌ای که به او سپرده شده را به طور موثر انجام می‌دهد اعتماد داشته است. در مطالعه صاحب محصول، برخی از پروژه‌ها، که در آن چندین تیم با هم کار کردند، دارای تیم‌هایی بودند که از رویکردهای کاری مختلف استفاده می‌کردند. به عنوان مثال، ممکن است یکی از تیم‌ها در مورد توزیع کار صحبت کند و تیم دیگر صحبت نکند. این به عنوان یک چالش شناخته شده است، زیرا اعضای تیم روش‌های متفاوتی برای مقابله با مشکل و تکمیل کار داشتند. Dahlin و همکاران (۲۰۰۵) گزارش داد که اعضای تیم در تیم‌های مختلف ممکن است توانایی‌های متفاوتی برای شناسایی راه‌حل‌ها و استفاده از اطلاعات مربوطه داشته باشند. هنگامی که اعضای تیم‌های مختلف گرد هم می‌آیند، این تنوع را افزایش می‌دهد و سپس آنها به زمان نیاز دارند تا درک مشترکی را ایجاد کنند. چنین مشکلاتی ممکن است باعث تاخیر در انجام وظایف شود زیرا هر دو تیم برای درک نحوه کار هر تیم برای انجام یک کار به زمان نیاز داشتند. تفاهم بین اعضای تیم در مراحل بعدی ممکن است به هماهنگی بین اعضای تیم کمک کند. پاسخ دهندگان همچنین گزارش دادند که عدم دسترسی به صاحب محصول مشکلاتی را در ارتباطات ایجاد می‌کند و به همین دلیل، تصمیم‌گیری در مورد پروژه به تأخیر می‌افتد. Pikkarainen و

همکاران (۲۰۰۸)، در طی یک مطالعه موردی بر روی تیم های توسعه نرم افزار Agile، دریافتند که داشتن یک فضای باز در یک دفتر، جلسات روزانه و برنامه ریزی تکرار جلسات باعث بهبود ارتباطات می شود. آنها همچنین پیشنهاد کردند که ارتباط موثر در یک تیم مستلزم داشتن درک مشترک از پروژه ها است. ارتباطات در تیم نیز به مهارت های بین فردی اعضای تیم بستگی دارد. این مهارت ها شامل پیشنهادات مفید، گوش دادن فعال و شناخت علایق و دستاوردهای دیگر اعضای یک تیم بود. یکی از پاسخ دهندگان پروژه ای را بررسی کرد که در آن مدیریت در تیم به طور موثر مشارکت داشت. در این مورد، گاهی اوقات مدیریت مستقیماً به تیم مراجعه می کند تا به جای نزدیک شدن به صاحب محصول، تغییرات پروژه را درخواست کند. با این حال، در این مورد، تیم توسعه با مدیریت موافقت نکرد و آنها را به صاحب محصول ارجاع داد. چنین فعالیت هایی با فرآیندهای چابک مطابقت نداشت و این نوع فرهنگ در یک سازمان ممکن است بین تیم و صاحب محصول و همچنین مدیریت، تضاد بیشتری ایجاد کند. چنین تعارضاتی ممکن است بر روابط بین مالک محصول و تیم Agile تأثیر بگذارد. در یک تیم، روابط از طریق تعامل بین اعضای تیم شکل گرفت. روابط تیمی شامل به اشتراک گذاری اطلاعات، بازخورد و شناخت نیازها و احساسات اعضای تیم بود. در برخی از پروژه ها در دریافت بازخورد، پشتیبان گیری تیم و تعامل بین اعضای تیم مشکلاتی وجود داشت. بنابراین، به همین دلیل، مشکلات در روابط تیم ایجاد شد. این نشان می دهد که وقتی مشکلی در رابطه بین مالک محصول و تیم توسعه نرم افزار وجود داشته باشد و این ممکن است بر پروژه تأثیر بگذارد.

۴- آیا ماهیت روابط بین مشتری و تیم توسعه تأثیری بر نتیجه پروژه دارد؟

برای درک تأثیر چالش ها بر نتایج پروژه، وابستگی خطی بین جفت متغیرهای مطالعه با استفاده از ضریب همبستگی لحظه محصول پیرسون محاسبه شد. نتایج ارائه شده در بخش ۴،۴ نشان داد که بین زمان اتمام پروژه و اختلاف نظر با مشتری در مورد اولویتهای پروژه رابطه خطی مثبت وجود دارد ($r = 0.329$ $n = 73$ $p =$). این نشان می دهد، اگر اختلاف نظرهایی با صاحب محصول در مورد اولویتهای وجود داشته باشد، احتمالاً زمان اتمام پروژه را افزایش می دهد. وابستگی های خطی بین زمان اتمام پروژه و چالش های بین فردی بین اعضای تیم (ها) و مشتری مثبت بود ($r = 0.394$ $n = 73$ $p = 0.001$). این نشان داد که وقتی چالش های بین فردی بین اعضای تیم توسعه نرم افزار افزایش می یابد، زمان اتمام پروژه نیز افزایش می یابد. به منظور کاهش پیامدهای این روابط، ممکن است سازمان ها نیاز به ایجاد محیطی داشته باشند که به آنها کمک کند.

همچنین ضریب همبستگی بین در دسترس بودن صاحب محصول و اختلاف نظر بین اعضای تیم توسعه و مشتری در مورد مدت زمان پروژه محاسبه شد. داده‌ها رابطه خطی منفی بین موجود بودن صاحب محصول و اختلاف نظر با مشتری در مورد زمان پروژه را نشان می‌دهد. ($r = -0.348$ $n = 73$ $p = 0.001$). این نشان می‌دهد که در صورت افزایش در دسترس بودن صاحب پروژه، اختلاف نظر بین اعضای تیم توسعه و مشتری در مورد مدت زمان پروژه کاهش می‌یابد. این یافته با دیگران همخوانی داشت. به عنوان مثال، Turk و همکاران. (۲۰۰۴) دریافتند که در دسترس بودن مشتری در توسعه نرم افزار Agile در کمک به حفظ ارتباط خوب با تیم بسیار مهم است، زیرا در ردیابی وضعیت پروژه، دعوت از ایده‌ها و انتقادات و شناسایی مسائل پروژه‌ها کمک می‌کند. مفاهیم این رابطه در این مطالعه نشان می‌دهد که در دسترس بودن صاحب محصول در طول توسعه پروژه برای کاهش اختلاف نظر بین مالک و تیم‌های توسعه محصول مهم است.

همبستگی با عوامل دیگر مانند در دسترس بودن صاحب محصول و بازه زمانی پروژه ($r = -0.099$) محاسبه شد. این نشان داد که بین در دسترس بودن صاحب محصول و اعضای تیم، رابطه معناداری وجود ندارد. نشانه‌ای وجود داشت که این دو عامل می‌توانند رابطه منفی داشته باشند. ضریب همبستگی بین مشکلات در اشتراک ایده‌ها در تیم و زمان پروژه و بین چالش‌های بین فردی در تیم، زمان پروژه و بودجه پروژه محاسبه شد. داده‌ها نشان داد که بین این متغیرها رابطه معنی داری وجود ندارد. انتظار می‌رفت که چالش‌های بین فردی بر زمان بندی و بودجه پروژه تأثیر بگذارد. با این حال، داده‌ها نشان داد که بین این متغیرها رابطه معنی داری وجود ندارد. جدول ۴-۱۲ با نتایج ضریب همبستگی در فصل ۴ ارائه شده است: این یافته‌ها تا حدودی به سوال تحقیق ۴ این مطالعه پرداخته‌اند و نشان می‌دهند که ماهیت رابطه بین تیم توسعه نرم افزار Agile و مشتری بر نتیجه پروژه تأثیر داشته است. تأثیر عوامل مشخص شده از مطالعه پروژه‌های توسعه نرم افزار در جدول ۶-۱ خلاصه شده است:

جدول ۶-۱ تاثیر عوامل روی پروژه

عوامل	تاثیر
عدم توافق با مشتری در مورد اولویت های پروژه	زمان اتمام پروژه
در دسترس بودن مالک محصول	عدم توافق مشتری با اولویت های پروژه
چالش های بین فردی بین اعضای (اعضای) تیم و مشتری	زمان اتمام پروژه
اختلاف نظر بین اعضای تیم (ها) و مشتری در مورد الزامات پروژه	زمان اتمام پروژه

مدل کار گروهی در پروژه‌های توسعه نرم‌افزار چابک این بخش با استفاده از مدل کار تیمی برای پاسخ به سوالات تحقیق بحث می‌کند.

۵- آیا مدل کار گروهی می‌تواند به طور کافی ماهیت رابطه بین مشتری و تیم توسعه را در زمینه توسعه نرم‌افزار به سبک Agile توضیح دهد؟

پس از تجزیه و تحلیل داده‌های مصاحبه از تیم توسعه نرم‌افزار و مطالعه صاحب محصول، هفت عامل از مطالعه صاحب محصول و هشت عامل از مطالعه تیم توسعه نرم‌افزار مشخص شد. این عوامل مشخص شد به عنوان چالش‌هایی که ممکن است موفقیت پروژه‌های توسعه نرم‌افزار Agile را تحت تأثیر قرار دهد. جدول ۶-۲ عوامل تیم توسعه نرم‌افزار و مطالعه صاحب محصول را ارائه می‌دهد. جدول ۶-۳ شرح هر یک از عوامل شناسایی شده در این مطالعه کنونی را ارائه می‌دهد.

عوامل شناسایی شده از مطالعه تیم توسعه نرم افزار	عواملی که از مطالعه صاحب محصول مشخص شده است
ارتباطات	ارتباطات
پشتیبانی تیم	پشتیبانی تیم
بازخورد	بازخورد
مشارکت صاحب محصول	مشارکت صاحب محصول
تعامل تیمی	تعامل تیمی
اعتماد به نفس	اعتماد به نفس
فرهنگ سازمانی	فرهنگ سازمانی
انگیزه تیمی	

جدول ۳-۶ شرح عوامل شناسایی شده در مطالعه فعلی

عوامل	توضیحات
ارتباطات	ارتباط فرآیند انتقال اطلاعات بین افرادی است که در یک پروژه مشارکت دارند.
پشتیبانی تیم	بازخورد به ارائه اطلاعات در مورد عملکرد و پذیرش پیشنهادات ارائه شده در مورد پروژه اشاره دارد.
بازخورد	پشتیبانی تیم شامل در دسترس بودن و تمایل به کمک به دیگر اعضای تیم است. اعضای تیم توسعه، از جمله مالک محصول، برای حل مشکلات در محدوده پروژه‌ها با هم همکاری می‌کنند.
مشارکت صاحب محصول	یک تیم خودگردان با فرهنگ اشتراک مسئولیت. تیم‌های موفق توسعه چابک به مالکیت محصول، هم از درون تیم و هم از درون سازمان، نیاز دارند که ماهیت تکراری فرایند چابک را درک کرده و از آن پشتیبانی می‌کند.
تعامل تیمی	فعالیت‌های اعضای تیم در طول پروژه‌ها. فعالیت‌های هر یک از اعضای تیم به گونه‌ای تراز شده است که اهداف مرتبط با هدف پروژه به دست آید.
اعتماد به نفس	نقش صاحب محصول در فعالیتهای مختلف پروژه از شروع پروژه تا استقرار.
فرهنگ سازمانی	اعتماد، در زمینه این مطالعه، به اعتقاد به شایستگی اعضای تیم در انجام وظیفه اشاره دارد
انگیزه تیمی	انگیزه، در زمینه این مطالعه، به تمایل اعضای تیم توسعه برای تکمیل کار اشاره دارد

