

المشروع النهائي

على جودة مخرجات مشاريع (Agile) أثر تطبيق منهجية الإدارة الرشيدة  
البرمجيات في الشركات التقنية

دراسة تحليلية تكاملية للأدبيات والمعايير المهنية

إعداد الطالب

عبدالله محمد ناصر الوشيقي

الرقم الجامعي: 253000561

إشراف

د. الفیصل عبد الحمید

د. فايز جراد

العام الدراسي 2026

إقرار الباحث

أقر أنا الطالب /عبدالله محمد ناصر الوشيقري، بأن هذا المشروع بعنوان:

«أثر تطبيق منهجية الإدارة الرشيقة (Agile) على جودة مخرجات مشاريع البرمجيات في الشركات التقنية»

قد تم إعداده وتقديمه ضمن متطلبات الحصول على درجة الماجستير في إدارة الأعمال (MBA) في جامعة ميدأوشن.

كما أقر بأن هذا المشروع من إعدادي، وأن جميع المعلومات والأفكار والاقتباسات الواردة فيه قد تم توثيقها والإشارة إلى مصادرها في قائمة المراجع وفق الأصول العلمية المتبعة.

وأقر كذلك بأن هذا المشروع لم يسبق تقديمه كاملاً أو جزءاً منه للحصول على درجة علمية في جامعة أو جهة تعليمية أخرى.

اسم الطالب :عبدالله محمد ناصر الوشيقري

الرقم الجامعي: 253000561 :

التوقيع:.....

التاريخ: ..... / ..... / 2026م

## الشكر والتقدير

الحمد لله رب العالمين، الذي بنعمته تتم الصالحات، والصلاة والسلام على نبينا محمد وعلى آله وصحبه أجمعين.

أتقدم بخالص الشكر والتقدير إلى جامعة ميدأوشن، ممثلة في كلية الإدارة، وإلى أعضاء هيئة التدريس في برنامج ماجستير إدارة الأعمال، لما قدموه من علم وتوجيه ودعم طوال فترة الدراسة.

كما أتقدم بجزيل الشكر والامتنان إلى المشرفين على هذا المشروع:

د. الفیصل عبد الحمید

د. فايز جراد

وذلك لما قدماه من توجيهات علمية وملاحظات قيمة أسهمت في تطوير هذا المشروع وتحسين محتواه وإخراجه بالصورة الأكاديمية المناسبة.

كما أتوجه بالشكر إلى كل من دعمني وشجعني خلال مرحلة إعداد هذا المشروع، وأسأل الله أن يجزي الجميع خير الجزاء.

## الباحث

عبدالله محمد ناصر الوشيقي

هدفت هذه الدراسة إلى تحليل أثر تطبيق منهجية الإدارة الرشيقة (Agile) على جودة مخرجات مشاريع البرمجيات في الشركات التقنية، والتعرف على أبرز الممارسات الرشيقة التي تسهم في تحسين جودة المنتجات البرمجية، إلى جانب تحديد العوامل التنظيمية والفنية التي تدعم نجاح تطبيق هذه المنهجية أو تحد من فاعليتها.

اعتمدت الدراسة على المنهج الوصفي التحليلي من خلال مراجعة تكاملية للأدبيات العلمية والدراسات السابقة والمعايير المهنية ذات العلاقة، ومنها بيان الإدارة الرشيقة، ودليل Scrum، ودليل الممارسات الرشيقة الصادر عن معهد إدارة المشاريع، ومعايير جودة البرمجيات ISO/IEC 25010:2023. وقد تم تحليل العلاقة بين ممارسات الإدارة الرشيقة ومجموعة من أبعاد جودة البرمجيات، مثل الملاءمة الوظيفية، والموثوقية، وكفاءة الأداء، وسهولة الاستخدام، والأمان، والقابلية للصيانة.

أظهرت نتائج الدراسة أن تطبيق منهجية الإدارة الرشيقة يمكن أن يسهم في تحسين جودة مخرجات مشاريع البرمجيات من خلال التسليم التكراري، والحصول على التغذية الراجعة المبكرة، وتعزيز التعاون بين أعضاء الفريق وأصحاب المصلحة، واكتشاف المشكلات والعيوب في مراحل مبكرة. كما تبين أن تحقيق هذه النتائج يعتمد بدرجة كبيرة على جودة التطبيق، ومدى دعم الإدارة العليا، وتمكين فرق العمل، ووجود ممارسات هندسية فعالة مثل الاختبارات الآلية، والتكامل المستمر، ومراجعة الشفرة البرمجية، وتعريف واضح لمعايير الإنجاز.

وخلصت الدراسة إلى أن تطبيق Agile لا يضمن تحسين الجودة بصورة تلقائية، إذ قد يؤدي التطبيق الشكلي أو الجزئي إلى زيادة سرعة التسليم دون تحقيق تحسن فعلي في جودة المنتج. وبناءً على ذلك، أوصت الدراسة بربط تطبيق الإدارة الرشيقة بنظام واضح لحوكمة الجودة، واعتماد مؤشرات أداء تجمع بين سرعة الإنجاز وجودة المخرجات، وتطوير مهارات فرق العمل، وتطبيق المنهجية تدريجياً بما يتناسب مع طبيعة الشركة ومشاريعها.

الكلمات المفتاحية: الإدارة الرشيقة، Agile، جودة البرمجيات، مشاريع البرمجيات، الشركات التقنية.

## **Abstract**

**This study aimed to analyze the impact of applying Agile management methodology on the quality of software project deliverables in technology companies. It also sought to identify the most important Agile practices that contribute to improving software product quality, as well as the organizational and technical factors that support or limit the effectiveness of Agile implementation.**

**The study adopted a descriptive analytical approach through an integrative review of relevant academic literature, previous studies, and professional standards. These included the Agile Manifesto, the Scrum Guide, the Agile Practice Guide issued by the Project Management Institute, and the ISO/IEC 25010:2023 software quality model. The relationship between Agile practices and several software quality dimensions was examined, including functional suitability, reliability, performance efficiency, usability, security, and maintainability.**

**The findings indicated that Agile methodology can improve the quality of software project deliverables through iterative delivery, early feedback, stronger collaboration among team members and stakeholders, and earlier detection of defects and problems. The results also showed that these benefits depend largely on the quality of implementation, senior management support, team empowerment, and the use of effective engineering practices such as automated testing, continuous integration, code review, and a clear definition of done.**

**The study concluded that Agile implementation does not automatically guarantee quality improvement. A superficial or partial application may increase delivery speed without achieving a real improvement in the final product. Therefore, the study recommended linking Agile implementation to a clear quality governance system, adopting performance indicators that balance delivery speed and output quality, developing team capabilities, and applying Agile practices gradually according to the nature of the company and its projects.**

**Keywords: Agile management, software quality, software projects, technology companies, project management.**

| الموضوع                                                | الصفحة |
|--------------------------------------------------------|--------|
| إقرار الباحث                                           | 2      |
| الشكر والتقدير                                         | 3      |
| الملخص                                                 | 4      |
| Abstract                                               | 5      |
| الفهرس                                                 | 6      |
| الفصل الأول (الإطار العام للدراسة)                     | 7      |
| الفصل الثاني (الإطار النظري للدراسة)                   | 12     |
| الفصل الثالث (الدراسات السابقة والثغرة البحثية)        | 21     |
| الفصل الرابع (منهجية الدراسة)                          | 27     |
| الفصل الخامس (نتائج الدراسة ومناقشتها)                 | 36     |
| الفصل السادس (النموذج التطبيقي – المقترحات – التوصيات) | 47     |
| الخاتمة                                                | 60     |
| المراجع                                                | 61     |

## 1.1 مقدمة الدراسة

شهد قطاع التقنية خلال السنوات الأخيرة تطورًا متسارعًا نتيجة الاعتماد المتزايد على الأنظمة الرقمية والتطبيقات البرمجية في مختلف المجالات، مثل التعليم، والصحة، والخدمات المالية، والتجارة الإلكترونية، والاتصالات، والخدمات الحكومية. وقد أدى هذا التطور إلى زيادة الطلب على تطوير برمجيات ذات جودة عالية، وقادرة على تلبية احتياجات المستخدمين، والاستجابة السريعة للتغيرات المستمرة في بيئة العمل والتقنيات المستخدمة.

وتواجه مشاريع البرمجيات عددًا من التحديات، من أبرزها تغير متطلبات العملاء، وارتفاع مستوى المنافسة، وضغط الوقت، وزيادة توقعات المستخدمين، والحاجة إلى إطلاق المنتجات والخدمات الرقمية خلال فترات زمنية قصيرة. كما قد تواجه بعض المشاريع مشكلات تتعلق بتأخر التسليم، وارتفاع التكلفة، وكثرة الأخطاء البرمجية، وعدم توافق المنتج النهائي مع احتياجات العميل الفعلية. اعتمدت العديد من الشركات في السابق على المنهجيات التقليدية في إدارة مشاريع البرمجيات، والتي تقوم على تقسيم المشروع إلى مراحل متتابعة تبدأ بجمع المتطلبات، ثم التصميم، ثم التطوير، ثم الاختبار، وأخيرًا التسليم. وعلى الرغم من أن هذه المنهجيات قد تكون مناسبة في المشاريع التي تتميز بوضوح المتطلبات واستقرارها، فإنها قد تواجه صعوبة في التعامل مع المشاريع التي تتغير متطلباتها بصورة مستمرة. ونتيجة لهذه التحديات، ظهرت منهجية الإدارة الرشيقة Agile بوصفها أحد الأساليب الحديثة في إدارة وتطوير مشاريع البرمجيات. تعتمد هذه المنهجية على تنفيذ المشروع بصورة تدريجية وتكرارية، وتقسيم العمل إلى دورات زمنية قصيرة، مع التركيز على التعاون المستمر بين أعضاء الفريق والعمل، وتقديم أجزاء قابلة للاستخدام من المنتج خلال مراحل المشروع، بدلًا من الانتظار حتى نهاية المشروع لتسليم المنتج كاملاً. وتساعد الإدارة الرشيقة على الاستجابة السريعة للتغيرات، والحصول على ملاحظات العميل بصورة مبكرة، واكتشاف الأخطاء والمشكلات أثناء مراحل التطوير، مما قد يساهم في تحسين جودة مخرجات المشروع. ومع ذلك، فإن مجرد تطبيق منهجية Agile لا يعني بالضرورة تحقيق نتائج إيجابية، إذ تعتمد فعالية المنهجية على مستوى التطبيق الفعلي، ودعم الإدارة العليا، وخبرة الفريق، ووضوح الأدوار، واستخدام الممارسات الهندسية المناسبة.

ومن هذا المنطلق، تسعى هذه الدراسة إلى تحليل أثر تطبيق منهجية الإدارة الرشيقة على جودة مخرجات مشاريع البرمجيات في الشركات التقنية، والتعرف على أهم الممارسات والعوامل التي تساعد على تحقيق جودة أفضل للمنتجات البرمجية.

## 1.2 مشكلة الدراسة

تعاني بعض مشاريع البرمجيات من مشكلات تتعلق بانخفاض جودة المخرجات، وكثرة العيوب البرمجية، وتأخر اكتشاف الأخطاء، وعدم توافق المنتج النهائي مع احتياجات المستخدمين. وقد تظهر هذه المشكلات نتيجة ضعف التواصل بين الأطراف المشاركة في المشروع، أو عدم وضوح المتطلبات، أو تأخر مشاركة العميل في مراجعة المنتج، أو تنفيذ الاختبارات في المراحل النهائية فقط. وقد دفعت هذه التحديات عددًا كبيرًا من الشركات التقنية إلى تطبيق منهجية الإدارة الرشيقة، بهدف زيادة المرونة، وتسريع عمليات التطوير، وتحسين مستوى التعاون، ورفع جودة المنتجات البرمجية. إلا أن نتائج تطبيق Agile تختلف من شركة إلى أخرى، فقد تحقق بعض الشركات تحسنًا واضحًا في الجودة والأداء، بينما تكتفي شركات أخرى بتطبيق عدد من الاجتماعات والمصطلحات الرشيقة دون تغيير حقيقي في طريقة العمل.

وقد يركز بعض الفرق على سرعة إنجاز المهام وتسليم الإصدارات، دون الاهتمام الكافي بجودة الشفرة البرمجية، أو الاختبارات الآلية، أو مراجعة الأخطاء، أو إدارة الدين التقني. وفي هذه الحالة، قد تتحسن سرعة التسليم بصورة مؤقتة، ولكن قد تتراكم المشكلات الفنية وتؤثر لاحقًا في موثوقية النظام وقابليته للصيانة والتطوير.