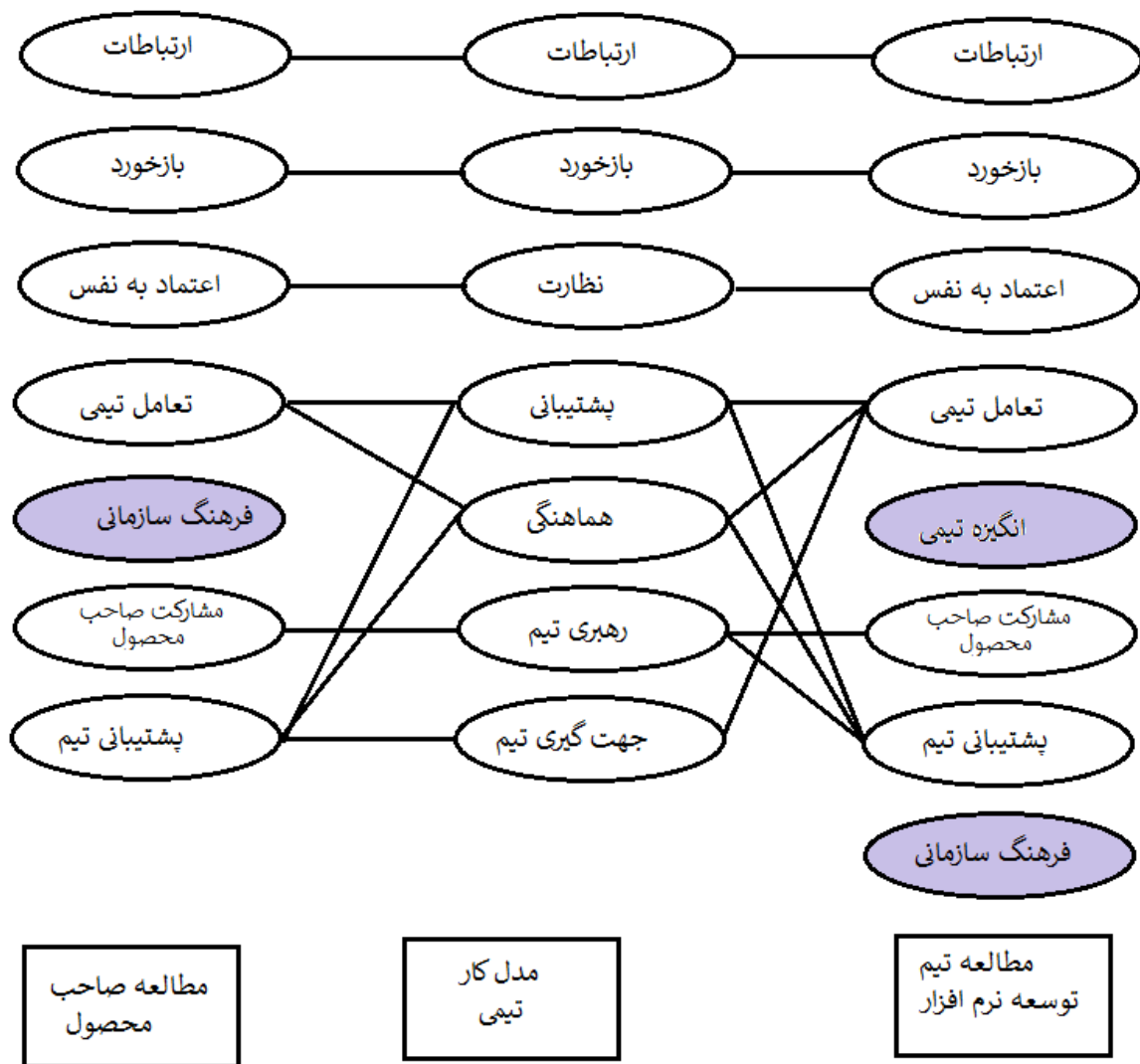
Dickinson و McIntyre (۱۹۹۷) مدل کار گروهی خود را با هفت جزء ایجاد کردند: ارتباط، جهت‌گیری تیم، رهبری تیم، نظارت، بازخورد، پشتیبان‌گیری و هماهنگی. آنها پیشنهاد کردند که رهبری تیم و جهت‌گیری تیم بر نظارت تأثیر می‌گذارد، که این امر به آگاهی از فعالیت‌ها و عملکرد اعضای تیم اشاره می‌کند. نظارت برای اطمینان از اینکه اعضای تیم قادر به انجام وظایف فردی خود هستند و درک اساسی از وظایف سایر اعضا دارند مفید بود.

دیکنسون و مک اینتایر (۱۹۹۷) پیشنهاد کردند که نظارت بر عملکرد تیم هم بازخورد و هم رفتار پشتیبان به موقع را ارائه می‌دهد. وقتی اعضای تیم بر یکدیگر نظارت می‌کنند، می‌توانند در مورد کار یکدیگر بازخورد ارائه دهند. اعضای تیم همچنین می‌توانند در صورت مواجهه با هرگونه مشکل با سایر اعضای تیم کمک کنند، زیرا هر یک از اعضا از کار اعضای دیگر از طریق نظارت آگاه بودند. در این مطالعه مشخص شد که اعضای تیم در نظارت بر کار یکدیگر نقش ندارند. با این حال، سه نفر از پاسخ دهندگان صاحب محصول گزارش دادند که آنها در نظارت بر عملکرد تیم نقش دارند. این نشان می‌دهد که نظارت یک رویکرد معمول نیست که برای مشاهده کار اعضای تیم و ارائه بازخورد در مورد کار آنها استفاده می‌شود. جدول ۶-۴ عوامل همراه با شرح مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷) را ارائه می‌دهد.

یک تمرین نقشه‌برداری برای شناسایی عوامل مشترک از مدل کار تیمی، مطالعه تیم توسعه نرم‌افزار و مطالعه صاحبان محصول انجام شد. شکل ۶-۱ نداشت مدل کار تیمی را با این مطالعه فعلی ارائه می‌دهد. شرح هر یک از این عوامل در رابطه با مدل کار گروهی در قسمت زیر توضیح داده شده است.

جدول ۴-۶ اجزای مدل کار تیمی

ارتباط	ارتباط شامل تبادل فعال اطلاعات بین اعضای تیم است. ارتباط بین سایر اجزای مدل کار گروهی پیوند ایجاد می کند.
جهت گیری تیم	این مولفه شامل خودآگاهی و نگرش اعضای تیم نسبت به رهبری است.
رهبری تیم	این مولفه از مدل کار گروهی شامل هدایت اعضای تیم است و دلالت بر برنامه ریزی و سازماندهی فعالیتها دارد تا اعضای تیم بتوانند پاسخ دهند.
بازخورد	این مولفه به یادگیری از عملکرد اعضای تیم اشاره دارد که مستلزم دریافت، ارائه و بازخورد از سایر اعضای تیم است.
پشتیبان	این مورد به عضوی از تیم اشاره می کند که به اعضای دیگر تیم در انجام یک کار کمک می کند.
هماهنگی	این مولفه به فعالیت هایی اشاره می کند که به اعضای تیم تا عملکردها را تکمیل کنند، کمک می کند (دیکنسون و مک اینتایر ، ۱۹۹۷).



شکل ۶-۱ نگاشت مدل کار تیمی با نتایج تیم توسعه نرم افزار و مطالعات صاحب محصول

۴-۱ پشتیبانی تیم

پشتیبانی تیم یکی از عوامل مهم در مطالعات تیم توسعه نرم افزار و مطالعات صاحب محصول بود. موضوع پشتیبانی تیم در این مطالعه مربوط به تمایل اعضای تیم برای کمک به یکدیگر برای انجام یک کار است. صاحبان محصول و پاسخ دهندگان تیم توسعه نرم افزار گزارش داده اند که حمایت تیم از طریق رهبری مشترک، همکاری و داشتن نگرش کمک کننده نسبت به سایر اعضای تیم نشان داده شده است. با این حال، ممکن است اعضای تیم به منظور حمایتی که می توانند از دیگر اعضای تیم انجام دهند، کار سایر اعضای تیم را مشاهده نکنند. این بدان معناست که اعضای تیم ممکن است مدت زیادی طول بکشد تا بفهمند اعضای دیگر چه کار می کردند تا قبل از اینکه از

آنها حمایت کنند. تأثیر دیگر این امر این بود که شواهد کمی از پشتیبان‌گیری (پشتیبانی تیم) بین تیم و مالک محصول وجود داشت زیرا اعضای تیم از کار سایر اعضا آگاه نبودند. به عنوان مثال، اعضای تیم در واقع مایل بودند در صورت نیاز از سایر اعضای تیم پشتیبانی کنند. برخی پروژه‌ها وجود داشت که در آن اعضای تیم حاضر نبودند از سایر اعضای تیم کمک بگیرند زیرا این اعضا دوست داشتند به تنهایی کار کنند. این نشان می‌دهد که پشتیبانی تیم با جهت‌گیری تیم، ساختار پشتیبان و هماهنگی مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷) همپوشانی دارد. حمایت تیمی برای کار تیمی موثر مهم بود. شنگ و همکاران (۲۰۱۰) اشاره کرد که حمایت تیم به طور قابل توجهی بر رفتار کار تیمی و اعتماد به تیم تأثیر گذاشته است. همچنین مشخص شد که بر نگرش‌ها و رفتار اعضای تیم در داخل تیم، از جمله تعهد آنها تأثیر گذاشته است (شنگ، تیان، و چن، ۲۰۱۰). این نشان می‌دهد که وقتی فردی نمی‌تواند وظایف محوله را به طور جداگانه انجام دهد، می‌تواند برای انجام کار از همکاران خود درخواست پشتیبانی کند.

۲-۴ بازخورد

سبزه پاسخ دهنده از تیم توسعه نرم‌افزار و سه نفر از صاحب محصول گزارش کردند که اعضای تیم در ارائه و دریافت اطلاعات مربوط به پروژه مشارکت داشته‌اند. مدل کار تیمی نشان داد که بازخورد درون تیم می‌تواند در مورد هماهنگی در تیم اطلاعاتی ارائه دهد. این امر اهمیت بازخورد را برای کار گروهی موثر نشان می‌دهد. وقتی اعضای تیم احساس کردند که کار آنها و خودشان برای پروژه مهم و ارزشمند است، این احساسات مثبتی را ایجاد کرد که تیم را تشویق کرد تا وظایف داده شده را انجام دهد (Amabile & Kramer, ۲۰۰۷). چهار عضو تیم گزارش کردند که صاحب محصول آنها را تشویق کرده است تا با گوش دادن به بازخوردی که اعضای تیم ارائه کرده‌اند، در مورد پروژه‌ها بازخورد ارائه دهند.

کاتزنباخ و اسمیت (۲۰۰۵) دریافتند که بازخورد اعضای تیم موثر است زیرا باعث ارتقاء عملکرد فردی و همچنین تیم می‌شود. پاسخ دهندگان این مطالعه اظهار داشتند که پذیرش بازخورد به تجربه اعضای تیم در ارائه بازخورد بستگی دارد. آنها همچنین گزارش دادند که مواردی وجود داشته است که صاحب محصول بازخورد را پذیرفته اما بعداً این تصمیم را تغییر داده است. این نشان می‌دهد که صاحب محصول از برخی الزامات پروژه‌ها اطمینان ندارد. یکی از پاسخ دهندگان مالک محصول گفت که آنها هرگز از کاربران نهایی محصول خود بازخوردی دریافت نکرده‌اند زیرا هیچ تماسی با آنها نداشته‌اند. این ممکن است به این معنی باشد که بازخوردی از کاربر دریافت نشده است، به توسعه دهندگان دقیقاً آنچه مورد نیاز است ارائه نشده است یا صاحب محصول از آنچه دقیقاً مورد نیاز است

مطمئن نیست. این عدم بازخورد ممکن است در صورت لزوم مانع پیشرفت محصول شود. با گفتن این نکته، دشوار بود بدانیم دقیقاً چگونه می‌توان این بازخورد را دریافت کرد. در پروژه‌های دیگر این مطالعه که مالک محصول به کاربران نهایی دسترسی داشته باشد، وجود نداشت. این نشان داد که فرایند بازخورد در پروژه به عواملی مانند دسترسی به کاربر نهایی و پذیرش بازخورد توسط صاحب محصول بستگی دارد. این همچنین نشان داد که وقتی اعضای تیم به کاربران نهایی دسترسی پیدا کنند، می‌توانند بازخورد پروژه را در حال توسعه دریافت کنند.

با توجه به اینکه مالک محصول در اکثر پروژه‌ها تیم را زیر نظر نداشت، بازخورد کمی از مالک محصول در مورد عملکرد اعضای تیم وجود داشت. این نشان می‌دهد که وقتی صاحب محصول از کار اعضای تیم مطلع بود، در صورت نیاز می‌توان بازخورد ارائه کرد. در چنین سازمان‌هایی که بازخورد کمی وجود داشت یا هیچ خبری نبود، اعضای تیم روی کار خود کار می‌کردند و به طور کامل درگیر کار گروهی نبودند. برخی از تیم‌ها بازخوردی را برای صاحب محصول ارائه کردند اما پذیرش بازخورد توسط صاحب محصول به تجربه اعضای تیم بستگی داشت که بازخورد را ارائه کردند. این نشان می‌دهد که صاحب محصول تمایل به گوش دادن به افراد باتجربه به جای افراد کم تجربه را دارد. رامش و همکاران (۲۰۰۶) دریافتند که اعضای ارشد، که اعتماد صاحب محصول را از روابط کاری قبلی جلب کرده بودند، توسط صاحب محصول مورد توجه قرار گرفت. این همچنین نشان می‌دهد که برخی از سازمانها وجود دارند که در آنها سطح مشارکت صاحب محصول باید بهبود یابد و صاحب محصول برای تکمیل کار باید به تیم خود ایمان داشته باشد. باور به تیم ممکن است با عملکرد خوب و با انجام به موقع وظایف افزایش یابد.

وقتی تیم از طریق نظارت بر کار یکدیگر آگاه شدند، می‌توانند وظایف یکدیگر را بفهمند، روی چه چیزی کار می‌کنند و در تلاش برای حل چه چیزی بوده‌اند. اطلاعات بیست و دو پاسخ دهنده از توسعه نرم‌افزار و مطالعه صاحب محصول نشان می‌دهد که اعضای تیم در مشاهده کارهای دیگر اعضای تیم دخالت ندارند. این ممکن است به این معنی باشد که اعضای تیم ممکن است بر روی کار خود تمرکز کرده و به صورت فردی کار کرده باشند. لانگفرد (۲۰۰۰) اظهار داشت که هنگامی که تیم‌ها سعی می‌کردند همانند یک نفر عمل کنند، می‌تواند تأثیر منفی بر عملکرد تیم داشته باشد. وقتی استقلال فردی پایین باشد، اعتماد به نفس افزایش می‌یابد و این باعث عملکرد بهتر در تیم شد (لانگفرد، ۲۰۰۴). مارکس و همکاران (۲۰۰۱) دریافتند که اعضای تیم برای ارائه کمک به تیم باید از کارهای یکدیگر آگاه باشند. این نشان می‌دهد که یک تیم، از جمله صاحب محصول، باید با یکدیگر

همکاری نزدیک داشته باشند تا از کار یکدیگر مطلع باشند و بتوانند در مورد کار و عملکرد خود پشتیبانی و بازخورد ارائه دهند. در این مطالعه، مشخص شد که این یک روش معمول در همه پروژه ها نیست.

در مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷) ساختارهای بازخورد و پشتیبان‌گیری تیم برای هماهنگی اعضای تیم مهم بود. این مهم بود زیرا وقتی اعضای تیم از کار سایر اعضای تیم مطلع بودند و می‌توانستند بازخورد و پشتیبانی پشتیبان ارائه دهند، به احتمال زیاد ارتباط خوبی بین اعضای تیم وجود دارد که هماهنگی را بین اعضای تیم به نحوه خوبی تسهیل می‌کرد. پاسخ دهندگان این مطالعه گزارش دادند که مشارکت تیم توسعه نرم‌افزار Agile در یک پروژه بین سازمان‌ها متفاوت است. در شصت و چهار درصد از پروژه‌ها، اعضای تیم توسعه نرم‌افزار در تجزیه و تحلیل داستان‌های کاربر برای ارائه جزئیات یک برنامه کاری مشارکت داشتند. سی و شش درصد از شرکت‌کنندگان در مطالعه توسعه نرم‌افزار در تیم‌هایی بودند که به جای مشارکت در برنامه ریزی، وظایفی به آنها اختصاص داده شد. تقسیم کار بر اساس مهارت‌هایی بود که اعضای تیم داشتند، که به اعضای تیم این امکان را می‌داد تا کار را به نحو احسن انجام دهند. در چهار پروژه که توسط پاسخ دهندگان در این مطالعه توصیف شده بود، اعضای تیم به صورت فردی هدایت شدند. این شیوه کار فردی، هر چند متعلق به تیم بود، اما در مورد هماهنگی و پیروی از فرایند چابکی مشکلاتی را در تیم‌های مربوط ایجاد کرد. مشکلاتی که مشاهده شد این بود که اعضای تیم هنگامی که به تنهایی کار می‌کردند از سایر اعضای تیم مطلع نبودند. در این شرایط، اعضای تیم ممکن است از پیشرفت‌هایی که در یک کار خاص ایجاد می‌شود یا از هر گونه مشکلی که در حین انجام این کار رخ می‌دهد، آگاه نباشند. مفهوم این امر برای نتایج موفق جایی است که اعضای تیم می‌توانند بازخورد خود را ارائه کرده و به صورت گروهی کار کنند. سه شرکت کننده در مطالعه صاحب محصول گزارش دادند که تیم هیچ نظری در مورد اولویت پروژه‌ها نداشت. این ممکن است به این معنی باشد که صاحب محصول به کار با فرایندهای چابک عادت نداشته است. پیامدهای این امر بر نتیجه یک پروژه ممکن است این باشد که صاحب محصول باید از روش صحیح چابک پیروی کرده و با تیم هماهنگ باشد.

۳-۴ مشارکت صاحب محصول

مشارکت صاحب محصول یکی دیگر از عوامل ایجاد نرم‌افزار Agile در توسعه نرم‌افزار و مطالعه صاحب محصول گزارش شده است. مشارکت صاحب محصول در اینجا به مشارکت صاحب محصول در طول فعالیت‌های مختلف یک پروژه، از ابتدا تا مرحله انتشار اشاره دارد. از این مطالعه، ۱۹ نفر از ۲۹ نفر (۶۶ درصد) از پاسخ دهندگان گزارش دادند که صاحب محصول با تیم در تصحیح الزامات، برنامه‌ریزی عقب افتاده و شفاف‌سازی جنبه‌هایی که

توسط اعضای تیم درک نمی‌شود، مشارکت داشته است. سه سازمان وجود داشت که صاحبان محصولات در چنین فعالیت‌هایی شرکت نمی‌کردند. ممکن است آنها فرایند چابک را دنبال نکنند یا نقش آنها این باشد که تمام پروژه‌ها و همچنین تیم توسعه را بررسی کنند و در صورت بروز مشکل به تیم مراجعه کنند. مشارکت صاحب محصول در پروژه در توسعه نرم‌افزار Agile بسیار مهم بود (Beck et al., ۲۰۰۱). به عنوان مثال، Moe و همکاران، (۲۰۰۸) دریافتند که مشارکت اعضای تیم، از جمله صاحب محصول در پروژه، وابستگی احساسی به سازمان، انگیزه برای عملکرد بهتر و تمایل به مسئولیت‌پذیری را بهبود می‌بخشد. این می‌تواند به این معنی باشد که وقتی صاحب محصول در پروژه درگیر بود، اعضای تیم در تکمیل وظایف خود احساس بهتری داشتند زیرا می‌توانستند به سرعت بازخورد و راهنمایی دریافت کنند.

همه صاحبان محصولات درگیر در این مطالعه از اختیار کامل برای تصمیم‌گیری در مورد پروژه‌هایی که به آنها اختصاص داده شده برخوردار نبودند. برخی اوقات مدیریت ارشد در برخی از سازمان‌ها در تصمیم‌گیری مشارکت داشتند. در این موارد، عدم اختیار ممکن است باعث ایجاد سردرگمی در تیم‌ها شود و احتمال از دست رفتن مهلت را افزایش می‌دهد. مو و همکاران (۲۰۱۲) دریافتند که اختیاراتی که به صاحب محصول داده می‌شود در تصمیم‌گیری‌ها کمک می‌کند، مانند مواردی که باید در گام بعدی تهیه شود، چگونه مشکل حل شود و منابع را کجا تخصیص دهیم. هنگامی که صاحب محصول اختیار کامل در پروژه‌ای نداشت، تصمیم‌گیری در طول توسعه پروژه بیشتر طول می‌کشید. در برخی از پروژه‌های این مطالعه فعلی، در دسترس بودن صاحب محصول یک مسئله بود. هنگامی که صاحب محصول در دسترس نبود، تأخیر در تکمیل وظایف پروژه وجود داشت. این به این دلیل بود که تیم باید منتظر بماند تا نظر صاحب محصول را بشنود و این می‌تواند زمان تکمیل کار را افزایش دهد. گاهی اوقات، این می‌تواند باعث ناامیدی در تیم‌ها شود.

سه پاسخ دهنده از صاحبان محصول گزارش کردند که چندین بار به صاحب محصول چندین نقش محول شده است و این به عنوان یکی از چالش‌های آنها گزارش شده است. به عنوان مثال، در یک مورد، یک پاسخ دهنده نقش عادی خود را در زمینه توسعه نرم‌افزار در تیم ایفا می‌کرد در حالی که آنها همچنین به عنوان مالک محصول عمل می‌کردند. این بدان معناست که صاحب محصول، در این مورد خاص، در پروژه‌ای با نقش‌های متعدد مشارکت داشته است. هنگامی که صاحب محصول نقش‌های متعددی در یک پروژه دارد، می‌تواند با مشکلاتی روبرو شود و از نظر اعضای تیم و صاحبان محصول به پروژه نگاه کند. این ممکن است باعث ایجاد تعارض در نقش آنها شود. (Mahnic, ۲۰۱۲) دریافت که بهتر است نقش‌های متعدد صاحب محصول را جدا کنید، بنابراین فرد می‌تواند

بر نقش خارج از تعهدات معمول خود تمرکز کند. یافته‌های حاصل از توسعه نرم‌افزار و مطالعات مالک محصول نشان می‌دهد که مشارکت و هماهنگی صاحب محصول، با توجه به مدل دیکنسون و مک اینتایر (۱۹۹۷) به عنوان مالک محصول در فعالیتهایی که شرکت داشتند به اعضای تیم در تکمیل وظایف خود کمک می‌کند. صاحبان محصول همچنین در جهت دهی به اعضای تیم جهت برنامه‌ریزی و سازماندهی فعالیت‌ها به منظور انجام وظایف خود مشارکت داشتند.

پیامدهای سطوح مشارکت صاحب محصول می‌تواند دو تأثیر داشته باشد، یکی مثبت و دیگری منفی. تأثیر مثبت بر تیم این بود که ممکن است ارتباطات بهتری برقرار شود زیرا مالک محصول می‌تواند در زمان نیاز تیم به آنها در دسترس باشد. اگر تیم نگران باشد، مانند جدول زمانی دو سرعت یا پیچیدگی کار، آنها می‌توانند به صاحب محصول مراجعه کنند. در جنبه‌های منفی، تأثیر حضور کمتر مالک محصول در ارتباط با تیم می‌تواند روند ارتباط را به تأخیر بیندازد. همچنین ممکن است صاحب محصول از جزئیات پروژه مطلع نباشد.

۴-۴ تعامل تیمی

جالب است بدانید که مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷) شامل مشارکت تیمی به عنوان یک جزء جداگانه نبود. اغلب انتظار می‌رود که تیم در فعالیتهایی مانند استندآپ و توسعه نرم افزار Agile مشغول است (Drury et همکاران، ۲۰۱۲). این فعالیتهای سطح بالایی از مشارکت نیاز داشت سطح مشارکت اعضای تیم ممکن است بر ارتباط و همکاری و مولفه بازخورد مدل کار گروهی تأثیر داشته باشد. این نشان دهنده همپوشانی بین موضوع تعامل تیمی از این مطالعه و جهت‌گیری تیم، پشتیبانی و هماهنگی

از مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷) است. از آنجا که اعضای تیم در فعالیتهایی مشارکت داشتند که به تیم برای انجام وظایف خود کمک می‌کند، باید سطح هماهنگی و تعامل وجود داشته باشد. اعضای تیم همچنین در برنامه ریزی و سازماندهی فعالیت‌ها برای تکمیل وظایف مشارکت داشتند. جالب بود که در این مطالعه موضوع تعامل تیم قوی بود و افراد درگیر در توسعه نرم‌افزار در مورد اهمیت آن در تکمیل پروژه به صورت تیمی صحبت کردند. تأثیر مثبت مشارکت تیم در شیوه‌های مختلف چابک در طول پروژه‌های توسعه نرم‌افزار به تیم کمک می‌کند تا از پیشرفت‌های انجام شده در پروژه‌ها آگاه شوند. این همچنین فرصتهایی را برای اعضای تیم فراهم کرد تا در صورت داشتن هرگونه مشکل در مورد پروژه‌ها به طور مفصل صحبت کنند. اگر اعضای تیم به طور کامل در پروژه‌ها مشارکت نداشته باشند، ممکن است در ارتباط و هماهنگی تیم مشکلاتی ایجاد شود.

۵-۴ اعتماد به نفس

در این بخش، دو اصطلاح اطمینان و اعتماد استفاده شد. اطمینان تیمی، در زمینه این مطالعه، به اعتقادی اشاره دارد که صاحب محصول در تکمیل کار در اعضای تیم دارد. اعتماد به توانایی اعضای تیم برای تکمیل وظایف به روش درست اشاره کرد. اعتماد همچنین به اعتماد به نفس، قدرت و توانایی اعضای تیم در انجام وظیفه (Dingsøyr et al., ۲۰۱۲; Gefen & Straub, ۲۰۰۴) اشاره دارد. هفت عضو تیم توسعه نرم‌افزار درگیر در این مطالعه گزارش کردند که صاحبان محصول به تیم اعتماد دارند زیرا صاحب محصول به نگرانی‌های آنها گوش داده است. با بیان این که دو شرکت کننده در مطالعه تیم توسعه نرم‌افزار اظهار داشتند که اعتماد به تیم به تجربه اعضای تیم بستگی دارد. با بیان این مطلب، تصور درون تیم این بود که بیشتر اعضای جوان تیم به اندازه اعضای ارشد تیم مورد اعتماد صاحب محصول نیستند. کسانی که تجربه بیشتری در کار در پروژه توسعه نرم‌افزار دارند و روابط کاری خود را در پروژه های قبلی نشان داده اند. در حالی که آنها اطلاعات چندانی از اعضای جوان تر و توانایی های آنها نداشتند. پروژه هایی وجود داشت که در آن صاحب محصول کار تیمی را از نزدیک مشاهده می کرد زیرا آنها از توانایی تیم در انجام وظایف خود اطمینان نداشتند. ممکن است اعضای تیم هنوز کار خود را نشان نداده باشند یا در رابطه کاری با صاحب محصول تازه کار باشند. مفاهیم این سطح از نظارت تیمی می تواند دو اثر داشته باشد، یکی منفی، یکی مثبت. تأثیر منفی این بود که تیم ها از مدیریت ناراضی بودند و از کار با صاحب محصول خاص خوشحال نبودند. جنبه معکوس آن تأثیر مثبتی داشت زیرا صاحبان محصول با تیم های خود در سطح بسیار عمیق تری آشنا می شدند و می توانستند مشاهده کنند که آنها قادر به انجام چه کاری هستند و بنابراین، به آنها اعتماد بیشتری در آینده بدهند.

پنج شرکت کننده صاحب محصول گزارش کردند که به تیم خود برای تکمیل وظایف خود اعتماد دارند. با این حال، دو شرکت کننده صاحب محصول گزارش کردند که به تیم خود اعتماد ندارند، زیرا فکر می کردند وظایف خود را به درستی برآورد نکرده است. این می تواند به این دلیل باشد که صاحب محصول ممکن است برآورد خود را از کار در پروژه های دیگر، که ممکن است انتظارات صاحب محصول را برآورده نکرده است، مشاهده کرده باشند. عدم اعتماد به تیم ممکن است زمان بندی پروژه را افزایش دهد زیرا ممکن است زمان بیشتری برای مشاهده کار یک فرد یا تیم لازم باشد. برای دستیابی به اعتماد، نیاز به رهبری، بازخورد و ارتباط مشترک است، به طوری که به اعضای تیم اجازه داده شود آنچه را که می توانند انجام دهند نشان دهند و همچنین فرصتی برای گسترش مهارت های خود فراهم شود (ثلاث، سیمز، و برک، ۲۰۰۵). آنها همچنین اشاره کردند که اعتماد زمانی قابل

دستیابی است که اعضای تیم مایل به پذیرش اشتباهات تصدیق پیام دریافتی و شفاف‌سازی پیام‌های دریافتی باشند. تحقیقات قبلی توسط لانگفرد (۲۰۰۴) نشان داد که اعتماد به تیم با عملکرد بهتر تیم ارتباط دارد. با این حال، آنها همچنین ذکر کردند که اعتماد بیش از حد به یک تیم ممکن است مضر باشد زیرا خودمختاری فردی تیم بالا خواهد بود. این نشان می‌دهد که برای حفظ خودمختاری افراد نیاز به نظارت بر تیم است. همچنین پیشنهاد شد که اعتماد برای کار تیمی موثر و در توسعه پروژه‌های موفق مهم است. مشخص شد که اعتماد با ساختار نظارتی مدل کار تیمی دیکنسون و مک اینتایر همپوشانی دارد (دیکنسون و مک اینتایر، ۱۹۹۷). این همپوشانی در شرایطی مشاهده شد که صاحب محصول از فعالیت‌ها و عملکرد اعضای تیم آگاهی داشت و در کنار آن به تیم برای اطمینان از انجام کار اطمینان داشت.