وبناءً على ذلك، تتمثل مشكلة الدراسة في الحاجة إلى معرفة مدى تأثير تطبيق منهجية الإدارة الرشيدة على جودة مخرجات مشاريع البرمجيات، وتحديد الممارسات التي تحقق أثراً فعلياً في الجودة، والعوامل التي تساعد على نجاح التطبيق أو تعيقه.

ويمكن صياغة مشكلة الدراسة في السؤال الرئيس الآتي:

ما أثر تطبيق منهجية الإدارة الرشيدة (Agile) على جودة مخرجات مشاريع البرمجيات في الشركات التقنية؟

### 1.3 أسئلة الدراسة

يتفرع من السؤال الرئيس مجموعة من الأسئلة الفرعية:

1. ما المقصود بمنهجية الإدارة الرشيدة، وما أهم مبادئها وممارساتها؟
2. ما المقصود بجودة مخرجات مشاريع البرمجيات، وما أهم أبعادها؟
3. كيف يسهم تطبيق Agile في تحسين جودة المنتجات البرمجية؟
4. ما الممارسات الرشيدة الأكثر ارتباطاً بتحسين الجودة؟
5. ما العوامل التنظيمية والفنية التي تساعد على نجاح تطبيق المنهجية؟
6. ما أبرز التحديات التي قد تحد من تأثير Agile في جودة المخرجات؟
7. ما المؤشرات التي يمكن استخدامها لقياس جودة مخرجات مشاريع البرمجيات؟
8. ما النموذج المقترح الذي يمكن أن يساعد الشركات التقنية على ربط تطبيق Agile بالجودة؟

### 1.4 أهداف الدراسة

تهدف هذه الدراسة إلى تحقيق مجموعة من الأهداف، من أهمها:

1. توضيح مفهوم منهجية الإدارة الرشيدة ومبادئها الأساسية .
2. التعرف على مفهوم جودة مخرجات مشاريع البرمجيات وأبعادها .
3. تحليل أثر تطبيق Agile على جودة المنتجات البرمجية .
4. تحديد الممارسات الرشيدة التي تسهم بصورة مباشرة في تحسين الجودة .
5. التعرف على العوامل التنظيمية والفنية التي تدعم نجاح تطبيق المنهجية .
6. توضيح المخاطر الناتجة عن التطبيق الجزئي أو الشكلي لمنهجية Agile .
7. تحديد مجموعة من مؤشرات الأداء المناسبة لقياس جودة مخرجات البرمجيات .
8. تقديم نموذج تطبيقي يساعد الشركات التقنية على تحقيق التوازن بين سرعة التسليم وجودة المنتج .
9. تقديم توصيات عملية يمكن الاستفادة منها في تطوير إدارة مشاريع البرمجيات.

### 1.5 أهمية الدراسة

تنبع أهمية هذه الدراسة من الأهمية المتزايدة لمشاريع البرمجيات في دعم عمليات الشركات والمؤسسات، ومن حاجة الشركات التقنية إلى تطوير منتجات رقمية ذات جودة مرتفعة خلال فترات زمنية قصيرة.

### 1.5.1 الأهمية العلمية

تتمثل الأهمية العلمية للدراسة في مساهمتها في الربط بين مجال إدارة المشاريع الرشيدة ومجال جودة البرمجيات. كما تقدم الدراسة تحليلاً للممارسات الرشيدة من حيث تأثيرها في خصائص الجودة، بدلاً من الاكتفاء بالنظر إلى Agile بوصفها وسيلة لتسريع إنجاز المشروع فقط.

كما تسهم الدراسة في توضيح أن نجاح المنهجية لا يعتمد على تطبيق الاجتماعات أو تقسيم العمل إلى دورات قصيرة فحسب، بل يعتمد على مجموعة متكاملة من العناصر التنظيمية والفنية، مثل دعم القيادة، وتمكين الفرق، وجودة التواصل، والاختبارات المستمرة، والتكامل بين مراحل التطوير.

### 1.5.2 الأهمية العملية

تساعد نتائج الدراسة مديري المشاريع ومديري المنتجات وفرق التطوير والجودة في الشركات التقنية على فهم الطريقة المناسبة لتطبيق Agile كما تساعد على تحديد الممارسات التي يجب التركيز عليها لتحقيق تحسين فعلي في جودة المنتجات. وقد تستفيد الشركات من الدراسة في بناء مؤشرات أداء توازن بين سرعة التسليم وجودة المخرجات، وتحديد الفجوات الموجودة في تطبيق المنهجية، ووضع خطة تدريجية للتحسين.

### 1.5.3 أهمية الدراسة للشركات التقنية

تساعد الدراسة الشركات التقنية على تقليل الأخطاء البرمجية، وتحسين رضا العملاء، ورفع موثوقية المنتجات، وتقليل عمليات إعادة العمل، وتحسين القدرة على الاستجابة للتغيرات. كما يمكن أن تسهم في تقليل تكاليف معالجة الأخطاء التي يتم اكتشافها بعد إطلاق المنتج، لأن اكتشاف المشكلات خلال مراحل التطوير المبكرة يكون عادة أقل تكلفة وأكثر سهولة.

### 1.6 فرضيات الدراسة

استناداً إلى موضوع الدراسة والأدبيات المرتبطة به، تعتمد الدراسة على الفرضيات التحليلية الآتية:

#### الفرضية الأولى

يوجد أثر إيجابي لتطبيق منهجية الإدارة الرشيقة على جودة مخرجات مشاريع البرمجيات.

#### الفرضية الثانية

يسهم التسليم التكراري والتغذية الراجعة المبكرة من العميل في تحسين الملاءمة الوظيفية للمنتج البرمجي.

#### الفرضية الثالثة

تسهم الممارسات الهندسية، مثل الاختبارات الآلية والتكامل المستمر ومراجعة الشفرة البرمجية، في تحسين موثوقية البرمجيات وقابليتها للصيانة.

#### الفرضية الرابعة

يسهم دعم الإدارة العليا وتمكين فرق العمل في تعزيز أثر Agile على جودة المخرجات.

#### الفرضية الخامسة

يؤدي التطبيق الشكلي أو الجزئي لمنهجية Agile إلى ضعف أثرها في جودة المنتجات البرمجية.

### 1.7 حدود الدراسة

تتضمن الدراسة مجموعة من الحدود التي تحدد نطاقها، وذلك على النحو الآتي:

#### 1.7.1 الحدود الموضوعية

تركز الدراسة على أثر تطبيق منهجية الإدارة الرشيدة على جودة مخرجات مشاريع البرمجيات. ولا تتناول بصورة تفصيلية جميع الآثار المالية أو التسويقية أو الاستراتيجية الناتجة عن تطبيق المنهجية.

#### 1.7.2 الحدود المنهجية

تعتمد الدراسة على المنهج الوصفي التحليلي والمراجعة التكاملية للأدبيات والدراسات السابقة والمعايير المهنية. ولا تعتمد على جمع بيانات ميدانية جديدة من خلال استبانة أو مقابلات.

#### 1.7.3 الحدود المكانية

تتناول الدراسة الشركات التقنية بصورة عامة، مع محاولة تقديم نتائج وتوصيات يمكن الاستفادة منها في الشركات التقنية العربية.

#### 1.7.4 الحدود الزمنية

تعتمد الدراسة على مجموعة من المراجع والدراسات العلمية والمهنية، مع التركيز قدر الإمكان على المصادر الحديثة، والاستفادة من المراجع الأساسية التي ساهمت في تأسيس مفهوم الإدارة الرشيدة.

### 1.8 مصطلحات الدراسة

#### 1.8.1 الإدارة الرشيدة Agile

هي منهجية لإدارة وتطوير المشاريع تعتمد على تقسيم العمل إلى دورات قصيرة، والتعاون المستمر بين الفريق والعميل، والاستجابة السريعة للتغيرات، والتسليم التدريجي للمنتج.

#### 1.8.2 جودة البرمجيات

هي درجة قدرة المنتج البرمجي على تلبية احتياجات المستخدمين والمتطلبات المحددة، وتحقيق مستوى مناسب من الأداء والموثوقية والأمان وسهولة الاستخدام والقابلية للصيانة.

#### 1.8.3 مشاريع البرمجيات

هي المشاريع التي تهدف إلى تطوير أو تحديث أو تشغيل نظام أو تطبيق أو منصة برمجية لخدمة مجموعة محددة من المستخدمين أو العمليات.

#### 1.8.4 مخرجات المشروع

هي المنتجات أو الوظائف أو الإصدارات البرمجية الناتجة عن تنفيذ المشروع، والتي يمكن استخدامها أو اختبارها أو تسليمها للعميل.

#### 1.8.5 التسليم التكراري

هو أسلوب يتم من خلاله تطوير المنتج وتسليمه على مراحل قصيرة ومتكررة، بحيث يتم تقديم جزء قابل للاستخدام أو الاختبار في نهاية كل دورة.

#### 1.8.6 التغذية الراجعة

هي الملاحظات والمعلومات التي يقدمها العميل أو المستخدم أو أصحاب المصلحة بعد مراجعة المخرجات، بهدف تعديل المنتج أو تحسينه.

#### 1.8.7 تعريف الإنجاز

هو مجموعة من المعايير المتفق عليها التي يجب استيفائها قبل اعتبار المهمة أو الوظيفة البرمجية مكتملة، مثل اجتياز الاختبارات والمراجعة والتوثيق.

## 1.9 هيكل الدراسة

تتكون هذه الدراسة من ستة فصول رئيسية.

يتناول الفصل الأول الإطار العام للدراسة، ويشمل المقدمة، ومشكلة الدراسة، والأسئلة، والأهداف، والأهمية، والفرضيات، والحدود، والمصطلحات.

أما الفصل الثاني فيتناول الإطار النظري للإدارة الرشيدة وجودة البرمجيات، وأهم المبادئ والممارسات والأطر المستخدمة.

ويتناول الفصل الثالث الدراسات السابقة، وتحليل نتائجها، وتحديد الثغرة البحثية، وبناء النموذج المفاهيمي للدراسة.

ويتناول الفصل الرابع منهجية الدراسة، ونوع المنهج، ومصادر البيانات، ومعايير اختيار المراجع، وطريقة التحليل، والاعتبارات الأخلاقية.

أما الفصل الخامس فيعرض نتائج التحليل ومناقشتها، ويبين أثر الممارسات الرشيدة على أبعاد جودة البرمجيات.

ويقدم الفصل السادس النموذج التطبيقي المقترح، وخارطة التنفيذ، ومؤشرات الأداء، والمخاطر، والتوصيات، ثم تختتم الدراسة بالخاتمة والمراجع والملاحق.

## الفصل الثاني: الإطار النظري للدراسة

### 2.1 تمهيد

يتناول هذا الفصل الأسس النظرية المرتبطة بموضوع الدراسة، من خلال توضيح مفهوم الإدارة الرشيقة Agile، ونشأتها، وأهم قيمها ومبادئها، إلى جانب عرض أشهر الأطر والممارسات المستخدمة في إدارة مشاريع البرمجيات. كما يتناول الفصل مفهوم جودة البرمجيات وأبرز أبعادها، ويوضح العلاقة بين تطبيق الممارسات الرشيقة وتحسين جودة مخرجات المشاريع التقنية. ويساعد هذا الإطار النظري في بناء فهم متكامل للمتغيرات الرئيسية في الدراسة، وتفسير النتائج التي ستم مناقشتها في الفصول اللاحقة.

### 2.2 مفهوم الإدارة الرشيقة Agile

الإدارة الرشيقة هي منهجية حديثة لإدارة المشاريع وتطوير المنتجات، تقوم على المرونة، والعمل التكراري، والتعاون المستمر، والاستجابة السريعة للتغيرات. وقد ارتبط ظهورها بصورة أساسية بمجال تطوير البرمجيات، نتيجة الحاجة إلى أسلوب أكثر قدرة على التعامل مع تغير المتطلبات وعدم وضوحها بصورة كاملة منذ بداية المشروع. وتعتمد Agile على تقسيم المشروع إلى أجزاء صغيرة يمكن تطويرها واختبارها وتسليمها على مراحل متكررة. ويتم خلال كل مرحلة مراجعة النتائج والحصول على ملاحظات العميل أو المستخدم، ثم تعديل خطة العمل والأولويات بناءً على المعلومات الجديدة. ويختلف هذا الأسلوب عن المنهجيات التقليدية التي تعتمد غالباً على إعداد خطة تفصيلية كاملة في بداية المشروع، ثم تنفيذ المراحل بصورة متتابعة. ففي البيئة الرشيقة لا يُنظر إلى الخطة على أنها ثابتة لا يمكن تعديلها، بل تُعد أداة قابلة للتحديث وفق التغيرات والنتائج الفعلية. ولا تعني الإدارة الرشيقة غياب التخطيط أو التوثيق أو الرقابة، وإنما تعني استخدام هذه العناصر بالمستوى الذي يخدم أهداف المشروع ويحقق قيمة فعلية للعميل، دون التسبب في إجراءات معقدة أو تأخير غير ضروري.