با توجه به مدل کار تیمی دیکنسون و مک اینتایر (دیکنسون و مک اینتایر، ۱۹۹۷)، نظارت برای اطمینان از اینکه اعضای تیم قادر به انجام وظایف فردی خود هستند و درک قابل توجهی از وظایف سایر اعضا دارند، مفید بود. نظارت ممکن است فرصتی را برای مالک محصول ایجاد کند تا پیشرفت اعضای تیم را ببیند. هنگامی که صاحب محصول از کار و پیشرفت اعضای تیم مطلع بود که ممکن است به اعضای تیم در جلب اعتماد صاحب محصول کمک کند. تنها یکی از پاسخ دهندگان از مطالعه صاحب محصول اشاره کرد که مالک محصول در نظارت بر عملکرد اعضای تیم نقش داشته است زیرا مالک محصول به تیم اعتماد ندارد. این نشان می‌دهد که نظارت رایج در پروژه‌های توسعه نرم‌افزار Agile استفاده نمی‌شود. جنبه مثبت این امر می‌تواند این باشد که صاحبان محصول به تیم‌ها اعتماد داشتند. وقتی صاحب محصول به تیم اعتماد داشت، تیم انگیزه پیدا کرد. با این حال، تأثیر منفی عدم نظارت بر تیم ممکن است باعث شود که تیم انتظارات صاحب محصول را برآورده نکند و ممکن است نیاز به انجام دوباره کارهایی در پروژه باشد که می‌تواند اتمام پروژه را به تأخیر بیندازد.

۶-۴ ارتباطات

ارتباطات بیشترین (۶۴٪) چالش‌های گزارش شده توسط تیم توسعه نرم‌افزار و صاحبان محصول در پروژه‌ها را گزارش کرده است. چالش‌هایی مانند اختلاف نظر در مورد الزامات پروژه و در دسترس بودن صاحب محصول، بر زمان‌بندی پروژه‌ها تأثیر دارد. هر دو مطالعه گزارش داده‌اند که ارتباطات با استفاده از رسانه‌های مختلف مانند ایمیل، یا استفاده از برخی برنامه‌ها یا حضوری انجام شده است. هر دو مطالعه همچنین گزارش داده‌اند که چالش‌هایی می‌تواند در طول ارتباط رخ دهد. چنین چالش‌هایی به دلیل تفاوت در زبان و فرهنگ منتهی به ارتباط نادرستی شدند. مطالعه‌ای که توسط (Matveev & Nelson، ۲۰۰۴) با ۱۲۴ مدیر فوقانی و میانی انجام شد،

نشان داد که بین ارتباطات چند فرهنگی و عملکرد تیم رابطه وجود دارد. در طول توسعه پروژه، تیم توسعه و مالک محصول ممکن است در مورد پروژه‌ها بحث‌های غیر رسمی داشته باشند. این بحث می‌تواند در مورد ویژگی‌های پروژه‌ها، تخصیص منابع یا تغییرات در پروژه‌ها باشد. برخی از این بحث‌های غیر رسمی را می‌توان از دست داد، زیرا مستند نمی‌شوند. این نشان می‌دهد که اطلاعات به اشتراک گذاشته شده در طول ارتباطات غیررسمی فراموش شده است یا ممکن است این بحث مهم در محل صحیح انجام نشده باشد، مانند، زمان استندآپ. دیکنسون و مک اینتایر (۱۹۹۷) گفتند که ارتباطات در کار گروهی در حین نظارت، بازخورد، پشتیبان‌گیری، رهبری تیم، جهت‌گیری تیم و هماهنگی تیم دخیل است. آنها همچنین ذکر کردند که داشتن ارتباط موثر در طول فرایند برای کار تیمی بهتر مهم است. در مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷)، ارتباط مولفه اصلی بود که فرآیندهای کار تیمی را با هم مرتبط می‌کرد. در این مطالعه گزارش شده بود که استندآپ‌های روزانه رایج‌ترین مکانیسم ارتباطی در پروژه‌های چابک بودند (Paasivaara, Durasiewicz, & Lassenius, ۲۰۰۸). این به این دلیل بود که در طول استندآپ همه اعضای تیم دور هم جمع می‌شدند و اطلاعات را به اشتراک می‌گذاشتند. این شامل پیشرفت پروژه، مشکلات در پروژه و سایر اطلاعات مربوط به پروژه بود. در طول توسعه نرم‌افزار، اعضای تیم ممکن است در قسمت‌های مختلف کار کنند. ممکن است در ارتباط با کار بین مناطق زمانی مختلف، به زبان‌های مختلف و عدم دسترسی صاحب محصول، چالش‌هایی وجود داشته باشد. مشکلات مربوط به تفاوت منطقه زمانی را می‌توان با یافتن یک زمان عمومی که در دسترس همه مردم است برطرف کرد. اگر این مشکل همچنان پابرجا بود، مشکلی در ارتباط وجود دارد. با توجه به مدل کار گروهی، مشکلات ارتباطی می‌تواند مسائلی در هماهنگی، بازخورد و رفتار پشتیبان ایجاد کند. این نشان می‌دهد که ارتباط در گردآوری همه اجزای مدل کار تیمی برای دستیابی به کار تیمی خوب و موثر در تیم‌های نرم‌افزاری چابک مهم است. در بخش بالا، عوامل مختلف، مشخص شده از هر دو مطالعه، در زمینه مدل دیکنسون و مک اینتایر (۱۹۹۷) مورد بحث قرار گرفت. موضوعات مشخص شده در این مطالعه، مانند اعتماد به نفس تیم، مالک محصول و تعامل تیمی، همپوشانی وجود داشت. با این حال، مدل کار گروهی از فرهنگ سازمانی یا انگیزه تیم الگو نمی‌گیرد. این دو عامل اضافی مشخص شده از این مطالعه در مطالعه قبلی توسط Moe و همکاران (۲۰۱۰) مشخص نشد. آنها در مطالعه خود پیشنهاد کردند که کار تیمی تمدید شود. این مدل می‌تواند به تیم توسعه نرم‌افزار چابک بالغ برای درک بهتر از کار تیمی کمک کند. هر یک از عوامل شناسایی شده در این مطالعه، در مدل کار گروهی وجود ندارد، اکنون به تفصیل مورد بحث قرار می‌گیرد.

۷-۴ انگیزه تیمی

یازده نفر از پاسخ دهندگان گزارش دادند که انگیزه اعضای تیم دلیل موفقیت تیم و موفقیت پروژه است. اعضای تیم انگیزه داشتند زیرا احساس می کردند برای مشارکت در پروژه ها از حقوق کافی برخوردار هستند. یکی دیگر از عوامل مهم، فرهنگ اجتماعی سازمان بود. در سازمان هایی که به طور منظم مسائل کارکنان برگزار می شود، سطح انگیزشی آنها بیشتر از سازمان هایی بود که فرهنگ اجتماعی نداشتند. دو نفر از بیست و دو شرکت کننده گزارش کردند که برخی از فعالیت های مدیریت، مانند عدم گوش دادن به نگرانی های تیم، بر انگیزه اعضای تیم تأثیر می گذارد. به عنوان مثال، یکی از پاسخ دهندگان می گوید که صاحب محصول آنها به بازخورد گوش نمی دهد و به همین دلیل، آنها انگیزه ای برای ارتباط بیشتر با صاحب محصول در مورد هر چیزی ندارند. نگرانی هایی که در حین توسعه پروژه وجود آمد. مطالعه ای که توسط Moe و همکاران انجام شده است. (۲۰۱۲) اشاره کرد که افزایش انگیزه در اعضای تیم باعث می شود کارکنان به کار خود توجه بیشتری داشته باشند و باعث ایجاد خلاقیت بیشتر، کمک به رفتار و بهره وری بیشتر می شود. مطالعه دیگری که توسط Ceschi و همکاران (۲۰۰۵) انجام شد اشاره کرد که آموزش مداوم و ارتباط منظم توانایی و انگیزه کار گروهی را بهبود می بخشد (سسکی و همکاران ، ۲۰۰۵). انگیزه تیم در توسعه نرم افزار چابک می تواند جزء مهم و موثر برای اطمینان از کار تیمی باشد.

۸-۴ فرهنگ سازمانی

یکی دیگر از ساختارهای اضافی در مدل کار گروهی مشخص شده از این مطالعه، فرهنگ سازمانی بود. در توسعه نرم افزار چابک، فرهنگ سازمانی باید تیم های خودمدیریتی و مسئولیت پذیری را تشویق کند. (Dybå & Dingsøyr, ۲۰۰۸). فرهنگ سازمانی خوب افراد را قادر می سازد تا کار خود را به گونه ای سازماندهی کنند که مالکیت مشترک و مسئولیت مشترک پروژه را در اختیار تیم توسعه قرار دهد (Dingsøyr et al., ۲۰۱۲). فرهنگ سازمانی خوب همچنین محیطی را برای کار تیمی موثر در تیم های چابک فراهم می کند. گودمن ، زاموتو و گیفورد (۲۰۰۱) دریافتند که فرهنگ یک سازمان می تواند کیفیت زندگی کاری را تحت تأثیر قرار دهد همانطور که بر نگرش ها، تعهد، انگیزه و رضایت اخلاقی افراد تأثیر می گذارد. زیرا عوامل انسانی بر کار گروهی تأثیر می گذارند. در مطالعه ای که توسط لئونارد و همکاران (۲۰۰۴) بر روی تیمی انجام شد، ذکر شد که فرهنگ سازمانی که در آن افراد برای انجام وظایف انگیزه دارند و می توانند در تیم صحبت کنند و نگرانی خود را بیان کنند، بر کار تیمی تأثیر مثبتی دارد (لئونارد، گراهام و بناکوم ، ۲۰۰۴). این نشان داد که فرهنگ سازمانی می تواند بر روابط بین تیم های توسعه نرم افزار Agile و مالک محصول تأثیر بگذارد.

هنگامی که مدل کار گروهی با یافته‌های این مطالعه ترسیم شد، دو عامل ارتباط و بازخورد، بر اساس موارد موجود در این مطالعه ترسیم شد. نگاشت عوامل مشترک از مطالعه تیم توسعه نرم‌افزار، مطالعه صاحب محصول و مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷) در شکل ۲۵ نشان داده شده است عوامل مانند اعتماد به نفس تیم، مشارکت تیم، مشارکت صاحب محصول و تیم پشتیبانی توضیحات مشابهی با نظارت، پشتیبانی، هماهنگی، رهبری تیم و جهت‌گیری تیم دارد. دو عامل، انگیزه تیمی و فرهنگ سازمانی، دو عامل، شباهت‌های کافی برای ترسیم با عوامل مدل کار گروهی نداشتند. بنابراین، این دو عامل به عنوان سازه‌های اضافی در یک مدل کار تیمی تجدید نظر شده پیشنهاد شد.

۹-۴ جمع بندی

در این فصل، یافته‌های حاصل از مطالعه تیم توسعه نرم‌افزار و مطالعه صاحب محصول گرد هم آمده و مورد بحث قرار گرفته است. هر یک از سوالات تحقیقاتی که این مطالعه بر اساس آن انجام شده بود با استفاده از یافته‌های این مطالعه پاسخ داده شد. از مدل کار گروهی (Dickinson & McIntyre, ۱۹۹۷) برای توضیح روابط تیم‌های توسعه نرم‌افزار Agile و بین تیم‌ها و صاحب محصول استفاده شد. نقشه‌برداری با مدل کار گروهی در این مطالعه انجام شد. دو ساختار دیگر از مدل کار تیمی از این مطالعه مشخص شد.

نتیجه گیری و جمع بندی پژوهش

بسیاری از پروژه‌های نرم‌افزاری با استفاده از رویکرد چابک انجام می‌شوند. چابک یک رویکرد رو به رشد و متداول برای توسعه نرم‌افزار است که میزان موفقیت آن بهتر از رویکرد برنامه‌ریزی شده است. توسعه نرم‌افزار چابک شامل یک رابطه قوی‌تر با مشتری، اغلب در نقش مالک محصول، نسبت به رویکرد برنامه‌ریزی شده است.

هنگام مرور مستندات، عوامل موفقیت، عوامل شکست، کار گروهی، روابط در توسعه نرم‌افزار چابک، مورد بررسی قرار گرفت. مشخص شد که تأثیر رابطه بین مالک محصول و تیم توسعه نرم‌افزار بر نتیجه آن است

پروژه‌های توسعه نرم‌افزار قبلاً مطالعه نشده بود. ممکن است به این دلیل باشد که وقتی پروژه‌های توسعه نرم‌افزار برنامه‌ریزی می‌شود، تأثیر رابطه بین مالک محصول و تیم توسعه نرم‌افزار بر نتیجه پروژه اغلب نادیده گرفته می‌شود یا مورد غفلت قرار می‌گیرد. عواملی مانند ارتباط و هماهنگی، نقش مهمی در ایجاد رابطه کاری خوب بین تیم‌های توسعه نرم‌افزار و صاحب محصول ایفا کردند.

این پایان نامه ضمن تمرکز بر رویکرد توسعه نرم افزار Agile، این ایده را بررسی می کند. هدف این تحقیق شناسایی چالش های توسعه نرم افزار Agile و تأثیرات آنها بر روابط بین مالک محصول و تیم توسعه نرم افزار بود. این تحقیق همچنین با هدف درک تأثیرات روابط بر نتیجه پروژه و شناسایی وسیله ای برای پرداختن به ماهیت روابط برای کمک به بهبود نتیجه پروژه انجام می شود. از مدل کار گروهی برای توضیح رابطه بین مالک محصول و تیم توسعه نرم افزار استفاده شد. سوالات تحقیق برای این تحقیق، همانطور که در بخش ۲،۹،۱ ارائه شده است، عبارتند از:

RQ1: اسناد نشان می دهد که چالش هایی بین تیم توسعه نرم افزار و نماینده مشتری در پروژه های توسعه نرم افزار چابک وجود دارد. این چالش ها تا چه حد وجود دارد؟

RQ2: چه نوع چالش هایی در پروژه های توسعه نرم افزار Agile وجود دارد؟

RQ3: تأثیر چالش ها بر رابطه بین مشتری و تیم توسعه نرم افزار Agile چیست؟

RQ4: آیا ماهیت رابطه بین مشتری و تیم توسعه بر نتیجه پروژه تأثیر می گذارد؟

RQ5: آیا مدل کار گروهی می تواند به طور کافی ماهیت رابطه بین مشتری و تیم توسعه را در زمینه توسعه چابک توضیح دهد؟

دو روش مطالعه ترکیبی از طرح تحقیق استفاده شد. یک مطالعه تجربیات و درک اعضای تیم توسعه نرم افزار را از طریق پرسشنامه و با نمونه فرعی از اعضا مصاحبه شده جمع آوری کرد. مطالعه بیشتر تجربیات و برداشت صاحبان محصولات را از طریق یک پرسشنامه کوتاه و مصاحبه های عمیق جمع آوری کرد. داده ها با استفاده از روش های آماری توصیفی و استنباطی و روش های تحلیل موضوعی، همانطور که در بخش ۳،۲ گزارش شده است، تجزیه و تحلیل شد.

یافته های پژوهش

RQ1: اسناد نشان می دهد که چالش هایی بین تیم توسعه نرم افزار و نماینده مشتری در پروژه های توسعه نرم افزار چابک وجود دارد. این چالش ها تا چه حد وجود دارد؟

هم تیم توسعه نرم افزار و هم مطالعه صاحبان محصولات این موضوع را تایید کردند که چالش های ارتباطی، چالش های بین فردی، اولویت های پروژه، به اشتراک گذاری ایده ها و زمان اتمام پروژه، در پروژه های توسعه نرم افزار چابک قابل مشاهده بود. میزان چنین چالش هایی در جدول ۴-۹ ارائه شده است.

RQ2: چه نوع چالش هایی در پروژه های توسعه نرم افزار Agile وجود دارد؟

در طول توسعه نرم افزار چابک بین اعضای تیم و مالک محصول مشارکت وجود داشت.

این مطالعه تأیید کرد که چالش هایی در طول فرایند توسعه نرم افزار وجود دارد

چالش های مشخص شده در این مطالعه عبارتند از:

- ارتباطات

- در دسترس بودن صاحب محصول

- انگیزه تیمی
- فرهنگ سازمانی
- چالش های بین فردی
- اختلاف نظر با مشتری در مورد اولویت ها و الزامات پروژه
- انگیزه تیمی
- پشتیبانی تیم

RQ3: تأثیر چالش ها بر رابطه بین مشتری و تیم توسعه نرم افزار Agile چیست؟

یافته های این مطالعه جاری نشان داد که چالش های ارتباطی، تعامل تیمی، مشارکت صاحب محصول، اعتماد به نفس و انگیزه تیمی، بر روابط بین تیم توسعه نرم افزار و صاحب محصول تأثیر می گذارد. این تحقیق نشان داده است که در برخی پروژه ها مشکلاتی در دریافت بازخورد، پشتیبانی تیم و تعامل بین اعضای تیم وجود دارد، بنابراین مشکلی در روابط تیم ایجاد شده است. این موضوع در قسمت ۶،۱ به تفصیل توضیح داده شده است.

RQ4: آیا ماهیت رابطه بین مشتری و تیم توسعه بر نتیجه پروژه تأثیر می گذارد؟

نتایج ارائه شده در بخش ۴،۴ تأثیر روابط بین مالک محصول و تیم های توسعه نرم افزار را نشان می دهد. این یافته ها تا حدودی به سوال تحقیق ۴ در این مطالعه پرداخته و نشان می دهد که ماهیت روابط بین تیم توسعه نرم افزار Agile و مشتری بر نتیجه پروژه تأثیر می گذارد. به عنوان مثال، نتایج این مطالعه نشان داد که بین زمان اتمام پروژه و اختلاف نظر با مشتری در مورد اولویت های پروژه رابطه خطی مثبت وجود دارد. $r = 0.329$ $n = 73$ $p = 0.004$.

این نشان می دهد که اگر اختلاف نظرهایی با صاحب محصول در مورد اولویت ها وجود داشته باشد، احتمالاً زمان اتمام پروژه را افزایش می دهد.

RQ5: آیا مدل کار گروهی می تواند به طور کافی ماهیت رابطه بین مشتری و تیم توسعه را در زمینه توسعه چابک توضیح دهد؟

روابط در تیم توسعه نرم افزار Agile با استفاده از مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷) مورد بررسی قرار گرفت. این مدل انتخاب شد زیرا سازه های این مدل مربوط به تیم توسعه نرم افزار Agile بود. با انجام این کار، این تحقیق عواملی را کشف کرد که قبلاً تصور نمی شد در این نوع مطالعه مناسب باشد و استحکام مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷) را آزمایش کرد.

هنگام نگاشت یافته های این مطالعه و مدل کار گروهی، برخی از عوامل مشخص شده در این مطالعه، مانند اعتماد به نفس تیم، و مالک محصول و تعامل تیمی، همپوشانی داشت. با این حال، مدل کار گروهی هیچ فرهنگ سازمانی یا انگیزه تیمی را الگو قرار نداد. نتایج این مطالعه نشان داد که مدل کار گروهی می تواند ماهیت روابط را توضیح

دهد. با این حال، برای کار گروهی موثر، همانطور که از این مطالعه مشخص شد، پیشنهاد شد که در مدل کار گروهی فعلی گنجانده شود. این در بخش ۶،۲ توضیح داده شده است.

مشارکت این تحقیق

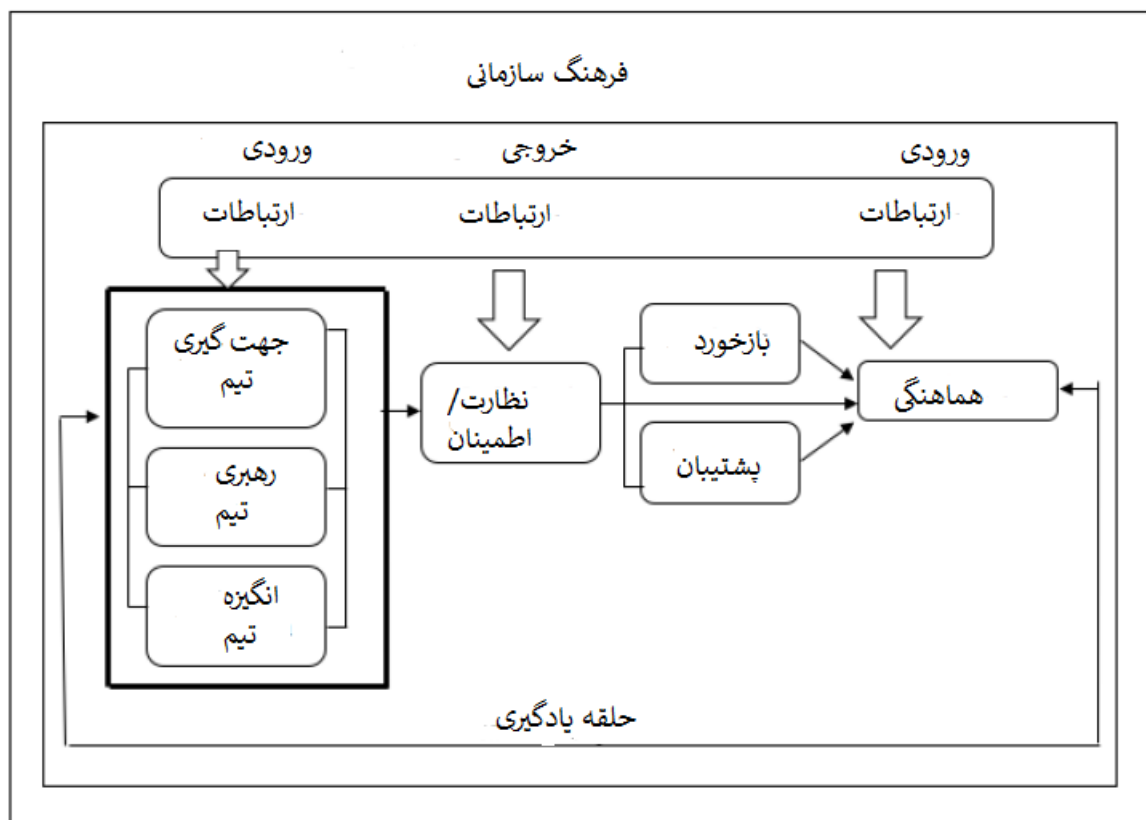
این مطالعه بر اساس رابطه بین مالک محصول و تیم توسعه نرم افزار بر نتیجه پروژه تأثیر داشت. این قبلاً گزارش نشده بود، و به طور خاص، به سوال تحقیق ۴ (RQ4) پاسخ داده شد.

در این تحقیق از مدل کار تیمی دیکنسون و مک اینتایر (۱۹۹۷) استفاده شده است که در مطالعات قبلی برای مطالعه تیم‌های توسعه نرم افزار Agile استفاده شده است. این تحقیق دو ساختار دیگر را برای مدل کار گروهی مشخص کرد: انگیزه تیمی و فرهنگ سازمانی. این ساختارهای اضافی برای کار تیمی موثر در تیم های توسعه نرم افزار Agile مهم هستند. انگیزه تیم بر هماهنگی، ارتباط و پشتیبانی تیم در پروژه تأثیر می‌گذارد. انگیزه تیم از کار گروهی منجر به توجه بیشتر کارکنان به کار خود و افزایش تعامل بیشتر در داخل تیم برای کمک و تولید بهره‌وری بیشتر می‌شود. عملکرد تیم نه تنها به شایستگی تیم بستگی دارد، بلکه به فرهنگ سازمانی و زمینه نیز بستگی دارد. فرهنگ سازمانی خوب افراد را قادر می‌سازد تا کار خود را به گونه‌ای سازماندهی کنند که به تیم توسعه مالکیت عمومی و مسئولیت مشترک پروژه را بدهد.

از این رو، فرهنگ سازمانی تأثیر کلی بر روابط بین تیم توسعه نرم افزار و مشتری در دستیابی به کار تیمی موثر دارد. مدل کار گروهی پیشنهادی این مطالعه را می‌توانید در شکل ۷،۱ مشاهده کنید.

صاحب محصول می‌تواند از سازمان مشتری یا داخلی سازمان توسعه نرم افزار باشد (در این مورد آنها را صاحبان محصولات پروکسی می‌نامند) (Kettunen, ۲۰۰۹). در این مطالعه، اکثر پاسخ دهندگان تیم توسعه نرم افزار (۶۵٪ ؛ ۸۹٪) ذکر کردند که در پروژه‌های خود صاحبان محصولات پروکسی دارند. این مطالعه تأیید کرد که صاحب محصول پروکسی در هدایت تحویل نهایی پروژه ها یا با ارائه ورودی، هماهنگی با تیم های توسعه، یا بررسی و تأیید محصول، مشارکت داشته است. این مطالعه همچنین تأیید کرده است که در توسعه نرم افزار Agile چالش‌های ارتباطی وجود دارد. این چالش‌ها شامل چالش‌های ارتباطی، اولویت‌های پروژه، الزامات پروژه و چالش‌های بین فردی بود. در مطالعه قبلی در زمینه توسعه نرم افزار چابک، مشخص شد که چالش‌های بین فردی بر نتیجه پروژه تأثیر داشته است (Liu et al., ۲۰۱۱).

با این حال، این مطالعه نمی‌تواند رابطه معنی‌داری بین چالش‌های بین فردی و نتیجه پروژه را تأیید کند.



شکل ۷-۱ مدل کار گروهی پیشنهادی

محدودیت‌های مطالعه

محدودیت‌های این مطالعه شامل تحقیقات کمی و کیفی بود. اول، موقعیت گسترده افراد حرفه‌ای در توسعه نرم افزار، تعیین محل شرکت‌ها با استفاده از رویکرد توسعه نرم افزار Agile را دشوار کرد. دوم، در مکان‌یابی مشتریان واقعی که در پروژه‌های توسعه نرم افزار Agile مشارکت داشتند، مشکلاتی وجود داشت. صاحبان محصولات پروکسی نمایندگان مشتری در بسیاری از توسعه نرم افزارهای Agile بودند، بنابراین ممکن است این بدان معنا باشد که اطلاعات جمع‌آوری شده از صاحبان محصولات پروکسی ممکن است منعکس کننده اطلاعات خود سازمان مشتری نباشد. نتایج حاصل از این مطالعه ممکن است نمایانگر همه محیط‌ها نباشد، ما رویکرد نرم افزاری Agile را که توسط یک تیم توسعه نرم افزار که در کنار هم قرار دارند استفاده می‌کنیم. بنابراین نتایج می‌تواند با یک نرم افزار سورس باز و پروژه‌هایی با تیم‌های توسعه نرم افزار مجازی یا از راه دور متفاوت باشد.