### 2.3 نشأة الإدارة الرشيقة

ظهرت الحاجة إلى الإدارة الرشيقة نتيجة المشكلات التي واجهتها مشاريع البرمجيات عند استخدام الأساليب التقليدية. فقد كانت بعض المشاريع تستغرق فترات طويلة قبل عرض المنتج على العميل، وفي كثير من الحالات كان يتم اكتشاف أن المتطلبات تغيرت أو أن المنتج النهائي لا يتوافق بصورة كاملة مع توقعات المستخدم.

وفي عام 2001، اجتمع عدد من الخبراء والممارسين في مجال تطوير البرمجيات، وصاغوا ما يعرف باسم بيان الإدارة الرشيقة لتطوير البرمجيات Agile Manifesto. وقد تضمن البيان أربع قيم أساسية واثنى عشر مبدأً توجه طريقة العمل في المشاريع الرشيقة. ركز البيان على تقديم منتجات تعمل بصورة فعلية، والتعاون مع العميل، وتمكين فرق العمل، والاستجابة للتغيير، بدلاً من الاعتماد المفرط على الإجراءات الجامدة والتوثيق الطويل والخطط غير القابلة للتعديل. ومنذ ذلك الوقت، توسع استخدام Agile خارج مجال البرمجيات ليشمل مجالات أخرى، مثل تطوير المنتجات، والتسويق، والعمليات، والابتكار، وإدارة الأعمال. إلا أن مجال البرمجيات ما يزال من أكثر المجالات ارتباطاً بها.

### 2.4 قيم الإدارة الرشيقة

يقوم بيان Agile على أربع قيم رئيسية، وهي:

#### 2.4.1 الأفراد والتفاعلات أهم من العمليات والأدوات

تعطي الإدارة الرشيدة أهمية كبيرة للتواصل والتعاون بين أعضاء الفريق. فوجود أدوات وأنظمة متقدمة لا يضمن نجاح المشروع إذا كان التواصل ضعيفاً أو كانت الأدوار غير واضحة.

ويعني ذلك أن نجاح المشروع يعتمد بصورة أساسية على كفاءة الأفراد، وقدرتهم على العمل المشترك، وسرعة تبادل المعلومات، وحل المشكلات بصورة جماعية.

#### 2.4.2 البرمجيات العاملة أهم من التوثيق الشامل

لا ترفض Agile التوثيق، لكنها ترفض أن يصبح التوثيق هدفاً في حد ذاته. فالقيمة الحقيقية للمشروع تظهر في وجود منتج برمجي يعمل ويحقق احتياجات المستخدم.

ويجب أن يكون التوثيق مناسباً لطبيعة المشروع ومخاطره، بحيث يدعم العمل والاستمرارية دون أن يتحول إلى عبء يستهلك الوقت والموارد.

#### 2.4.3 التعاون مع العميل أهم من التفاوض حول العقود

تركز Agile على إشراك العميل بصورة مستمرة طوال دورة تطوير المنتج، بدلاً من الاكتفاء بتحديد المتطلبات في بداية المشروع ثم الانتظار حتى التسليم النهائي.

ويساعد التعاون المستمر على توضيح الأولويات، وفهم التغييرات، واكتشاف عدم التوافق مبكراً، وتحسين رضا العميل عن المنتج النهائي.

#### 2.4.4 الاستجابة للتغيير أهم من اتباع الخطة

تدرك الإدارة الرشيدة أن متطلبات مشاريع البرمجيات قد تتغير بسبب تغير احتياجات المستخدم أو ظروف السوق أو التقنية. ولذلك تسمح بإعادة ترتيب الأولويات وتعديل الخطة عندما تظهر معلومات جديدة.

ولا يعني ذلك العمل دون خطة، بل يعني أن الخطة يجب أن تكون مرنة وقابلة للتحديث بما يحقق أفضل قيمة ممكنة للمشروع.

#### 2.5 مبادئ الإدارة الرشيدة

يتضمن بيان Agile اثني عشر مبدأً، ويمكن تلخيص أبرزها فيما يأتي:

1. تحقيق رضا العميل من خلال التسليم المبكر والمستمر للبرمجيات ذات القيمة .
  2. الترحيب بتغير المتطلبات، حتى في المراحل المتقدمة من المشروع، عندما يؤدي التغيير إلى تحقيق قيمة أفضل .
  3. تسليم منتجات أو وظائف برمجية بصورة متكررة خلال فترات زمنية قصيرة .
  4. تعزيز التعاون اليومي بين ممثلي الأعمال وفرق التطوير .
  5. بناء المشاريع حول أفراد مؤهلين، وتوفير الدعم والثقة اللازمة لهم .
  6. الاعتماد على التواصل المباشر والواضح بين أعضاء الفريق .
  7. اعتبار البرمجيات العاملة من أهم مؤشرات تقدم المشروع .
  8. المحافظة على وتيرة عمل مستقرة وقابلة للاستمرار .
  9. الاهتمام المستمر بالجودة الفنية والتصميم الجيد .
  10. التركيز على البساطة وتجنب الأعمال والإجراءات غير الضرورية .
  11. تمكين الفرق ذاتية التنظيم من اتخاذ القرارات المناسبة للعمل .
  12. مراجعة طريقة العمل بصورة دورية وتحديد فرص التحسين .
- وتوضح هذه المبادئ أن Agile لا تركز فقط على سرعة التسليم، بل تربط السرعة بالجودة، والتعاون، والاستدامة، والتحسين المستمر.

## 2.6 خصائص الإدارة الرشيدة

تتميز الإدارة الرشيدة بمجموعة من الخصائص التي تجعلها مناسبة لمشاريع البرمجيات، ومنها:

### 2.6.1 التطوير التكراري

يتم تطوير المنتج من خلال دورات قصيرة ومتكررة، ويجري في نهاية كل دورة إنتاج جزء قابل للاختبار أو الاستخدام. يساعد هذا الأسلوب على اكتشاف المشكلات بصورة مبكرة، ويقلل مخاطر انتظار نهاية المشروع قبل تقييم المنتج.

### 2.6.2 التسليم التدريجي

لا يتم تسليم المنتج كاملاً دفعة واحدة، بل يتم تطويره وتسليمه على مراحل. ويتيح ذلك تقديم الوظائف الأكثر أهمية أولاً، والحصول على قيمة مبكرة.

### 2.6.3 المرونة في إدارة المتطلبات

يمكن إعادة ترتيب المتطلبات والأولويات بناءً على ملاحظات العميل أو التغيرات في السوق أو نتائج الاختبار.

### 2.6.4 التعاون المستمر

تعتمد Agile على التواصل المستمر بين المطورين، ومدير المنتج، وفريق الجودة، والعميل، وأصحاب المصلحة.

### 2.6.5 الشفافية

تساعد الاجتماعات الدورية ولوحات العمل وقوائم المهام على إظهار حالة المشروع، والعوائق، والأعمال المنجزة والمتبقية.

### 2.6.6 التحسين المستمر

تتم مراجعة طريقة العمل بصورة دورية، وتحديد ما تم بنجاح، وما يحتاج إلى تحسين، ثم تطبيق إجراءات تطويرية في الدورات التالية.

## 2.7 أشهر أطر الإدارة الرشيدة

تتضمن الإدارة الرشيدة مجموعة من الأطر والممارسات، ولا تمثل Agile إطاراً واحداً محدداً. ومن أشهر هذه الأطر ما يأتي:

### 2.7.1 إطار Scrum

يعد Scrum من أكثر الأطر الرشيدة استخداماً في مشاريع البرمجيات. ويقوم على تقسيم العمل إلى دورات زمنية قصيرة تسمى Sprint، تستمر عادة من أسبوع إلى أربعة أسابيع.

ويتكون فريق Scrum غالباً من:

أ. مالك المنتج Product Owner

يتولى تحديد أولويات المنتج، وإدارة قائمة المتطلبات، والتواصل مع أصحاب المصلحة، والتأكد من أن الفريق يعمل على المهام التي تحقق أعلى قيمة.

ب. Scrum Master

يتولى دعم الفريق في تطبيق Scrum، وإزالة العوائق، وتيسير الاجتماعات، وتعزيز التحسين المستمر. ولا يعد مديرًا تقليدياً للفريق، بل يعمل بوصفه ميسراً وقائداً داعماً.

ج. فريق التطوير

يتكون من أفراد يمتلكون المهارات اللازمة لتصميم المنتج وتطويره واختباره وتسليمه. ويعمل الفريق بصورة مشتركة ويتحمل مسؤولية تحقيق هدف الدورة.

ومن أبرز أحداث Scrum:

### 1. تخطيط الدورة Sprint Planning

2. الاجتماع اليومي Daily Scrum.

3. مراجعة الدورة Sprint Review.

4. الاجتماع الاستعادي Sprint Retrospective.

ويؤثر Scrum في الجودة من خلال المراجعة المتكررة، والتغذية الراجعة، وتعريف الإنجاز، وإنتاج زيادات قابلة للاستخدام خلال كل دورة.

#### 2.7.2 منهجية Kanban

تركز Kanban على تحسين تدفق العمل وتصور المهام من خلال لوحة توضح مراحل العمل، مثل:

- الأعمال المطلوب تنفيذها .
- الأعمال قيد التنفيذ .
- الأعمال قيد المراجعة أو الاختبار .
- الأعمال المكتملة .

ومن أهم مبادئ Kanban تحديد حجم العمل قيد التنفيذ، بهدف تقليل التشتت واختناقات وتحسين سرعة إنجاز المهام. وتساعد Kanban فرق البرمجيات على اكتشاف المراحل التي تتراكم فيها الأعمال، وتحسين تدفق المهام، وتقليل زمن الدورة، ورفع القدرة على الاستجابة للطلبات الجديدة.

#### 2.7.3 البرمجة القصوى Extreme Programming

تعد البرمجة القصوى، المعروفة باسم XP، من المنهجيات الرشيقة التي تركز بصورة كبيرة على الجودة الفنية للبرمجيات. ومن أبرز ممارساتها:

1. التطوير المدفوع بالاختبار .
2. البرمجة الثنائية .
3. التكامل المستمر .
4. إعادة هيكلة الشفرة البرمجية .
5. الإصدارات الصغيرة والمتكررة .
6. مشاركة العميل بصورة مستمرة .

وتسهم هذه الممارسات في تقليل الأخطاء، وتحسين تصميم النظام، وزيادة قابلية الشفرة للصيانة والتطوير.

#### 2.7.4 DevOps

DevOps هو نهج يهدف إلى تعزيز التعاون بين فرق تطوير البرمجيات وفرق التشغيل والبنية التحتية. ويركز على الأتمتة، والتكامل المستمر، والتسليم المستمر، والمراقبة بعد النشر.

ويساعد DevOps على تقليل الوقت الفاصل بين تطوير الوظائف ونشرها، وتحسين استقرار عمليات الإطلاق، واكتشاف المشكلات التشغيلية بصورة أسرع.

ولا يعد DevOps بديلاً عن Agile، بل يمكن اعتباره امتداداً لها، لأنه يساعد على نقل مبادئ التعاون والتحسين المستمر إلى مراحل التشغيل والنشر.

## 2.8 مفهوم جودة البرمجيات

تشير جودة البرمجيات إلى قدرة المنتج البرمجي على تلبية المتطلبات المحددة واحتياجات المستخدمين، والعمل بصورة موثوقة وأمنة وفعالة وسهلة الاستخدام.

ولا تعني جودة البرمجيات خلو المنتج من الأخطاء فقط، بل تشمل مجموعة واسعة من الخصائص التي تحدد مدى نجاح المنتج في أداء وظائفه وتحقيق القيمة المطلوبة.

وقد تختلف معايير الجودة باختلاف طبيعة المنتج. فالأنظمة البنكية تحتاج إلى مستويات مرتفعة من الأمان والموثوقية، بينما قد تركز تطبيقات التجارة الإلكترونية على سهولة الاستخدام وسرعة الأداء، وقد تحتاج الأنظمة الصحية إلى الأمان والدقة والسلامة بصورة أكبر.

## 2.9 أبعاد جودة البرمجيات

وفقاً لمعايير جودة البرمجيات، يمكن تناول أبرز أبعاد الجودة كما يأتي:

### 2.9.1 الملاءمة الوظيفية

تعني قدرة المنتج على تقديم الوظائف التي يحتاجها المستخدم بصورة كاملة وصحيحة ومناسبة. وتشمل:

- اكتمال الوظائف .
- صحة النتائج .
- مناسبة الوظائف للاستخدام المطلوب .

### 2.9.2 كفاءة الأداء

تشير إلى مستوى أداء النظام مقارنة بالموارد المستخدمة.

وتشمل:

- سرعة الاستجابة .
- استهلاك الموارد .
- القدرة على التعامل مع عدد كبير من المستخدمين أو العمليات .

### 2.9.3 الموثوقية

تعني قدرة النظام على العمل بصورة مستقرة ومتواصلة دون أعطال متكررة.

وتشمل:

- التوافر .
- تحمل الأعطال .
- القدرة على الاستعادة .
- استقرار النظام .

### 2.9.4 سهولة الاستخدام

تعني مدى سهولة تعلم النظام واستخدامه من قبل المستخدمين.

وتشمل:

- وضوح الواجهات .
- سهولة التعلم .

- سهولة تنفيذ المهام .
- تقليل أخطاء المستخدم .
- رضا المستخدم عن التجربة .

#### 2.9.5 الأمان

يشير إلى قدرة النظام على حماية البيانات والوظائف من الاستخدام غير المصرح به أو التغيير أو التسريب. وتشمل الجودة الأمنية:

- سرية المعلومات .
- سلامة البيانات .
- التحقق من الهوية .
- إدارة الصلاحيات .
- إمكانية تتبع العمليات .

#### 2.9.6 القابلية للصيانة

تعني سهولة تعديل النظام وإصلاحه وتطويره مستقبلاً.

وتتضمن:

- سهولة تحليل الأخطاء .
- سهولة تعديل الشفرة .
- سهولة الاختبار .
- قابلية إعادة الاستخدام .
- وضوح تصميم النظام .

#### 2.9.7 التوافق

يعني قدرة النظام على العمل والتكامل مع الأنظمة الأخرى، وتبادل البيانات والمعلومات بصورة صحيحة.

#### 2.9.8 المرونة

تشير إلى قدرة المنتج على التكيف مع بيانات أو متطلبات أو تقنيات جديدة دون الحاجة إلى إعادة بناء كاملة.

### 2.10 العلاقة بين Agile وجودة البرمجيات

يمكن أن تؤثر الإدارة الرشيدة إيجابياً في جودة البرمجيات من خلال مجموعة من الآليات والممارسات، إلا أن هذا التأثير يعتمد على مستوى التطبيق وجودته.

#### 2.10.1 اكتشاف الأخطاء بصورة مبكرة

يساعد تقسيم العمل إلى دورات قصيرة على اختبار الوظائف أولاً بأول. وبدلاً من اكتشاف العيوب في نهاية المشروع، يمكن اكتشافها أثناء التطوير ومعالجتها بسرعة.

ويؤدي ذلك إلى تقليل تكلفة الإصلاح، لأن معالجة الخطأ في المراحل المبكرة تكون غالباً أسهل وأقل تكلفة من معالجته بعد إطلاق النظام.

#### 2.10.2 تحسين فهم متطلبات العميل

تتيح المراجعات المتكررة للعميل الاطلاع على المنتج خلال مراحل التطوير، وتقديم ملاحظاته بصورة مستمرة.

ويساعد ذلك على اكتشاف سوء الفهم أو المتطلبات غير الواضحة، ويزيد احتمالية أن يتوافق المنتج النهائي مع احتياجات المستخدم.

### 2.10.3 تحسين التواصل بين أعضاء الفريق

يسهم التواصل المستمر والاجتماعات القصيرة في تقليل تأخر المعلومات، والكشف عن العوائق، وتحسين التنسيق بين المطورين والمختبرين ومالك المنتج.

### 2.10.4 دمج الجودة داخل دورة التطوير

تتعامل الممارسات الرشيدة الفعالة مع الجودة بوصفها جزءاً من كل دورة، وليس مرحلة منفصلة تأتي بعد انتهاء التطوير. ويتم ذلك من خلال الاختبارات المستمرة، ومراجعة الشفرة، وتعريف الإنجاز، والتحقق من الوظائف قبل اعتبارها مكتملة.

### 2.10.5 تقليل مخاطر المشروع

يساعد التسليم التدريجي على تقليل حجم المخاطر، لأن الفريق لا ينتظر حتى نهاية المشروع لاكتشاف المشكلات. كما يمكن تعديل الأولويات أو إيقاف الوظائف غير المفيدة قبل استهلاك موارد كبيرة.

### 2.10.6 دعم التحسين المستمر

تتيح الاجتماعات الاستيعادية للفريق مراجعة طريقة العمل وتحديد أسباب الأخطاء والتأخير، ثم الاتفاق على إجراءات تحسين في الدورة التالية.

## 2.11 الممارسات الرشيدة المؤثرة في الجودة

توجد مجموعة من الممارسات التي ترتبط بصورة مباشرة بتحسين جودة البرمجيات، ومنها:

### 2.11.1 الاختبارات الآلية

تسمح الاختبارات الآلية بتنفيذ اختبارات متكررة بسرعة، والتأكد من أن التعديلات الجديدة لم تتسبب في تعطيل وظائف سابقة.

### 2.11.2 التكامل المستمر

يعني دمج التعديلات البرمجية بصورة متكررة، وتشغيل عمليات البناء والاختبار تلقائياً، مما يساعد على اكتشاف أخطاء الدمج مبكراً.

### 2.11.3 مراجعة الشفرة البرمجية

تسهم مراجعة الشفرة من قبل أعضاء آخرين في الفريق في اكتشاف الأخطاء، وتحسين أسلوب البرمجة، ونقل المعرفة بين الأفراد.

### 2.11.4 تعريف الإنجاز

يساعد تعريف الإنجاز على توحيد المعايير التي يجب تحققها قبل اعتبار الوظيفة مكتملة. ويمكن أن يشمل اجتياز الاختبارات، ومراجعة الشفرة، والتحقق الأمني، والتوثيق المطلوب.

### 2.11.5 مراجعة المنتج مع العميل

تسمح مراجعة المنتج في نهاية كل دورة بالحصول على تغذية راجعة حقيقية، والتأكد من أن الفريق يطور الوظائف المناسبة.

### 2.11.6 إعادة هيكلة الشفرة

تهدف إعادة الهيكلة إلى تحسين تصميم الشفرة وتنظيمها دون تغيير الوظائف الظاهرة للمستخدم. وتساعد على تقليل التعقيد والدين التقني وزيادة قابلية الصيانة.

## 2.12 عوامل نجاح تطبيق Agile

يعتمد نجاح تطبيق الإدارة الرشيدة على مجموعة من العوامل، من أبرزها:

### 2.12.1 دعم الإدارة العليا

يساعد دعم الإدارة في توفير الموارد، وإزالة العوائق، وتمكين الفريق، وتسهيل التغيير التنظيمي.

## 2.12.2 تمكين فريق العمل

يجب أن يمتلك الفريق صلاحية مناسبة لتنظيم العمل واتخاذ القرارات الفنية اليومية، بدلاً من الاعتماد على موافقات متكررة تعطل الإنجاز.

## 2.12.3 وضوح الأدوار والمسؤوليات

يساعد وضوح دور مالك المنتج وقائد الفريق والمطورين وفريق الجودة على تقليل التعارض وتحسين سرعة اتخاذ القرار.

## 2.12.4 مشاركة العميل

يجب أن يكون العميل أو ممثله متاحاً لتوضيح المتطلبات ومراجعة المخرجات وترتيب الأولويات.

## 2.12.5 الكفاءة الفنية

لا يمكن تحقيق جودة مرتفعة دون توفر مهارات مناسبة في البرمجة، والاختبار، والتصميم، والأمن، والأتمتة.

## 2.12.6 الثقافة التنظيمية

تحتاج Agile إلى ثقافة تشجع التعاون والشفافية والتعلم من الأخطاء والتحسين المستمر.

## 2.12.7 استخدام مؤشرات أداء متوازنة

ينبغي ألا تقتصر المؤشرات على عدد المهام المنجزة أو سرعة التسليم، بل يجب أن تشمل العيوب، ورضا المستخدم، والموثوقية، والدين التقني، وزمن معالجة المشكلات.

## 2.13 تحديات تطبيق الإدارة الرشيقة

على الرغم من فوائد Agile ، فقد تواجه الشركات عدداً من التحديات أثناء التطبيق، ومنها:

### 2.13.1 مقاومة التغيير

قد يرفض بعض الموظفين أو المديرين تغيير أساليب العمل المعتادة، خاصة إذا لم تتضح لهم أهداف التحول وفوائده.

### 2.13.2 التطبيق الشكلي

قد تكتفي بعض الشركات بعقد الاجتماعات اليومية واستخدام المصطلحات الرشيقة، دون تمكين الفرق أو تغيير طريقة اتخاذ القرار أو دمج الجودة في العمل.

### 2.13.3 ضعف مشاركة العميل

قد يؤدي عدم تفرغ العميل أو ممثله إلى تأخر القرارات وعدم وضوح الأولويات.

### 2.13.4 نقص الخبرات

قد يؤدي ضعف الخبرة في الاختبارات الآلية والتكامل المستمر والتصميم الرشيق إلى عدم تحقيق الأثر المطلوب.

### 2.13.5 التركيز المفرط على السرعة

قد يؤدي ضغط التسليم المستمر إلى إهمال الجودة الفنية، وتأجيل معالجة المشكلات، وتراكم الدين التقني.

### 2.13.6 صعوبة التطبيق في المؤسسات الكبيرة

قد تواجه الشركات الكبيرة تحديات تتعلق بتعدد الإدارات، وطول إجراءات الموافقة، والاعتماد المتبادل بين الفرق، وصعوبة توحيد أساليب العمل.

## 2.14 التطبيق الهجين لمنهجية Agile

لا تستخدم جميع الشركات منهجية رشيقة خالصة، بل تعتمد بعض الشركات نهجاً هجيناً يجمع بين الأساليب التقليدية والرشيقة.

وقد يكون النهج الهجين مناسباً للمشاريع التي تتطلب مستوى مرتفعاً من التوثيق أو الاعتمادات الرسمية أو الالتزام التنظيمي، مع الحاجة في الوقت نفسه إلى المرونة والتسليم التدريجي.

ويمكن، على سبيل المثال، إعداد خطة عامة وميزانية ومراحل اعتماد وفق الأسلوب التقليدي، مع تنفيذ أعمال التطوير الفعلية من خلال دورات قصيرة ومراجعات مستمرة.

ويتطلب نجاح النهج الهجين وضوح طريقة دمج الأساليب، وتجنب تكرار الإجراءات أو فرض رقابة تقليدية تتعارض مع تمكين الفرق.

## 2.15 خلاصة الفصل

تناول هذا الفصل مفهوم الإدارة الرشيدة، ونشأتها، وقيمها، ومبادئها، وخصائصها، وأشهر الأطر المرتبطة بها، مثل Scrum و Kanban و DevOps و Extreme Programming.

كما تناول مفهوم جودة البرمجيات وأبرز أبعادها، مثل الملاءمة الوظيفية، والموثوقية، وكفاءة الأداء، وسهولة الاستخدام، والأمان، والقابلية للصيانة.

وأظهر الإطار النظري أن Agile يمكن أن تسهم في تحسين جودة البرمجيات من خلال التسليم التكراري، والتغذية الراجعة المبكرة، والاختبارات المستمرة، وتحسين التواصل، ودعم التحسين المستمر. إلا أن تحقيق هذه الفوائد يعتمد على جودة التطبيق، ودعم الإدارة، وتمكين الفرق، ووجود ممارسات هندسية فعالة.

وسيتناول الفصل التالي الدراسات السابقة المرتبطة بموضوع الدراسة، وتحليل نتائجها، وتحديد الثغرة البحثية التي يسعى المشروع إلى معالجتها.

### 3.1 تمهيد

تمثل الدراسات السابقة أساساً مهماً في بناء البحث العلمي، إذ تساعد على التعرف على النتائج التي توصل إليها الباحثون في الموضوع نفسه أو في موضوعات قريبة منه، كما تساهم في توضيح أوجه الاتفاق والاختلاف بين الدراسات، وتحديد الجوانب التي ما تزال بحاجة إلى مزيد من التحليل.