چه پیشنهاداتی را برای ادامه کار ارائه نموده است

نتایج این مطالعه تعدادی جهت برای کارهای آینده ارائه داد. ابتدا، این مطالعه چندین چالش موجود در تیم‌های توسعه نرم افزار Agile را برجسته کرد. بر این اساس، کارهای آینده باید بر شناسایی چالش‌هایی متمرکز شود که مشتری واقعی نقش مالک محصول را بر عهده داشته باشد. ثانیاً، کارهای آینده برای تأیید اثربخشی مدل کار تیمی

ارائه شده در این پایان نامه مفید خواهد بود. کارهای آینده نیز می‌تواند برای تأیید ساختارهای اضافی مشخص شده از این مطالعه انجام شود.

خلاصه

اسناد اهمیت روابط تیمی را در توسعه نرم‌افزار چابک تشخیص داده است. استفاده از رویکرد چابک برای توسعه نرم‌افزار به عنوان راهی برای افزایش احتمال نتایج موفق در پروژه‌های توسعه نرم‌افزار در نظر گرفته شد. تعدادی از عوامل بر ارتباط بین مالک محصول و تیم توسعه نرم‌افزار تأثیر گذاشتند و اینها بر نتیجه پروژه های Agile تأثیر گذاشتند.

این مطالعه چارچوبی را برای کمک به پزشکان و محققان برای درک و مدیریت کار گروهی موثر در پروژه‌های توسعه نرم‌افزار چابک فراهم کرد. این تحقیق تحقیقی تأثیرات مثبت و منفی عوامل مختلف مانند ارتباط، بازخورد، پشتیبانی تیم، اطمینان از اجرای رویکرد توسعه نرم‌افزار چابک و چارچوب نظری برای کار گروهی موثر را ارائه کرده است. این تحقیق همچنین به سازمان توسعه نرم‌افزار کمک کرد تا اهمیت انگیزه تیم و فرهنگ سازمانی را در تیم‌های توسعه نرم‌افزار چابک درک کند.

منابع

- Alchian, A. A., & Demsetz, H. (1972). Production, information costs, and economic organization. *The American Economic Review*, 62(5), 777-795.
- Alliance, A. (2001). Agile alliance. *Internet URL."* [www. agilealliance. com](http://www.agilealliance.com).
- Amabile, T. M., & Kramer, S. J. (2007). Inner work life. *Harvard Business Review*, 85(5), 72-83.
- Anderson, D. J. (2003). *Agile management for software engineering: Applying the theory of constraints for business results*: Prentice Hall Professional.

Arksey, H., & Knight, P. T. (1999). *Interviewing for social scientists: An introductory resource with examples*: Sage.

Atkinson, R., & Flint, J. (2001). Accessing hidden and hard-to-reach populations: Snowball research strategies. *Social research update*, 33(1), 1-4.

Aubert, B. A., Patry, M., & Rivard, S. (2005). A framework for information technology outsourcing risk management. *ACM SIGMIS Database*, 36(4), 9-28.

Awad, M. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia*.

Axelsson, K., Melin, U., & Lindgren, I. (2013). Public e-services for agency efficiency and citizen benefit — Findings from a stakeholder centered analysis. *Government Information Quarterly*, 30(1), 10-22. doi:10.1016/j.giq.2012.08.002

Bahli, B., & Rivard, S. (2003). The information technology outsourcing risk: a transaction cost and agency theory-based perspective. *Journal of Information Technology*, 18(3), 211-221.

Balijepally, V., Mahapatra, R., & Nerur, S. P. (2006). Assessing personality profiles of software developers in agile development teams. *Communications of the Association for Information Systems*, 18(1), 4.

Bass, J. M. (2015). How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering*, 20(6), 1525-1557.

Batra, D., Xia, W., VanderMeer, D., & Dutta, K. (2010). Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry. *Communications of the Association for Information Systems*, 27(1), 21.

Beck, K. (2000). *Extreme programming explained: embrace change*: Addison-Wesley Professional.

Beck, K., & Andres, C. (2004). *Extreme programming explained: embrace change*: Addison-Wesley Professional.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001a). The agile manifesto.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001b). Manifesto for agile software development.

Begel, A., & Nagappan, N. (2007a). Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study. 255-264. doi:10.1109/esem.2007.12

Begel, A., & Nagappan, N. (2007b). *Usage and perceptions of agile software development in an industrial context: An exploratory study*. Paper presented at the Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on.

190

Bell, D. E., Raiffa, H., & Tversky, A. (1988). *Decision making: Descriptive, normative, and prescriptive interactions*: Cambridge University Press.

Bengtsson, M., & Kock, S. (2014). Coopetition—Quo vadis? Past accomplishments and

future challenges. *Industrial Marketing Management*, 43(2), 180-188.

Bernard, H. R. (2011). *Research methods in anthropology: Qualitative and quantitative approaches*: Rowman Altamira.

Berscheid, E. (1999). The greening of relationship science. *American Psychologist*, 54(4), 260.

Biehl, J. T., Czerwinski, M., Smith, G., & Robertson, G. G. (2007). *FASTDash: a visual dashboard for fostering awareness in software teams*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64-69.

Boehm, B., & Turner, R. (2003a). *Balancing agility and discipline: A guide for the perplexed*: Addison-Wesley Professional.

Boehm, B., & Turner, R. (2003b). People factors in software management: lessons from comparing agile and plan-driven methods. *Crosstalk-The Journal of Defense Software Engineering*, (Dec 2003).

Boehm, B., & Turner, R. (2003c). Using risk to balance agile and plan-driven methods. *Computer*, 36(6), 57-66.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *Software, IEEE*, 22(5), 30-39.

Booch, G. (1986). Object-oriented development. *Software Engineering, IEEE Transactions on*(2), 211-221.

Boudreau, M.-C., Gefen, D., & Straub, D. W. (2001). Validation in information systems research: a state-of-the-art assessment. *MIS Quarterly*, 1-16.

Brannick, M. T., Salas, E., & Prince, C. W. (1997). *Team performance assessment and measurement: Theory, methods, and applications*: Psychology Press.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77-101.

Brown, A. D., & Starkey, K. (1994). The effect of organizational culture on communication and information. *Journal of Management studies*, 31(6), 807-828.

Bryman, A. (2012). *Social research methods*: Oxford university press.

Bustard, D., Wilkie, G., & Greer, D. (2013). *The maturation of agile software development principles and practice: Observations on successive industrial studies in 2010 and 2012*. Paper presented at the Engineering of Computer Based Systems (ECBS), 2013 20th IEEE International Conference and Workshops on the.

Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *Software, IEEE*, 25(1), 60-67.

Ceschi, M., Sillitti, A., Succi, G., & De Panfilis, S. (2005). Project management in plan-based and agile companies. *Software, IEEE*, 22(3), 21-27.

Chatman, J. A., & Flynn, F. J. (2001). The influence of demographic heterogeneity on the emergence and consequences of cooperative norms in work teams. *Academy of management journal*, 44(5), 956-974.

Chen, I. J., & Popovich, K. (2003). Understanding customer relationship management (CRM) People, process and technology. *Business process management journal*, 9(5), 672-688.

Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961-971.
doi:10.1016/j.jss.2007.08.020

Cochran, W. G. (2007). *Sampling techniques*: John Wiley & Sons. 191

Cockburn, A. (2006). *Agile software development: the cooperative game*: Pearson Education.

Cockburn, A., & Highsmith, J. (2001a). Agile software development, the people factor. *Computer*, 34(11), 131-133.

Cockburn, A., & Highsmith, J. (2001b). Agile software development: The people factor. *Computer*(11), 131-133.

Cohen, D., & Crabtree, B. (2006). Qualitative research guidelines project.

Cohen, S. G., & Bailey, D. E. (1997). What makes teams work: Group effectiveness research from the shop floor to the executive suite. *Journal of management*, 23(3), 239-290.

Cohn, M. (2004). *User stories applied: For agile software development*: Addison-Wesley Professional.

Conboy, K., Coyle, S., Wang, X., & Pikkarainen, M. (2010). People over process: key people challenges in agile development. *IEEE software*, 99(1), 47-57.

Conboy, K., & Fitzgerald, B. (2010). Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 20(1), 2.

Coram, M., & Bohner, S. (2005). *The impact of agile methods on software project management*. Paper presented at the Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the.

Cramton, C. D., & Webber, S. S. (2005). Relationships among geographic dispersion, team processes, and effectiveness in software development work teams. *Journal of Business Research*, 58(6), 758-765.

Creswell, J. (2011). Controversies in mixed methods research. *The Sage handbook of qualitative research*, 4, 269-284.

Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*: Sage publications.

Cuevas-Rodríguez, G., Gomez-Mejia, L. R., & Wiseman, R. M. (2012). Has Agency Theory Run its Course?: Making the Theory more Flexible to Inform the Management of Reward Systems. *Corporate Governance: An International Review*, 20(6), 526-546.
doi:10.1111/corg.12004

Daft, R. L., Lengel, R. H., & Trevino, L. K. (1987). Message equivocality, media selection, and manager performance: Implications for information systems. *MIS Quarterly*, 355-366.

Dahlin, K. B., Weingart, L. R., & Hinds, P. J. (2005). Team diversity and information use. *Academy of management journal*, 48(6), 1107-1123.

Davies, D., & Dodd, J. (2002). Qualitative research and the question of rigor. *Qualitative health research*, 12(2), 279-289.

Davis, J. H., Schoorman, F. D., & Donaldson, L. (1997). Toward a stewardship theory of management. *Academy of management review*, 22(1), 20-47.

De Dreu, C. K., & Weingart, L. R. (2003). Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis. *Journal of applied Psychology*, 88(4), 741.

DeChurch, L. A., & Mesmer-Magnus, J. R. (2010). The cognitive underpinnings of effective teamwork: a meta-analysis: American Psychological Association.

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010). The scrum primer. *Scrum Primer is an in-depth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at: <http://assets.scrumtraininginstitute.com/downloads/1/scrumprimer121.pdf>, 1285931497, 15.*

DiCicco-Bloom, B., & Crabtree, B. F. (2006). The qualitative research interview. *Medical education*, 40(4), 314-321.

192

Dickinson, T. L., & McIntyre, R. M. (1997). A conceptual framework for teamwork measurement. *Team performance assessment and measurement*, 19-43.

Dillon, S. M. (1998). Descriptive decision making: Comparing theory with practice. *Unpublished manuscript, Department of Management Systems, University of Waikato, New Zealand.*

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012a). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012b). A decade of agile methodologies: Towards explaining agile software development: Elsevier.

Donaldson, L., & Davis, J. H. (1991). Stewardship theory or agency theory: CEO governance and shareholder returns. *Australian Journal of management*, 16(1), 49-64.

Drury, M., Conboy, K., & Power, K. (2012). Obstacles to decision making in Agile software development teams. *Journal of Systems and Software*, 85(6), 1239-1254.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), 833-859. doi:10.1016/j.infsof.2008.01.006

Dybå, T., Dingsøyr, T., & Moe, N. B. (2014). Agile project management *Software project management in a changing world* (pp. 277-300): Springer.

Eby, L. T., Rhodes, J. E., & Allen, T. D. (2007). Definition and evolution of mentoring. *The*

Blackwell handbook of mentoring: A multiple perspectives approach, 7-20.

Eckfeldt, B., Madden, R., & Horowitz, J. (2005). *Selling agile: target-cost contracts*. Paper presented at the Agile Conference, 2005. Proceedings.

Eisenhardt, K. M. (1989). Agency theory: An assessment and review. *Academy of management review*, 14(1), 57-74.

Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. *Management Science*, 46(12), 1554-1568.

Fenton, N., & Bieman, J. (2014). *Software metrics: a rigorous and practical approach*: CRC Press.

Ferran, C., & Watts, S. (2008). Videoconferencing in the field: A heuristic processing model. *Management Science*, 54(9), 1565-1578.

Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, 15(2), 200-213.

Fitzgerald, B., Russo, N. L., & O'Kane, T. (2003). Software development method tailoring at Motorola. *Communications of the ACM*, 46(4), 64-70.

Fornell, C. (1992). A national customer satisfaction barometer: the Swedish experience. *the Journal of Marketing*, 6-21.

Fowler, M. (2001). The new methodology. *Wuhan University Journal of Natural Sciences*, 6(1-2), 12-24.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, 9(8), 28-35.

Franklin, T. (2008). *Adventures in agile contracting: Evolving from time and materials to fixed price, fixed scope contracts*. Paper presented at the Agile, 2008. AGILE'08. Conference.

Fuggetta, A., & Di Nitto, E. (2014). *Software process*. Paper presented at the Proceedings of the on Future of Software Engineering.

Gabarro, J. J. (1990). The development of working relationships. *Intellectual teamwork: Social and technological foundations of cooperative work*, 79-110.

Garbarino, E., & Johnson, M. S. (1999). The different roles of satisfaction, trust, and commitment in customer relationships. *the Journal of Marketing*, 70-87. 193

Gefen, D., & Straub, D. W. (2004). Consumer trust in B2C e-Commerce and the importance of social presence: experiments in e-Products and e-Services. *Omega*, 32(6), 407-424.