ويتناول هذا الفصل مجموعة من الدراسات والمرجعيات العلمية المرتبطة بتطبيق منهجية الإدارة الرشيقة Agile في مشاريع البرمجيات، مع التركيز على أثرها في جودة المخرجات، ونجاح المشاريع، ورضا العملاء، وكفاءة فرق العمل، والعوامل التي تساعد على نجاح التحول الرشيق أو تعيقه.

كما يتضمن الفصل تحليلاً نقدياً للدراسات السابقة، وبيان الثغرة البحثية التي تسعى الدراسة الحالية إلى معالجتها، ثم عرض النموذج المفاهيمي الذي يوضح العلاقة بين تطبيق Agile وجودة مخرجات مشاريع البرمجيات.

### 3.2 دراسة Dybå و Dingsøyr لعام 2008

أجرى الباحثان Dybå و Dingsøyr مراجعة منهجية للدراسات التجريبية المتعلقة بتطوير البرمجيات باستخدام المنهجيات الرشيقة. وقد هدفت الدراسة إلى تقييم مستوى الأدلة العلمية المتعلقة بفوائد Agile، والتعرف على أبرز النتائج التي توصلت إليها الدراسات السابقة في هذا المجال. أظهرت الدراسة أن تطبيق المنهجيات الرشيقة يمكن أن يساهم في تحسين التعاون بين أعضاء الفريق، وزيادة التواصل مع العميل، ورفع مستوى المرونة في التعامل مع التغيرات، وتحسين رضا الأطراف المشاركة في المشروع. كما أشارت إلى أن تقسيم العمل إلى دورات قصيرة يساعد الفرق على اكتشاف المشكلات في مراحل مبكرة، ويقلل من مخاطر استمرار المشروع لفترة طويلة دون التأكد من ملاءمة المنتج لاحتياجات العميل.

ومع ذلك، أوضح الباحثان أن بعض الدراسات التي تناولت Agile كانت تعاني من محدودية في حجم العينات أو ضعف في التصميم المنهجي، ولذلك أكدوا ضرورة التعامل بحذر مع النتائج وعدم افتراض أن المنهجية تحقق النتائج نفسها في جميع الشركات والمشاريع. وتفيد هذه الدراسة البحث الحالي من خلال تأكيدها أن أثر Agile قد يكون إيجابياً، ولكنه يعتمد على طبيعة التطبيق والسياق التنظيمي وجودة الأدلة المستخدمة في التقييم.

### 3.3 دراسة Stavru لعام 2014

تناولت دراسة Stavru مجموعة من المسوح الصناعية التي بحثت في استخدام المنهجيات الرشيقة داخل شركات تطوير البرمجيات. وركز الباحث على تقييم جودة هذه المسوح، ومدى إمكانية الاعتماد على نتائجها في إصدار أحكام عامة حول نجاح Agile. أظهرت الدراسة أن عدداً من التقارير الصناعية يشير إلى أن المنهجيات الرشيقة تساعد على تحسين سرعة التسليم، والتعامل مع المتطلبات المتغيرة، وتعزيز التعاون، ورفع مستوى رضا العملاء.

إلا أن الباحث أشار إلى وجود بعض أوجه القصور في عدد من المسوح، مثل عدم وضوح طريقة اختيار المشاركين، والاعتماد على التقييم الشخصي، وغياب المقارنة مع بيانات موضوعية، وعدم توضيح معنى النجاح أو الجودة بصورة دقيقة. وأكدت الدراسة أهمية عدم الاعتماد فقط على التصورات والانطباعات العامة للعاملين، وضرورة استخدام مؤشرات واضحة وقابلة للقياس، مثل عدد العيوب البرمجية، وزمن معالجة الأخطاء، ورضا المستخدمين، واستقرار النظام. وتدعم هذه الدراسة توجه البحث الحالي نحو عدم اختزال جودة البرمجيات في رضا الفريق أو سرعة الإنجاز، بل ضرورة ربط تطبيق Agile بمؤشرات جودة فنية ووظيفية محددة.

### 3.4دراسة Serrador و Pinto لعام 2015

هدفت دراسة Serrador و Pinto إلى تحليل مدى نجاح المنهجيات الرشيقة في إدارة المشاريع، واعتمدت على تحليل بيانات عدد كبير من المشاريع بلغ أكثر من ألف مشروع في مجالات وبيئات مختلفة.

وأظهرت النتائج وجود علاقة إيجابية بين استخدام الأساليب الرشيقة ونجاح المشروع، سواء من حيث الكفاءة أو من حيث رضا أصحاب المصلحة.

كما بينت الدراسة أن المشاريع التي تستخدم قدرًا أكبر من الممارسات الرشيقة قد تحقق نتائج أفضل في الاستجابة للتغيرات، والتعامل مع احتياجات العميل، وتحسين سرعة اتخاذ القرار.

وأشارت الدراسة إلى أن أثر Agile لا يقتصر على تسريع العمل، بل يمتد إلى تحسين قدرة المشروع على تحقيق أهدافه وتقديم قيمة لأصحاب المصلحة.

ومع ذلك، لم تركز الدراسة بصورة تفصيلية على جميع أبعاد الجودة الفنية للبرمجيات، مثل الأمان والموثوقية والقابلية للصيانة، وهو ما يوضح الحاجة إلى دراسات تربط بين نجاح المشروع العام وجودة المنتج البرمجي بصورة مباشرة.

وتعد هذه الدراسة من الدراسات المهمة التي تدعم الفرضية المتعلقة بوجود أثر إيجابي لتطبيق Agile على نتائج المشاريع، مع الحاجة إلى توسيع مفهوم النتائج ليشمل جودة البرمجيات.

### 3.5دراسة Dikert وزملائه لعام 2016

أجرى Dikert وزملاؤه مراجعة منهجية حول التحولات الرشيقة واسعة النطاق داخل المؤسسات الكبيرة. وركزت الدراسة على تحديد التحديات التي تواجه الشركات أثناء الانتقال إلى Agile، وعوامل النجاح التي تساعد على تحقيق نتائج أفضل.

توصلت الدراسة إلى وجود عدد كبير من التحديات، من أبرزها مقاومة التغيير، وضعف دعم الإدارة العليا، واستمرار الثقافة التنظيمية التقليدية، وعدم وضوح الأدوار، وضعف التدريب، وصعوبة التنسيق بين الفرق.

كما أظهرت الدراسة أن أهم عوامل النجاح تتمثل في دعم الإدارة العليا، ووجود رؤية واضحة للتحويل، وتدريب الموظفين، وتخصيص المنهجية بما يتناسب مع احتياجات المؤسسة، وإشراك الأفراد في التغيير، وتحسين التواصل بين الإدارات.

وأوضحت الدراسة أن تطبيق Agile داخل المؤسسات الكبيرة لا ينجح بمجرد اعتماد إطار أو مجموعة من الاجتماعات، بل يحتاج إلى تغيير تنظيمي وثقافي متكامل.

وتفيد هذه الدراسة البحث الحالي في توضيح أن جودة المخرجات لا تعتمد على الممارسات الفنية فقط، بل تتأثر أيضًا بعوامل القيادة والثقافة والتدريب والهيكل التنظيمي.

### 3.6دراسة Kuhrmann وزملائه لعام 2021

تناولت دراسة Kuhrmann وزملائه الممارسات المستخدمة فعليًا في تطوير البرمجيات داخل عدد كبير من الشركات والفرق حول العالم، وهدفت إلى معرفة ما الذي يجعل طريقة العمل رشيقة بالفعل.

أظهرت الدراسة أن كثيرًا من الشركات لا تستخدم منهجية رشيقة خالصة، بل تعتمد مزيجًا من الممارسات الرشيقة والتقليدية، وفقًا لطبيعة المشروع واحتياجات المنظمة.

كما بينت أن تصنيف المشروع بأنه Agile لا يعني بالضرورة أن جميع الممارسات الرشيقة مطبقة بصورة صحيحة. فقد تستخدم بعض الشركات Scrum أو Kanban من حيث الشكل، لكنها تستمر في اتخاذ القرارات بصورة مركزية أو تؤخر الاختبارات أو لا تشرك العميل بصورة مستمرة.

وأشارت النتائج إلى أن درجة الرشفقة الحقيقية تعتمد على الممارسات الفعلية، مثل التسليم المتكرر، والتعاون، والشفافية، وتمكين الفريق، والاستجابة للتغيير، أكثر من اعتمادها على اسم الإطار المستخدم. وتؤكد هذه الدراسة أهمية قياس مستوى تطبيق الممارسات الرشيفة بصورة فعلية، وعدم الاكتفاء بسؤال الشركات عما إذا كانت تستخدم Agile أم لا.

### 3.7 بيان Agile لعام 2001

يمثل بيان Agile المرجعية الأساسية التي انطلقت منها المنهجيات الرشيفة في تطوير البرمجيات. وقد تضمن أربع قيم رئيسة واثنى عشر مبدأ تهدف إلى تحسين طريقة إدارة وتطوير البرمجيات. ركز البيان على الأفراد والتفاعلات، والبرمجيات العاملة، والتعاون مع العميل، والاستجابة للتغيير. وتظهر أهمية البيان بالنسبة لموضوع الجودة في تركيزه على التسليم المستمر للبرمجيات ذات القيمة، والاهتمام بالتميز الفني، والتصميم الجيد، والتعاون اليومي، ومراجعة طريقة العمل بصورة دورية. ويؤكد البيان أن الرشفقة لا تعني التخلي عن الجودة أو التخطيط، بل تهدف إلى تحقيق قيمة أفضل من خلال أسلوب عمل مرن ومستمر. ويعد هذا البيان أساساً نظرياً مهماً للدراسة الحالية، لأنه يوضح أن الجودة جزء من فلسفة Agile وليست عنصراً منفصلاً عنها.

### 3.8 دليل Scrum لعام 2020

يقدم دليل Scrum الإطار الرسمي لتطبيق Scrum ، ويوضح الأدوار والأحداث والمخرجات التي يعتمد عليها الفريق في إدارة العمل المعقد. يركز Scrum على ثلاثة أسس رئيسة هي الشفافية، والفحص، والتكيف. وتعني الشفافية أن تكون حالة العمل واضحة، بينما يعني الفحص مراجعة المخرجات وطريقة العمل بصورة منتظمة، أما التكيف فيعني تعديل الخطط أو الإجراءات عندما تظهر مشكلات أو معلومات جديدة. ويرتبط دليل Scrum بجودة البرمجيات من خلال مفهوم الزيادة القابلة للاستخدام، ومفهوم تعريف الإنجاز. فلا ينبغي اعتبار الوظيفة مكتملة إلا إذا حققت المعايير المتفق عليها، مثل الاختبار والمراجعة والتوثيق والجاهزية للاستخدام. كما تساعد مراجعة الدورة والاجتماع الاستعادي على تحسين المنتج وطريقة العمل، مما يدعم الجودة بصورة مستمرة. وتفيد هذه المرجعية الدراسة الحالية في توضيح أن الجودة يجب أن تكون جزءاً من معايير الإنجاز داخل كل دورة، لا نشاطاً موقلاً إلى نهاية المشروع.

### 3.9 دليل الممارسات الرشيفة الصادر عن معهد إدارة المشاريع

أصدر معهد إدارة المشاريع بالتعاون مع Agile Alliance دليل الممارسات الرشيفة، بهدف مساعدة المؤسسات ومديري المشاريع على اختيار وتطبيق الأساليب الرشيفة المناسبة. أوضح الدليل أن اختيار المنهجية يجب أن يعتمد على طبيعة المشروع، ومستوى عدم اليقين، واستقرار المتطلبات، وسرعة التغيير، وقدرة الفريق والعمل على التعاون. كما عرض الدليل مجموعة من النماذج الرشيفة والهجينة، وأكد أنه لا توجد طريقة واحدة مناسبة لجميع المشاريع. ويشير الدليل إلى أهمية القيادة الخادمة، وتمكين الفرق، والتعلم المستمر، وإدارة تدفق القيمة، والتركيز على احتياجات العميل. ويفيد هذا الدليل في دعم فكرة أن تحقيق الجودة يتطلب اختيار الممارسات المناسبة للسياق، وليس تطبيق نموذج موحد على جميع الشركات والمشاريع.

### 3.10 معيار ISO/IEC 25010:2023

يعد معيار ISO/IEC 25010 من أهم المعايير الدولية المستخدمة في تحديد خصائص جودة المنتجات البرمجية والأنظمة. ويقدم المعيار نموذجًا شاملاً للجودة يضم مجموعة من الخصائص، مثل الملاءمة الوظيفية، وكفاءة الأداء، والتوافق، وقابلية التفاعل، والموثوقية، والأمان، والقابلية للصيانة، والمرونة، والسلامة. وتكمن أهمية المعيار في أنه يوفر إطارًا واضحًا يمكن للشركات استخدامه عند تحديد معنى الجودة في المنتج البرمجي. كما يساعد على تحويل مفهوم الجودة من مفهوم عام وغير محدد إلى مجموعة من الأبعاد التي يمكن قياسها ومتابعتها. وتستفيد الدراسة الحالية من هذا المعيار في تحديد المتغير التابع، وهو جودة مخرجات مشاريع البرمجيات، وربطه بالممارسات الرشيدة.

### 3.11 أوجه الاتفاق بين الدراسات السابقة

تتفق الدراسات والمرجعيات السابقة على مجموعة من النقاط الرئيسية. أولاً، يمكن أن تسهم Agile في تحسين التعاون والتواصل بين أعضاء الفريق والعميل. ثانياً، يساعد التسليم التكراري على اكتشاف المشكلات والتغيرات في وقت مبكر. ثالثاً، تسهم التغذية الراجعة المستمرة في تحسين توافق المنتج مع احتياجات المستخدمين. رابعاً، يحتاج تطبيق Agile إلى دعم إداري وثقافة تنظيمية مناسبة وتمكين للفرق. خامساً، لا يتحقق النجاح من خلال الاجتماعات والمصطلحات فقط، بل من خلال تطبيق الممارسات بصورة فعلية. سادساً، لا توجد طريقة واحدة مناسبة لجميع المشاريع، وقد يكون النهج الهجين أكثر ملاءمة لبعض المؤسسات. سابغاً، يجب ربط نجاح Agile بمؤشرات واضحة، وعدم الاعتماد فقط على الانتباعات أو سرعة الإنجاز.

### 3.12 أوجه الاختلاف بين الدراسات السابقة

على الرغم من اتفاق الدراسات على كثير من الفوائد، فإنها تختلف في بعض الجوانب. فبعض الدراسات ركزت على نجاح المشروع بصورة عامة، مثل الوقت والتكلفة ورضا أصحاب المصلحة، بينما ركزت دراسات أخرى على جودة المنتج أو جودة العملية. كما اختلفت الدراسات في طبيعة العينات، وحجم الشركات، ونوع المشاريع، والمناطق الجغرافية التي تناولتها. واعتمدت بعض الدراسات على مسوح واستبيانات وآراء العاملين، بينما استخدمت دراسات أخرى المراجعة المنهجية أو تحليل المشاريع الفعلية. كما تختلف تعريفات Agile بين الدراسات؛ فبعضها يقيسها وفق استخدام Scrum أو Kanban، بينما تقيسها دراسات أخرى وفق الممارسات والسلوكيات الفعلية. ويؤثر هذا الاختلاف في إمكانية مقارنة النتائج بصورة مباشرة، ويؤكد الحاجة إلى تعريفات تشغيلية واضحة لكل من Agile وجودة البرمجيات.

### 3.13 التحليل النقدي للدراسات السابقة

أظهر تحليل الدراسات السابقة أن معظم الأدلة تشير إلى أثر إيجابي لمنهجية Agile، إلا أن قوة هذا الأثر تختلف حسب جودة التطبيق والسياق التنظيمي.

ومن أبرز الملاحظات النقدية أن بعض الدراسات تعتمد على التقييم الذاتي للعاملين، وهو ما قد يؤدي إلى نتائج متأثرة بالتصورات الشخصية. كما أن بعض الدراسات تقيس نجاح Agile من خلال سرعة التسليم أو رضا الفريق، دون قياس مباشر لمؤشرات الجودة الفنية. وقد يتم اعتبار المشروع ناجحاً لأنه تم تسليمه بسرعة، على الرغم من وجود مشكلات في الموثوقية أو القابلية للصيانة أو الأمان.

كما أن كثيرًا من الدراسات لا تفرق بصورة كافية بين تطبيق المبادئ الرشيقة وتطبيق إطار معين من حيث الشكل. وقد تستخدم شركة Scrum ، لكنها لا تمنح الفريق الصلاحيات اللازمة، أو لا تشرك العميل، أو لا تنفذ الاختبارات المستمرة، مما يجعل التطبيق محدود التأثير. ومن الملاحظ كذلك أن جزءًا كبيرًا من الدراسات أُجري في دول وبيئات غربية، بينما ما تزال الدراسات التطبيقية في البيئة العربية محدودة نسبيًا.

### 3.14 الثغرة البحثية

تتمثل الثغرة البحثية في أن عددًا من الدراسات تناول أثر Agile على نجاح المشاريع بصورة عامة، لكنها لم تربط بصورة مباشرة ومتكاملة بين الممارسات الرشيقة المحددة وأبعاد جودة البرمجيات. كما أن بعض الدراسات استخدمت مؤشرات عامة، مثل رضا العميل أو سرعة الإنجاز، دون تحليل أثر Agile في خصائص مثل الموثوقية والأمان والأداء والقابلية للصيانة. وتوجد حاجة أيضًا إلى إطار يوضح أن أثر Agile لا يعتمد على المنهجية وحدها، بل يتأثر بعوامل وسيطة وداعمة، مثل دعم الإدارة، وتمكين الفرق، والممارسات الهندسية، والأتمتة، والثقافة التنظيمية. كما تظهر الحاجة إلى تقديم نموذج عملي يمكن للشركات التقنية الاستفادة منه عند تطبيق Agile ، بحيث يربط بين الممارسات الرشيقة ومؤشرات الجودة وخطوات التنفيذ. وتسعى الدراسة الحالية إلى معالجة هذه الثغرة من خلال تحليل العلاقة بين Agile وجودة مخرجات البرمجيات، وبناء نموذج تطبيقي يجمع بين الجوانب الإدارية والفنية.

### 3.15 النموذج المفاهيمي للدراسة

يقوم النموذج المفاهيمي للدراسة على وجود علاقة بين مستوى تطبيق منهجية الإدارة الرشيقة وجودة مخرجات مشاريع البرمجيات. يمثل تطبيق Agile المتغير المستقل في الدراسة، ويشمل مجموعة من الأبعاد، من أهمها:

1. التخطيط التكراري .
2. التسليم التدريجي .
3. التعاون مع العميل .
4. تمكين فريق العمل .
5. التحسين المستمر .
6. دمج الاختبار والجودة داخل دورة التطوير .
7. التكامل المستمر والأتمتة .

أما جودة مخرجات مشاريع البرمجيات فتُمثل المتغير التابع، وتشمل:

1. الملاءمة الوظيفية .
2. الموثوقية .
3. كفاءة الأداء .
4. سهولة الاستخدام .
5. الأمان .
6. القابلية للصيانة .

## 7. التوافق والمرونة .

ويتأثر مستوى العلاقة بين المتغيرين بمجموعة من العوامل الداعمة أو المعيقة، مثل:

1. دعم الإدارة العليا .
2. الثقافة التنظيمية .
3. خبرة فريق العمل .
4. وضوح الأدوار .
5. مشاركة العميل .
6. توفر الأدوات التقنية .
7. مستوى الأتمتة .
8. حجم الشركة وتعقيد المشروع .

وبناءً على هذا النموذج، فإن تطبيق Agile بصورة متكاملة ومتوازنة يُتوقع أن يؤدي إلى تحسين جودة مخرجات البرمجيات، بينما قد يؤدي التطبيق الجزئي أو الشكلي إلى نتائج محدودة.

## 3.16 العلاقة بين الدراسات السابقة وفرضيات الدراسة

تدعم الدراسات السابقة الفرضية الأولى التي تنص على وجود أثر إيجابي لتطبيق Agile على جودة مخرجات مشاريع البرمجيات. كما تدعم نتائج الدراسات المتعلقة بالتسليم التكراري والتعاون مع العميل الفرضية الثانية، التي تشير إلى دور التغذية الراجعة المبكرة في تحسين الملاءمة الوظيفية.

وتدعم المراجع المتعلقة بالبرمجة القصوى و DevOps الفرضية الثالثة، التي تربط الممارسات الهندسية بتحسين الموثوقية والقابلية للصيانة. أما الدراسات المتعلقة بالتحويلات واسعة النطاق فتدعم الفرضية الرابعة، التي تشير إلى أهمية دعم الإدارة العليا وتمكين الفرق. كما تدعم نتائج الدراسات النقدية الفرضية الخامسة، التي تفترض أن التطبيق الشكلي أو الجزئي يؤدي إلى ضعف أثر Agile في الجودة.

## 3.17 خلاصة الفصل

تناول هذا الفصل مجموعة من الدراسات والمرجعيات العلمية والمهنية المرتبطة بتطبيق منهجية الإدارة الرشيدة وجودة مخرجات مشاريع البرمجيات.

وأظهرت الدراسات وجود اتجاه عام يدعم الأثر الإيجابي لـ Agile في التعاون، والمرونة، ورضا العميل، والاستجابة للتغيير، ونجاح المشاريع. كما أوضحت أن تحقيق هذه الفوائد يعتمد على عوامل متعددة، مثل دعم الإدارة، وتمكين الفرق، وجودة التطبيق، والممارسات الهندسية، ومشاركة العميل.

وأظهر التحليل النقدي أن بعض الدراسات ركزت على سرعة الإنجاز أو النجاح العام، دون قياس تفصيلي لأبعاد جودة البرمجيات.

## الفصل الرابع: منهجية الدراسة

### 4.1 تمهيد

يتناول هذا الفصل المنهجية التي اعتمدت عليها الدراسة في تحليل أثر تطبيق منهجية الإدارة الرشيقة Agile على جودة مخرجات مشاريع البرمجيات في الشركات التقنية.

وتشمل منهجية الدراسة تحديد نوع البحث، والمنهج المستخدم، ومصادر البيانات، وطريقة اختيار المراجع، وأداة التحليل، وخطوات تحليل المعلومات، إضافة إلى توضيح الاعتبارات الأخلاقية والقيود المنهجية المرتبطة بالدراسة.

وقد تم اختيار منهجية تتناسب مع طبيعة موضوع البحث والوقت المتاح، مع الالتزام بعدم إدراج أي بيانات ميدانية أو نتائج غير موثقة.

### 4.2 نوع الدراسة

تعد هذه الدراسة من الدراسات الوصفية التحليلية، لأنها تهدف إلى وصف مفهوم الإدارة الرشيقة وممارساتها، وتوضيح مفهوم جودة البرمجيات وأبعادها، ثم تحليل العلاقة بينهما بالاعتماد على الأدبيات العلمية والمعايير المهنية.

ولا تقتصر الدراسة على عرض المعلومات بصورة وصفية فقط، بل تسعى إلى مقارنة نتائج الدراسات السابقة، وتحليل أوجه الاتفاق والاختلاف بينها، واستخلاص العوامل التي تؤثر في نجاح تطبيق Agile، ثم بناء نموذج تطبيقي يمكن للشركات التقنية الاستفادة منه.

### 4.3 المنهج المستخدم

اعتمدت الدراسة على منهج المراجعة التكاملية للأدبيات، وهو منهج يهدف إلى جمع وتحليل وتفسير مجموعة متنوعة من المصادر العلمية والمهنية المرتبطة بموضوع الدراسة.

ويسمح هذا المنهج بدمج نتائج الدراسات التجريبية والمراجعات المنهجية والمعايير الدولية والأدلة المهنية، بهدف تكوين رؤية شاملة حول أثر تطبيق Agile على جودة مخرجات البرمجيات.

وقد تم اختيار هذا المنهج للأسباب الآتية:

1. ملاءمته لطبيعة موضوع الدراسة .
2. إتاحته للاستفادة من عدد كبير من الدراسات والمصادر السابقة .
3. قدرته على الربط بين الجوانب الإدارية والفنية .
4. إمكانية استخدامه في بناء نموذج تطبيقي وتوصيات عملية .
5. عدم توفر بيانات ميدانية موثقة يمكن الاعتماد عليها ضمن الوقت المحدد للمشروع .
6. تجنب اختلاق نتائج استبانة أو مقابلات لم يتم تنفيذها فعلياً.