Ghazawneh, A., & Henfridsson, O. (2013). Balancing platform control and external contribution in third-party development: the boundary resources model. *Information systems journal*, 23(2), 173-192.

Ghimire, D., Gibbs, S., & Charters, S. Software Development Team Views of Success Factors in Agile projects.

Ghobadi, S., & Mathiassen, L. (2016). Perceived barriers to effective knowledge sharing in agile software teams. *Information systems journal*, 26(2), 95-125.

Golafshani, N. (2003). Understanding reliability and validity in qualitative research. *The qualitative report*, 8(4), 597-606.

Gollenia, L. A. (2016). *Business Transformation Management Methodology*: Routledge.

Gopal, A., & Sivaramakrishnan, K. (2008). Research Note-On Vendor Preferences for Contract Types in Offshore Software Projects: The Case of Fixed Price vs. Time and Materials Contracts. *Information Systems Research*, 19(2), 202-220.

Gordon, J., & Hartman, R. L. (2009). Affinity-seeking strategies and open communication in peer workplace relationships. *Atlantic Journal of Communication*, 17(3), 115-125.

Greenhaus, J. H., Collins, K. M., & Shaw, J. D. (2003). The relation between work-family balance and quality of life. *Journal of vocational behavior*, 63(3), 510-531.

Greer, D., & Hamon, Y. (2011). Agile Software Development. *Software: Practice and Experience*, 41(9), 943-944.

Grisham, P. S., & Perry, D. E. (2005). *Customer relationships and extreme programming*. Paper presented at the ACM SIGSOFT Software Engineering Notes.

Groves, R. M., Fowler Jr, F. J., Couper, M. P., Lepkowski, J. M., Singer, E., & Tourangeau, R. (2011). *Survey methodology* (Vol. 561): John Wiley & Sons.

Guba, E. G., & Lincoln, Y. S. (1994). Competing paradigms in qualitative research. *Handbook of qualitative research*, 2(163-194), 105.

Guzzo, R. A., & Dickson, M. W. (1996). Teams in organizations: Recent research on performance and effectiveness. *Annual review of psychology*, 47(1), 307-338.

Guzzo, R. A., & Shea, G. P. (1992). Group performance and intergroup relations in organizations. *Handbook of industrial and organizational psychology*, 3, 269-313.

Hackman, J. (1987). The design of work teams. Inj. w. lorsch (ed.), *Handbook of organizational behavior* (pp. 315-342): englewood cliffs, nj: prentice hall.

Hannafey, F. T., & Vitulano, L. A. (2013). Ethics and Executive Coaching: An Agency Theory Approach. *Journal of business ethics*, 115(3), 599-603.

Hannay, J. E., & Benestad, H. C. (2010). *Perceived productivity threats in large agile development projects*. Paper presented at the Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement.

Harrison, M., & Norman, C. (2001). What trust means in e-commerce customer relationships: an interdisciplinary conceptual typology. *International journal of electronic commerce*, 6(2), 35-59.

Harter, J. K., Schmidt, F. L., & Hayes, T. L. (2002). Business-unit-level relationship between employee satisfaction, employee engagement, and business outcomes: a metaanalysis. *Journal of applied Psychology*, 87(2), 268.

Haynes, S. N., Richard, D., & Kubany, E. S. (1995). Content validity in psychological assessment: A functional approach to concepts and methods. *Psychological assessment*, 7(3), 238.

Henttonen, K., & Blomqvist, K. (2005). Managing distance in a global virtual team: the evolution of trust through technology-mediated relational communication. *Strategic Change*, 14(2), 107-119.

- Herrnstein, R. J., & Murray, C. (2010). *Bell curve: Intelligence and class structure in American life*: Simon and Schuster.
- Highsmith, J. (2002). *Agile software development ecosystems* (Vol. 13): Addison-Wesley Professional.
- Highsmith, J. (2009). *Agile project management: creating innovative products*: Pearson Education.
- Highsmith, J. (2013). *Adaptive software development: a collaborative approach to managing complex systems*: Addison-Wesley.
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-127.
- Hill, R. (1998). What sample size is "enough" in internet survey research. *Interpersonal Computing and Technology: An electronic journal for the 21st century*, 6(3-4), 1-12.
- Hirsch, P., Michaels, S., & Friedman, R. (1987). Dirty hands" versus "clean models. *Theory and society*, 16(3), 317-336.
- Hochmüller, E. (2011). *The requirements engineer as a liaison officer in agile software development*. Paper presented at the Proceedings of the 1st workshop on agile requirements engineering.
- Hoda, R., Noble, J., & Marshall, S. (2010). *Organizing self-organizing teams*. Paper presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1.
- Hoda, R., Noble, J., & Marshall, S. (2011). The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology*, 53(5), 521-534. doi:<http://dx.doi.org/10.1016/j.infsof.2010.10.009>
- Hoda, R., Noble, J., & Marshall, S. (2013). Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*, 39(3), 422-444.
- Hoegl, M., & Gemuenden, H. G. (2001). Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. *Organization Science*, 12(4), 435-449.
- Hoegl, M., & Parboteeah, P. (2006). Autonomy and teamwork in innovative projects. *Human Resource Management*, 45(1), 67-79.
- Hole, S., & Moe, N. B. (2008). *A case study of coordination in distributed agile software development*. Paper presented at the European Conference on Software process improvement.
- Holmstrom, H., Conchúir, E. Ó., Agerfalk, J., & Fitzgerald, B. (2006). *Global software development challenges: A case study on temporal, geographical and socio-cultural distance*. Paper presented at the Global Software Engineering, 2006. ICGSE'06. International Conference on.
- Hossain, E., Babar, M. A., & Paik, H.-y. (2009). *Using scrum in global software development: a systematic literature review*. Paper presented at the Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on.

Hossain, E., Babar, M. A., Paik, H.-y., & Verner, J. (2009). *Risk identification and mitigation processes for using scrum in global software development: A conceptual framework*. Paper presented at the Software Engineering Conference, 2009. APSEC'09. Asia-Pacific.

Huda, M. N. (2011). *Comparison between adaptive software development and feature driven development*. Paper presented at the Computer Science and Network Technology (ICCSNT), 2011 International Conference on.

Hummel, M., & Epp, A. (2015). *Success factors of agile information systems development: A qualitative study*. Paper presented at the System Sciences (HICSS), 2015 48th Hawaii International Conference on.

195

Hummel, M., Rosenkranz, C., & Holten, R. (2013). The role of communication in agile systems development. *Business & Information Systems Engineering*, 5(5), 343-355.

Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004). *Software quality and agile methods*. Paper presented at the Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International.

Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., & Booch, G. (1999). *The unified software development process* (Vol. 1): Addison-wesley Reading.

Jakobsen, C. R., & Johnson, K. A. (2008). *Mature Agile with a Twist of CMMI*. Paper presented at the Agile, 2008. AGILE'08. Conference.

Janzen, D. S., & Saiedian, H. (2005). Test-driven development: Concepts, taxonomy, and future direction. *Computer*, 38(9), 43-50.

Jeong, H., Mason, S. P., Barabási, A.-L., & Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, 411(6833), 41-42.

Johnson, B., & Onwuegbuzie, A. (2004a). Mixed Methods Research: A Research Paradigm Whose Time Has Come. *EDUCATIONAL RESEARCHER*, 33: 14. doi:10.3102/0013189X033007014

Johnson, R. B., & Onwuegbuzie, A. J. (2004b). Mixed methods research: A research paradigm whose time has come. *EDUCATIONAL RESEARCHER*, 33(7), 14-26.

Kahkonen, T. (2004). *Agile methods for large organizations-building communities of practice*. Paper presented at the Agile Development Conference, 2004.

Katzenbach, J. R., & Smith, D. K. (2005). The discipline of teams. *Harvard Business Review*, 83(7), 162.

Kaur, R., & Sengupta, J. (2013). Software process models and analysis on failure of software development projects. *arXiv preprint arXiv:1306.1068*.

Keegan, A., & Turner, J. R. (2001). Quantity versus quality in project-based learning practices. *Management learning*, 32(1), 77-98.

Kettunen, P. (2009). Adopting key lessons from agile manufacturing to agile software product development—A comparative study. *Technovation*, 29(6), 408-422.

Khalid, A., Zahra, S., & Khan, M. F. (2014). Suitability and Contribution of Agile Methods

in Mobile Software Development. *International Journal of Modern Education and Computer Science*, 6(2), 56.

Kitto, S. C., Chesters, J., & Grbich, C. (2008). Quality in qualitative research. *Medical Journal of Australia*, 188(4), 243.

Kniberg, H. (2010). *Kanban and Scrum-making the most of both*: Lulu. com.

Koh, C., Ang, S., & Straub, D. W. (2004). IT outsourcing success: A psychological contract perspective. *Information Systems Research*, 15(4), 356-373.

Korkala, M., Abrahamsson, P., & Kyllönen, P. (2006). *A case study on the impact of customer communication on defects in agile software development*. Paper presented at the Agile Conference, 2006.

Kothari, C. R. (2004). *Research methodology: Methods and techniques*: New Age International.

Lalsing, V., Kishnah, S., & Pudaruth, S. (2012). People factors in agile software development and project management. *International Journal of Software Engineering & Applications*, 3(1), 117.

Langfred, C. W. (2000). The paradox of self-management: Individual and group autonomy in work groups. *Journal of Organizational Behavior*, 563-585.

Langfred, C. W. (2004). Too much of a good thing? Negative effects of high trust and individual autonomy in self-managing teams. *Academy of management journal*, 47(3), 385-399.

196

Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer*, 36(6), 47-56.

Lawshe, C. H. (1975). A quantitative approach to content validity¹. *Personnel psychology*, 28(4), 563-575.

Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012a). *Software Development Life Cycle AGILE vs Traditional Approaches*. Paper presented at the 2012 International Conference on Information and Network Technology (ICINT 2012) IPCSIT.

Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012b). *Software development life cycle AGILE vs traditional approaches*. Paper presented at the International Conference on Information and Network Technology.

Lee, A. S. (1991). Integrating positivist and interpretive approaches to organizational research. *Organization Science*, 2(4), 342-365.

Lee, G., & Xia, W. (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34(1), 87-114.

Lee Rodgers, J., & Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1), 59-66.

Leffingwell, D. (2007). Scaling software agility; best practices for large enterprises. . 31(2, p. 0).

Leonard, M., Graham, S., & Bonacum, D. (2004). The human factor: the critical importance of effective teamwork and communication in providing safe care. *Quality and Safety in Health Care*, 13(suppl 1), i85-i90.

Lin, J. (2015). Human Factors in Agile Software Development. *arXiv preprint arXiv:1502.04170*.

Lindsjörn, Y., Sjøberg, D. I., Dingsøyr, T., Bergersen, G. R., & Dybå, T. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software*, 122, 274-286.

Lindstrom, L., & Jeffries, R. (2004). Extreme programming and agile software development methodologies. *Information systems management*, 21(3), 41-52.

Liu, J. Y.-C., Chen, H.-G., Chen, C. C., & Sheu, T. S. (2011). Relationships among interpersonal conflict, requirements uncertainty, and software project performance. *International Journal of Project Management*, 29(5), 547-556. doi:10.1016/j.ijproman.2010.04.007

Lovelace, K., Shapiro, D. L., & Weingart, L. R. (2001). Maximizing cross-functional new product teams' innovativeness and constraint adherence: A conflict communications perspective. *Academy of management journal*, 44(4), 779-793.

Lu, Y., Xiang, C., Wang, B., & Wang, X. (2011). What affects information systems development team performance? An exploratory study from the perspective of combined socio-technical theory and coordination theory. *Computers in Human Behavior*, 27(2), 811-822.

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS-a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing*, 10(4), 325-337.

Lynn, M. R. (1986). Determination and quantification of content validity. *Nursing research*, 35(6), 382-386.

Mackenzie, N., & Knipe, S. (2006). Research dilemmas: Paradigms, methods and methodology. *Issues in educational research*, 16(2), 193-205.

Mahaney, R. C., & Lederer, A. L. (2003). Information systems project management: an agency theory interpretation. *Journal of Systems and Software*, 68(1), 1-9.

Mahnic, V. (2012). A capstone course on agile software development using Scrum. *IEEE Transactions on Education*, 55(1), 99-106. 197

Malone, T. W., & Crowston, K. (1990). *What is coordination theory and how can it help design cooperative work systems?* Paper presented at the Proceedings of the 1990 ACM conference on Computer-supported cooperative work.

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR)*, 26(1), 87-119.

Marks, M. A., Mathieu, J. E., & Zaccaro, S. J. (2001). A temporally based framework and taxonomy of team processes. *Academy of management review*, 26(3), 356-376.

Martin, F. (2005). The New Methodology.