### 4.4 مجتمع الدراسة

يتمثل مجتمع الدراسة النظري في الأدبيات العلمية والمهنية التي تناولت الموضوعات الآتية:

1. منهجية الإدارة الرشيقة Agile.
2. إطار Scrum.
3. منهجية Kanban.
4. البرمجة القصوى Extreme Programming.
5. منهج DevOps.

6. جودة البرمجيات .
  7. نجاح مشاريع البرمجيات .
  8. التحولات الرشيقة في الشركات التقنية .
  9. المعايير الدولية لجودة المنتجات البرمجية .
- ولا يقصد بمجتمع الدراسة في هذا البحث مجموعة من الأفراد أو الشركات التي تم جمع بيانات مباشرة منها، بل يقصد به مجموعة المصادر والمراجع التي تم تحليلها.

#### 4.5 عينة الدراسة

تكونت عينة الدراسة من مجموعة مختارة من المصادر العلمية والمهنية ذات الصلة المباشرة بموضوع البحث. وشملت العينة:

1. دراسات علمية محكمة تناولت أثر Agile على نجاح المشاريع أو جودة البرمجيات .
  2. مراجعات منهجية حول استخدام المنهجيات الرشيقة .
  3. دراسات تناولت عوامل نجاح التحول الرشيق والتحديات المرتبطة به .
  4. بيان Agile ومبادئه الأساسية .
  5. دليل Scrum الرسمي .
  6. دليل الممارسات الرشيقة الصادر عن معهد إدارة المشاريع .
  7. معيار ISO/IEC 25010:2023 لجودة المنتجات البرمجية .
  8. مراجع أكاديمية في إدارة المشاريع وهندسة البرمجيات .
- وقد تم اختيار المصادر وفق معايير محددة، لضمان ارتباطها بموضوع الدراسة وجودتها العلمية.

#### 4.6 مصادر البيانات

اعتمدت الدراسة على البيانات الثانوية، وهي البيانات التي سبق جمعها أو نشرها من قبل باحثين أو جهات علمية ومهنية. وتضمنت مصادر البيانات ما يأتي:

##### 4.6.1 الدراسات العلمية المحكمة

تم الاعتماد على الدراسات المنشورة في المجلات العلمية المتخصصة في إدارة المشاريع، وهندسة البرمجيات، ونظم المعلومات، وتطوير البرمجيات.

وتتميز هذه الدراسات بخضوعها للمراجعة العلمية قبل النشر، مما يزيد من مستوى موثوقيتها.

##### 4.6.2 المراجعات المنهجية

تم استخدام المراجعات المنهجية التي جمعت وحللت عددًا كبيرًا من الدراسات السابقة حول Agile ، وذلك للاستفادة من نتائجها الشاملة.

##### 4.6.3 الأدلة المهنية الرسمية

اعتمدت الدراسة على مجموعة من الأدلة المهنية، مثل دليل الممارسات الرشيقة الصادر عن معهد إدارة المشاريع، ودليل Scrum.

##### 4.6.4 المعايير الدولية

تم استخدام معيار ISO/IEC 25010:2023 بوصفه مرجعًا لتحديد أبعاد جودة البرمجيات وخصائصها.

##### 4.6.5 المراجع الأكاديمية

تم الاستفادة من الكتب والمراجع الأكاديمية المرتبطة بإدارة المشاريع وتطوير البرمجيات وجودة الأنظمة.

#### 4.7 معايير اختيار المصادر

تم اختيار المصادر المستخدمة في الدراسة وفق مجموعة من المعايير، من أهمها:

1. ارتباط المصدر المباشر بموضوع Agile أو جودة البرمجيات .
2. صدور المصدر عن مجلة علمية أو جهة مهنية أو مؤسسة معيارية معروفة .
3. وضوح منهجية الدراسة أو المرجع .
4. احتواء المصدر على نتائج أو مفاهيم قابلة للتحليل والمقارنة .
5. استخدام مصادر أصلية قدر الإمكان .
6. الاستفادة من الدراسات الحديثة إلى جانب المراجع التأسيسية المهمة .
7. توفر المعلومات اللازمة لتوثيق المصدر بصورة صحيحة .
8. ارتباط النتائج ببيئة مشاريع البرمجيات أو الشركات التقنية .

#### 4.8 معايير استبعاد المصادر

تم استبعاد المصادر التي تنطبق عليها واحدة أو أكثر من الحالات الآتية:

1. عدم وجود علاقة مباشرة بينها وبين موضوع الدراسة .
2. عدم وضوح الجهة الناشرة أو المؤلف .
3. اعتماد المحتوى على آراء غير موثقة أو ادعاءات تسويقية .
4. عدم توضيح المنهجية أو طريقة الوصول إلى النتائج .
5. تكرار المعلومات نفسها دون إضافة علمية واضحة .
6. استخدام نتائج لا يمكن التحقق من مصدرها .
7. التركيز على مجالات بعيدة عن مشاريع البرمجيات دون إمكانية الاستفادة منها .

#### 4.9 أداة الدراسة

تمثلت أداة الدراسة في مصفوفة تحليل محتوى، استخدمت لتنظيم المعلومات المستخرجة من المصادر ومقارنتها.

وتضمنت المصفوفة مجموعة من العناصر، منها:

1. اسم الباحث أو الجهة .
2. سنة النشر .
3. عنوان الدراسة أو المرجع .
4. نوع المصدر .
5. هدف الدراسة .
6. المنهج المستخدم .
7. طبيعة البيئة أو العينة .
8. الممارسات الرشيدة التي تناولها المصدر .

9. أبعاد الجودة أو النجاح التي تناولها .

10. أبرز النتائج .

11. عوامل النجاح .

12. التحديات والقيود .

13. مدى ارتباط المصدر بفرضيات الدراسة .

14. الدلالة التطبيقية التي يمكن الاستفادة منها .

وقد ساعدت هذه المصنوفة على تنظيم عملية التحليل، وتقليل الاعتماد على التلخيص العام غير المنظم.

4.10 خطوات إجراء الدراسة

تم تنفيذ الدراسة وفق مجموعة من الخطوات المنظمة، وذلك على النحو الآتي:

الخطوة الأولى: تحديد موضوع الدراسة

تم تحديد الموضوع في أثر تطبيق منهجية الإدارة الرشيقة على جودة مخرجات مشاريع البرمجيات في الشركات التقنية.

الخطوة الثانية: تحديد المشكلة والأسئلة

تمت صياغة مشكلة الدراسة، والسؤال الرئيس، والأسئلة الفرعية، بما يساعد على توجيه البحث واختيار المعلومات ذات الصلة.

الخطوة الثالثة: تحديد المفاهيم الأساسية

تم تحديد المفاهيم الرئيسية، وهي:

- الإدارة الرشيقة .
- ممارسات Agile.
- جودة البرمجيات .
- مخرجات المشاريع .
- عوامل النجاح التنظيمية والفنية .

الخطوة الرابعة: جمع المصادر

تم جمع مجموعة من الدراسات والمراجع والأدلة المهنية والمعايير الدولية المرتبطة بموضوع الدراسة.

الخطوة الخامسة: فرز المصادر

تمت مراجعة المصادر واستبعاد المحتوى غير المرتبط أو غير الموثوق أو غير القابل للتحليل.

الخطوة السادسة: استخراج المعلومات

تم استخراج المعلومات الرئيسية من كل مصدر، مثل الأهداف والمنهج والنتائج والعوامل المؤثرة والقيود.

الخطوة السابعة: تصنيف النتائج

تم تصنيف نتائج المصادر ضمن محاور رئيسية، هي:

1. أثر Agile على جودة البرمجيات .

2. أثر التغذية الراجعة والتسليم التكراري .

3. أثر الممارسات الهندسية .

4. أثر القيادة والثقافة التنظيمية .

5. تحديات التطبيق .

6. مؤشرات قياس الجودة .

#### الخطوة الثامنة: المقارنة والتحليل

تمت مقارنة النتائج لتحديد نقاط الاتفاق والاختلاف، وتحليل الأسباب التي قد تفسر تفاوت نتائج تطبيق Agile بين الشركات.

#### الخطوة التاسعة: بناء النموذج المقترح

تم تطوير نموذج عملي يربط بين ممارسات Agile وأبعاد الجودة والعوامل التنظيمية والفنية.

#### الخطوة العاشرة: صياغة التوصيات

تم إعداد مجموعة من التوصيات العملية المبنية على نتائج التحليل، بهدف مساعدة الشركات التقنية على تحسين تطبيق المنهجية.

#### 4.11 طريقة تحليل البيانات

اعتمدت الدراسة على أسلوب التحليل النوعي للمحتوى، وذلك من خلال قراءة المصادر وتحليل الأفكار والنتائج الواردة فيها.

وتضمنت عملية التحليل ما يأتي:

1. تحديد العبارات والنتائج المرتبطة بأثر Agile.
  2. تصنيف النتائج حسب موضوعها .
  3. مقارنة النتائج بين الدراسات .
  4. تحديد النتائج المتكررة بين أكثر من مصدر .
  5. تحليل التفسيرات المحتملة للاختلاف بين الدراسات .
  6. ربط النتائج بفرضيات الدراسة .
  7. ربط الممارسات الرشيدة بأبعاد جودة البرمجيات .
  8. استخلاص دلالات تطبيقية للشركات التقنية .
- ولم تعتمد الدراسة على التحليل الإحصائي، لعدم وجود بيانات ميدانية جديدة، وإنما اعتمدت على التحليل الوصفي والنقدي للمصادر.

#### 4.12 المتغيرات الرئيسية في الدراسة

تتضمن الدراسة متغيرين رئيسيين، إضافة إلى مجموعة من العوامل الداعمة.

##### 4.12.1 المتغير المستقل

يتمثل المتغير المستقل في مستوى تطبيق منهجية الإدارة الرشيدة.

ويشمل مجموعة من الأبعاد، منها:

1. التطوير التكراري .
2. التسليم التدريجي .
3. مشاركة العميل .
4. تمكين الفريق .
5. التعاون المستمر .
6. التحسين المستمر .
7. الاختبارات الآلية .
8. التكامل المستمر .
9. مراجعة الشفرة البرمجية .

#### 10. تعريف الإنجاز .

##### 4.12.2 المتغير التابع

يتمثل المتغير التابع في جودة مخرجات مشاريع البرمجيات.

ويشمل:

1. الملاءمة الوظيفية .
2. كفاءة الأداء .
3. الموثوقية .
4. سهولة الاستخدام .
5. الأمان .
6. القابلية للصيانة .
7. التوافق .
8. المرونة .

##### 4.12.3 العوامل الداعمة أو المعيقة

تشمل العوامل التي قد تؤثر في العلاقة بين المتغيرين ما يأتي:

1. دعم الإدارة العليا .
2. الثقافة التنظيمية .
3. خبرة الفريق .
4. وضوح الأدوار .
5. مشاركة العميل .
6. توفر الأدوات التقنية .
7. مستوى الأتمتة .
8. حجم الشركة .
9. تعقيد المشروع .
10. مستوى الاعتماد بين الفرق.

#### 4.13 صدق الدراسة

يشير الصدق إلى مدى قدرة المنهج والأداة المستخدمة على دراسة الموضوع بصورة مناسبة والوصول إلى نتائج منطقية.

وقد تم تعزيز صدق الدراسة من خلال:

1. الاعتماد على مصادر علمية ومهنية متنوعة .
2. استخدام دراسات محكمة ومعايير رسمية .
3. مقارنة نتائج أكثر من مصدر .
4. عدم الاعتماد على مصدر واحد في استخلاص النتائج .
5. استخدام تعريفات واضحة للمفاهيم الأساسية .
6. ربط النتائج بنموذج جودة دولي .

7. توضيح حدود الدراسة وعدم المبالغة في تعميم النتائج .

8. الفصل بين الأدلة التجريبية والتوصيات المهنية.

#### 4.14 ثبات الدراسة

يشير الثبات إلى إمكانية تطبيق خطوات التحليل نفسها على المصادر والوصول إلى نتائج مقاربة.

وقد تم دعم ثبات الدراسة من خلال:

1. وضع معايير واضحة لاختيار واستبعاد المصادر .

2. استخدام مصفوفة ثابتة لتحليل المحتوى .

3. تصنيف النتائج ضمن محاور محددة مسبقاً .

4. استخدام تعريفات تشغيلية ثابتة للمتغيرات .

5. توثيق خطوات جمع وتحليل المعلومات .

6. مراجعة النتائج ومقارنتها بفرضيات الدراسة.