- Martin, R. (2003). *Agile software development: principles, patterns, and practices*: Prentice Hall PTR.
- Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, 20(3), 377-399.
- Matveev, A. V., & Nelson, P. E. (2004). Cross cultural communication competence and multicultural team performance: Perceptions of American and Russian managers. *International Journal of Cross Cultural Management*, 4(2), 253-270.
- McHugh, O., Conboy, K., & Lang, M. (2011). Using agile practices to influence motivation within it project teams.
- McHugh, O., Conboy, K., & Lang, M. (2012). Agile practices: The impact on trust in software project teams. *IEEE software*, 29(3), 71-76.
- Melnik, G., & Maurer, F. (2004). *Direct verbal communication as a catalyst of agile knowledge sharing*. Paper presented at the Agile Development Conference, 2004.
- Melton, H. L., & Hartline, M. D. (2010). Customer and frontline employee influence on new service development performance. *Journal of Service Research*, 13(4), 411-425.
- Meredith, J. R., & Mantel Jr, S. J. (2011). *Project management: a managerial approach*: John Wiley & Sons.
- Miller, D., & Breton-Miller, L. (2006). Family governance and firm performance: Agency, stewardship, and capabilities. *Family Business Review*, 19(1), 73-87.
- Mishra, D., Mishra, A., & Ostrovska, S. (2012). Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation. *Information and Software Technology*, 54(10), 1067-1078.
- Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82(11), 1869-1890.
- Mittal, V., & Kamakura, W. A. (2001). Satisfaction, repurchase intent, and repurchase behavior: Investigating the moderating effect of customer characteristics. *Journal of marketing research*, 38(1), 131-142.
- Moe, N. B., Dingsøyr, T., & Dybå, T. (2008). *Understanding self-organizing teams in agile software development*. Paper presented at the Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on.
- Moe, N. B., Dingsøyr, T., & Dybå, T. (2009). Overcoming barriers to self-management in software teams. *IEEE software*, 26(6).
- Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology*, 52(5), 480-491.
- Mohrman, S. A., Cohen, S. G., & Morhman Jr, A. M. (1995). *Designing team-based organizations: New forms for knowledge work*: Jossey-Bass.
- Moniruzzaman, A., & Hossain, D. S. A. (2013). Comparative Study on Agile software development methodologies. *arXiv preprint arXiv:1307.3356*.

- Morgan, G., & Ramirez, R. (1984). Action learning: A holographic metaphor for guiding social change. *Human relations*, 37(1), 1-27.
- Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *IJCSI*, 5, 95-101.
- Muth, M. M., & Donaldson, L. (1998). Stewardship theory and board structure: A contingency approach. *Corporate governance*, 6(1), 5-28.
- Natour, A., Kiridena, S., & Gibson, P. (2011). Supply chain integration and collaboration for performance improvement: an agency theory approach.
- Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, 50(3), 79-83.
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72-78.
- Neuman, W. L. (2005). *Social research methods: Quantitative and qualitative approaches* (Vol. 13): Allyn and Bacon Boston.
- Noble, H., & Smith, J. (2015). Issues of validity and reliability in qualitative research. *Evidence Based Nursing*, 18(2), 34-35.
- Noy, C. (2008). Sampling knowledge: The hermeneutics of snowball sampling in qualitative research. *International Journal of Social Research Methodology*, 11(4), 327-344.
- Nuseibeh, B. (2001). Weaving together requirements and architectures. *Computer*, 34(3), 115-119.
- Olsson, H. H., Alahyari, H., & Bosch, J. (2012). *Climbing the "Stairway to Heaven"--A Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software*. Paper presented at the Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on.
- Opelt, A., Gloger, B., Pfarl, W., & Mittermayr, R. (2013). *Agile Contracts: Creating and Managing Successful Projects with Scrum*: John Wiley & Sons.
- Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2008). Using scrum in a globally distributed project: a case study. *Software Process: Improvement and Practice*, 13(6), 527-544.
- Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2009). *Using scrum in distributed agile development: A multiple case study*. Paper presented at the Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on.
- Paasivaara, M., Heikkilä, V. T., & Lassenius, C. (2012). *Experiences in scaling the product owner role in large-scale globally distributed scrum*. Paper presented at the Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on.
- Paetsch, F., Eberlein, A., & Maurer, F. (2003a). *Requirements engineering and agile software development*. Paper presented at the Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on.
- Paetsch, F., Eberlein, A., & Maurer, F. (2003b). *Requirements engineering and agile software development*. Paper presented at the null.

Paetsch, F., Eberlein, A., & Maurer, F. (2003c). *Requirements Engineering and Agile Software Development*. Paper presented at the WETICE.

Palmer, S. R., & Felsing, M. (2001). *A practical guide to feature-driven development*: Pearson Education.

Parnas, D. (2006). Agile methods and GSD: The wrong solution to an old but real problem. *Communications of the ACM*, 49(10), 29-29.

Patton, M. Q. (2005). *Qualitative research*: Wiley Online Library. 199

Perry, W. E. (2007). *Effective Methods for Software Testing: Includes Complete Guidelines, Checklists, and Templates*: John Wiley & Sons.

Petersen, K., & Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering*, 15(6), 654-693.

Phalnikar, R., Deshpande, V., & Joshi, S. (2009). *Applying agile principles for distributed software development*. Paper presented at the Advanced Computer Control, 2009. ICACC'09. International Conference on.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3), 303-337.

Polit, D. F., Beck, C. T., & Owen, S. V. (2007). Is the CVI an acceptable indicator of content validity? Appraisal and recommendations. *Research in nursing & health*, 30(4), 459-467.

Poppendieck, M., & Poppendieck, T. (2003a). *Lean software development: An agile toolkit*: Addison-Wesley Professional.

Poppendieck, M., & Poppendieck, T. (2003b). *Lean Software Development: An Agile Toolkit*: Addison-Wesley.

Poppendieck, M., & Poppendieck, T. (2007). *Implementing lean software development: from concept to cash*: Pearson Education.

Power, K. (2010). *Stakeholder identification in agile software product development organizations: A model for understanding who and what really counts*. Paper presented at the AGILE Conference, 2010.

Pressman, R. S. (2005). *Software engineering: a practitioner's approach*: Palgrave Macmillan.

Pym, D., & Sadler, M. (2010). Information Stewardship in cloud computing. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, 1(1), 50-67.

Rai, A., Maruping, L. M., & Venkatesh, V. (2009). Offshore information systems project success: the role of social embeddedness and cultural characteristics. *MIS Quarterly*, 617-641.

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information systems journal*, 20(5), 449-480.

Ramesh, B., Cao, L., Mohan, K., & Xu, P. (2006). Can distributed software development be agile? *Communications of the ACM*, 49(10), 41-46.

Reh, F. J. (2005). Pareto's principle-The 80-20 rule. *BUSINESS CREDIT-NEW YORK THEN COLUMBIA MD-*, 107(7), 76.

Rising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software*, 17(4), 26.

Ross, S. A. (1973). AssociationThe Economic Theory of Agency: The Principal's Problem. *The American Economic Re*, 63(2), 134-139.

Royce, W. W. (1970). *Managing the development of large software systems*. Paper presented at the proceedings of IEEE WESCON.

Ryan, S., & O'Connor, R. V. (2013). Acquiring and sharing tacit knowledge in software development teams: An empirical study. *Information and Software Technology*, 55(9), 1614-1624.

Salas, E., Cooke, N. J., & Rosen, M. A. (2008). On teams, teamwork, and team performance: Discoveries and developments. *Human factors*, 50(3), 540-547.

Salas, E., Sims, D. E., & Burke, C. S. (2005). Is there a "Big Five" in teamwork? *Small group research*, 36(5), 555-599.

200

Santos, V., Goldman, A., & De Souza, C. R. (2015). Fostering effective inter-team knowledge sharing in agile software development. *Empirical Software Engineering*, 20(4), 1006-1051.

Sarantakos, S. (2012). *Social research*: Palgrave Macmillan.

Schatz, B., & Abdelshafi, I. (2005). Primavera gets agile: a successful transition to agile development. *IEEE software*(3), 36-42.

Schillemans, T. (2013). Moving Beyond The Clash of Interests: On stewardship theory and the relationships between central government departments and public agencies. *Public Management Review*, 15(4), 541-562.

Schwaber, K. (1997). Scrum development process *Business Object Design and Implementation* (pp. 117-134): Springer.

Schwaber, K. (2004). *Agile project management with Scrum*: Microsoft Press.

Schwaber, K., & Beedle, M. (2002). *Scrum: The definitive guide to Scrum*.
 Scrum Software Development with Scrum.

Schwaber, K., & Sutherland, J. (2013). The definitive guide to scrum: The rules of the game. *Improving the Profession of Software development*.

Seale, C. (1999). Quality in qualitative research. *Qualitative inquiry*, 5(4), 465-478.

Shank, G., & Brown, L. (2013). *Exploring educational research literacy*: Routledge.

Sheehan, K. B. (2001). E-mail survey response rates: A review. *Journal of Computer - Mediated Communication*, 6(2), 0-0.

Sheng, C.-W., Tian, Y.-F., & Chen, M.-C. (2010). Relationships among teamwork behavior, trust, perceived team support, and team commitment. *Social Behavior and Personality: an international journal*, 38(10), 1297-1305.

Shore, J. (2007). *The art of agile development*: " O'Reilly Media, Inc."

Sias, P. M. (2005). Workplace relationship quality and employee information experiences. *Communication studies*, 56(4), 375-395.

Smith, H., & Fingar, P. (2003). *Business process management: the third wave* (Vol. 1): Meghan-Kiffer Press Tampa.

Sohaib, O., & Khan, K. (2010). *Integrating usability engineering and agile software development: A literature review*. Paper presented at the Computer design and applications (ICCD), 2010 international conference on.

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B. (2013a). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*.

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B. (2013b). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*, 86(6), 1663-1678. doi:10.1016/j.jss.2013.02.027

Stapleton, J. (1997). *DSDM, dynamic systems development method: the method in practice*: Cambridge University Press.

Stenbacka, C. (2001). Qualitative research requires quality concepts of its own. *Management decision*, 39(7), 551-556.

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica*, 17(4), 64.

Stol, K.-J., & Fitzgerald, B. (2014). *Two's company, three's a crowd: a case study of crowdsourcing software development*. Paper presented at the Proceedings of the 36th International Conference on Software Engineering.

Strauss, A., & Corbin, J. (1990). *Basics of qualitative research* (Vol. 15): Newbury Park, CA: Sage.

Sureshchandra, K., & Shrinivasavadhani, J. (2008). *Adopting agile in distributed development*. Paper presented at the Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on. 201

Sutherland, J., Jakobsen, C. R., & Johnson, K. (2008). *Scrum and CMMI level 5: The magic potion for code warriors*. Paper presented at the Hawaii International Conference on System Sciences, Proceedings of the 41st Annual.

Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). *Distributed scrum: Agile project management with outsourced development teams*. Paper presented at the System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on.

Svensson, H., & Höst, M. (2005). *Views from an organization on how agile development affects its collaboration with a software development team*. Paper presented at the

International Conference on Product Focused Software Process Improvement.

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard Business Review*, 64(1), 137-146.

Tan, C.-W., Pan, S. L., & Lim, E. T. (2005). Managing stakeholder interests in e-government implementation: Lessons learned from a Singapore e-government project. *Journal of Global Information Management (JGIM)*, 13(1), 31-53.

Tate, K. (2005). *Sustainable software development: an agile perspective*: Addison-Wesley Professional.

Taylor, R. (1990). Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography*, 6(1), 35-39.

Topp, L., Barker, B., & Degenhardt, L. (2004). The external validity of results derived from ecstasy users recruited using purposive sampling strategies. *Drug and alcohol dependence*, 73(1), 33-40.

Tuckman, B. W. (1965). Developmental sequence in small groups. *Psychological bulletin*, 63(6), 384.

Turk, D., France, R., & Rumpe, B. (2014a). Assumptions underlying agile software development processes. *arXiv preprint arXiv:1409.6610*.

Turk, D., France, R., & Rumpe, B. (2014b). Limitations of agile software processes. *arXiv preprint arXiv:1409.6600*.

Utting, M., & Legeard, B. (2010). *Practical model-based testing: a tools approach*: Morgan Kaufmann.

Van Lamsweerde, A. (2000). *Requirements engineering in the year 00: A research perspective*. Paper presented at the Proceedings of the 22nd international conference on Software engineering.

Van Selm, M., & Jankowski, N. W. (2006). Conducting online surveys. *Quality and Quantity*, 40(3), 435-456.

Van Teijlingen, E., & Hundley, V. (2002). The importance of pilot studies. *Nursing standard*, 16(40), 33-36.

Vidgen, R., & Wang, X. (2009). Coevolving systems and the organization of agile software development. *Information Systems Research*, 20(3), 355-376.

Vijayasarathy, L., & Turk, D. (2008). Agile software development: A survey of early adopters. *Journal of Information Technology Management*, 19(2), 1-8.

Voss, C., Tsikriktsis, N., & Frohlich, M. (2002). Case research in operations management. *International journal of operations & production management*, 22(2), 195-219.

West, D., Grant, T., Gerush, M., & D'silva, D. (2010). Agile development: Mainstream adoption has changed agility. *Forrester Research*, 2(1), 41.

Whitworth, E., & Biddle, R. (2007). *The social nature of agile teams*. Paper presented at the Agile conference (AGILE), 2007.

Wiegers, K., & Beatty, J. (2013). *Software requirements*: Pearson Education.

Wilding, R., Wagner, B., Fayezi, S., O'Loughlin, A., & Zutshi, A. (2012). Agency theory and

supply chain management: a structured literature review. *Supply chain management: an international journal*, 17(5), 556-570.

Williams, L., & Cockburn, A. (2003). Guest Editors' Introduction: Agile Software Development: It's about Feedback and Change. *Computer*, 36(6), 39-43.

Witzel, A. (2000). *The problem-centered interview*. Paper presented at the Forum Qualitative

Sozialforschung/Forum: Qualitative Social Research.

Wright, K. B. (2005). Researching Internet-based populations: Advantages and disadvantages

of online survey research, online questionnaire authoring software packages, and web survey services. *Journal of Computer - Mediated Communication*, 10(3), 00-00.

Zelkowitz, M. V., & Wallace, D. R. (1998). Experimental models for validating technology.

Computer, 31(5), 23-31.

Zhang, Y. (2000). Using the Internet for survey research: A case study. *Journal of the American society for information science*, 51(1), 57-68.

Zikmund, W., Babin, B., Carr, J., & Griffin, M. (2012). *Business research methods*: Cengage

Learning.