#### 4.15 الاعتبارات الأخلاقية

التزمت الدراسة بمجموعة من المبادئ الأخلاقية في البحث العلمي، ومن أهمها:

##### 4.15.1 الأمانة العلمية

تم توثيق الأفكار والمعلومات المستفادة من المصادر، وعدم نسب نتائج الباحثين إلى الدراسة الحالية.

##### 4.15.2 عدم اختلاق البيانات

لم يتم إدراج أي نتائج استنبأه أو مقابلات أو بيانات شركات لم يتم جمعها فعلياً.

##### 4.15.3 الدقة في عرض النتائج

تم تجنب المبالغة في تفسير أثر Agile ، والتأكيد على أن النتائج تختلف باختلاف السياق وجودة التطبيق.

##### 4.15.4 المحافظة على حقوق الملكية الفكرية

تمت الإشارة إلى المصادر المستخدمة في قائمة المراجع، والالتزام بصياغة المحتوى بأسلوب الباحث بدل النسخ المباشر المطول.

##### 4.15.5 الشفافية

تم توضيح أن الدراسة تعتمد على بيانات ثانوية ومراجعة للأدبيات، ولا تتضمن دراسة ميدانية مباشرة.

##### 4.15.6 حماية المشاركين مستقبلاً

في حال تنفيذ دراسة ميدانية لاحقة، يجب الحصول على موافقة المشاركين، والحفاظ على سرية بياناتهم، وعدم ذكر أسماء الأفراد أو الشركات دون موافقة.

#### 4.16 قيود الدراسة

تواجه الدراسة مجموعة من القيود التي ينبغي أخذها في الاعتبار عند تفسير النتائج.

##### 4.16.1 الاعتماد على البيانات الثانوية

تعتمد الدراسة على مصادر منشورة سابقاً، ولا تتضمن قياساً مباشراً داخل شركة تقنية محددة.

##### 4.16.2 اختلاف تعريف Agile

تختلف الدراسات في تعريف مستوى تطبيق Agile ، فقد تعتمد بعض الدراسات على اسم الإطار المستخدم، بينما تعتمد دراسات أخرى على الممارسات الفعلية.

#### 4.16.3 اختلاف مفهوم الجودة

تستخدم الدراسات مؤشرات مختلفة للجودة، مثل رضا العميل، وعدد العيوب، والموثوقية، وسرعة التسليم، مما قد يصعب المقارنة المباشرة.

#### 4.16.4 اختلاف البيانات التنظيمية

قد تختلف نتائج تطبيق Agile باختلاف حجم الشركة، ونوع المنتج، وخبرة الفريق، والثقافة التنظيمية.

#### 4.16.5 محدودية الدراسات العربية

ما تزال الدراسات التطبيقية في البيئة العربية حول أثر Agile على جودة البرمجيات محدودة مقارنة بالدراسات الدولية.

#### 4.16.6 عدم إثبات العلاقة السببية

تساعد المراجعة التحليلية على تحديد الاتجاهات والعلاقات المحتملة، لكنها لا تثبت بصورة نهائية أن Agile هي السبب الوحيد في تحسين الجودة.

#### 4.17 مبررات عدم استخدام الاستبانة

لم تستخدم الدراسة استبانة ميدانية، وذلك للأسباب الآتية:

1. ضيق الفترة الزمنية المتاحة لتنفيذ الدراسة .
2. الحاجة إلى الحصول على موافقات ومشاركين مناسبين .
3. ضرورة اختبار صدق وثبات الاستبانة قبل اعتماد نتائجها .
4. صعوبة الحصول على بيانات موثوقة من عدد كافٍ من الشركات التقنية خلال الوقت المتاح .
5. عدم جواز إدراج نتائج افتراضية أو غير حقيقية .
6. ملائمة منهج المراجعة التكاملية لأهداف الدراسة الحالية .

ولا يقلل عدم استخدام الاستبانة من قيمة الدراسة، ما دامت المنهجية موضحة، والمصادر موثوقة، والنتائج مبنية على تحليل علمي منظم.

#### 4.18 إمكانية تطوير الدراسة مستقبلاً

يمكن تطوير الدراسة مستقبلاً من خلال تنفيذ دراسة ميدانية تعتمد على استبانة أو مقابلات مع العاملين في الشركات التقنية.

ويمكن أن تشمل العينة:

1. مديري المشاريع .
2. مالكي المنتجات .
3. مطوري البرمجيات .
4. مهندسي الجودة والاختبار .
5. مسؤولي DevOps .
6. مديري التقنية .

ويمكن تحليل البيانات باستخدام برامج إحصائية لاختبار العلاقة بين مستوى تطبيق Agile وأبعاد جودة البرمجيات.

كما يمكن مقارنة النتائج بين الشركات الصغيرة والكبيرة، أو بين الشركات المحلية والدولية، أو بين المشاريع التي تستخدم منهجيات رشيدة خالصة والمشاريع التي تستخدم نهجاً هجيناً.

#### 4.19 خلاصة الفصل

تناول هذا الفصل منهجية الدراسة، حيث تم توضيح أن البحث يعتمد على المنهج الوصفي التحليلي والمراجعة التكاملية للأدبيات. كما تم توضيح مجتمع الدراسة وعينتها ومصادر البيانات ومعايير اختيار واستبعاد المراجع، إضافة إلى أداة تحليل المحتوى والخطوات المستخدمة في جمع المعلومات وتحليلها.

وتناول الفصل المتغيرات الرئيسية في الدراسة، وطرق تعزيز الصدق والثبات، والاعتبارات الأخلاقية، والقيود المنهجية. وقد تم التأكيد على أن الدراسة لا تتضمن بيانات ميدانية أو نتائج استبانة غير منفذة، بل تعتمد على تحليل علمي منظم للمصادر السابقة. وسيتناول الفصل الخامس نتائج التحليل ومناقشتها، وربطها بفرضيات الدراسة وأبعاد جودة مخرجات مشاريع البرمجيات.

## الفصل الخامس: نتائج الدراسة ومناقشتها

### 5.1 تمهيد

يتناول هذا الفصل نتائج التحليل الذي أجري على الدراسات السابقة والمراجع العلمية والأدلة المهنية المرتبطة بتطبيق منهجية الإدارة الرشيدة Agile وجودة مخرجات مشاريع البرمجيات. وقد تم تنظيم النتائج ضمن مجموعة من المحاور الرئيسية، تشمل أثر Agile على جودة البرمجيات، ودور التسليم التكراري والتغذية الراجعة، وأهمية الممارسات الهندسية، وتأثير دعم الإدارة والثقافة التنظيمية، إضافة إلى أبرز التحديات التي قد تحد من نجاح التطبيق. كما يتضمن الفصل مناقشة فرضيات الدراسة، وربط النتائج بالإطار النظري والدراسات السابقة، وتقديم مجموعة من مؤشرات الأداء التي يمكن استخدامها لقياس أثر Agile على جودة مخرجات المشاريع البرمجية.

### 5.2 النتيجة الأولى: وجود أثر إيجابي عام لتطبيق Agile على جودة مخرجات البرمجيات

أظهر تحليل الدراسات السابقة وجود اتجاه عام يشير إلى أن تطبيق منهجية Agile بصورة صحيحة يمكن أن يسهم في تحسين جودة مخرجات مشاريع البرمجيات.

ويظهر هذا التحسن من خلال عدد من الجوانب، من أهمها:

1. اكتشاف الأخطاء في مراحل مبكرة .
2. تحسين توافق المنتج مع احتياجات العميل .
3. زيادة سرعة الاستجابة للتغيرات .
4. تعزيز التعاون بين أعضاء الفريق .
5. رفع مستوى الشفافية في سير العمل .
6. تقليل مخاطر اكتشاف المشكلات بعد اكتمال المشروع .
7. تحسين رضا أصحاب المصلحة عن المنتج النهائي .

وتشير هذه النتيجة إلى أن Agile يمكن أن توفر بيئة تساعد على تحسين الجودة، لأنها تعتمد على مراجعة المنتج بصورة متكررة، بدلاً من الانتظار حتى نهاية المشروع لإجراء التقييم النهائي. ومع ذلك، لا يعني هذا أن جميع الشركات التي تطبق Agile تحقق المستوى نفسه من الجودة، لأن النتائج تعتمد على طريقة التطبيق ومدى التزام الشركة بالمبادئ والممارسات الأساسية.

### 5.3 مناقشة النتيجة الأولى

تتفق هذه النتيجة مع دراسة Dyba و Dingsøyr، التي أشارت إلى أن المنهجيات الرشيدة تساعد على تحسين التعاون والمرونة ورضا الأطراف المشاركة في المشروع.

كما تتفق مع دراسة Pinto و Serrador، التي أظهرت وجود علاقة إيجابية بين تطبيق الأساليب الرشيدة ونجاح المشروع ورضا أصحاب المصلحة.

ويمكن تفسير هذا الأثر بأن Agile تقلل الفترة الزمنية بين تنفيذ العمل ومراجعته، مما يسمح للفريق باكتشاف الأخطاء أو عدم توافق المتطلبات في وقت مبكر.

ولكن يجب الانتباه إلى أن هذه العلاقة ليست تلقائية أو مضمونة، لأن تطبيق Agile بصورة شكلية قد لا يؤدي إلى تحسين حقيقي في جودة المنتج.

#### 5.4 النتيجة الثانية: يسهم التسليم التكراري في اكتشاف الأخطاء مبكرًا

أظهر التحليل أن تقسيم المشروع إلى دورات قصيرة، وإنتاج أجزاء قابلة للاختبار في نهاية كل دورة، يساعد على اكتشاف العيوب والمشكلات بصورة مبكرة.

ففي المنهجيات التقليدية قد يتم تنفيذ معظم أعمال التطوير أولاً، ثم تبدأ مرحلة الاختبار في نهاية المشروع. وقد يؤدي ذلك إلى اكتشاف عدد كبير من الأخطاء في مرحلة متأخرة، مما يزيد من تكلفة الإصلاح ويؤثر في موعد التسليم. أما في البيئة الرشيقة، فيتم اختبار الوظائف بصورة متكررة، ويمكن معالجة الأخطاء خلال الدورة نفسها أو في الدورة التالية. وتتمثل فوائد اكتشاف الأخطاء المبكر فيما يأتي:

1. تقليل تكلفة معالجة العيوب .
2. الحد من تراكم المشكلات الفنية .
3. تحسين استقرار المنتج .
4. منع انتقال الأخطاء إلى مراحل لاحقة .
5. زيادة ثقة الفريق في الإصدارات الجديدة .
6. تقليل مخاطر إطلاق منتج يحتوي على عيوب رئيسة .

#### 5.5 مناقشة النتيجة الثانية

تتفق هذه النتيجة مع مبادئ Agile التي تؤكد أهمية التسليم المبكر والمستمر للبرمجيات العاملة. كما تتفق مع ممارسات Scrum التي تعتمد على إنتاج زيادة قابلة للاستخدام في نهاية كل دورة. ويمكن تفسير هذه النتيجة بأن الدورات القصيرة تقلل حجم العمل الذي يحتاج إلى المراجعة في كل مرة، مما يجعل اكتشاف سبب الخطأ ومعالجته أكثر سهولة.

كما أن الاختبار المبكر يحد من احتمال بناء وظائف جديدة فوق أجزاء غير مستقرة من النظام. وتدعم هذه النتيجة الفرضية الثانية التي تشير إلى أن التسليم التكراري والتغذية الراجعة المبكرة يساهمان في تحسين جودة المنتج.

#### 5.6 النتيجة الثالثة: التغذية الراجعة المستمرة تحسن الملاءمة الوظيفية

أظهرت النتائج أن مشاركة العميل أو المستخدم في مراجعة المخرجات بصورة منتظمة تساعد على تحسين الملاءمة الوظيفية للمنتج البرمجي. وقد يحدث في بعض المشاريع أن يفسر فريق التطوير المتطلبات بطريقة مختلفة عن توقعات العميل. وإذا لم يتم عرض المنتج إلا في نهاية المشروع، فقد يظهر اختلاف كبير بين المنتج المنفذ والاحتياجات الفعلية. أما في Agile، فيتم عرض الوظائف بصورة متكررة، ويقدم العميل ملاحظاته وتوجيهاته، مما يساعد على تعديل المنتج قبل استهلاك وقت وموارد إضافية.

وتسهم التغذية الراجعة في:

1. تحسين فهم المتطلبات .
2. تصحيح الأولويات .
3. اكتشاف الوظائف غير الضرورية .
4. التأكد من سهولة استخدام المنتج .
5. تحسين رضا المستخدم .

6. تقليل عمليات إعادة العمل .

7. رفع القيمة الفعلية للمنتج.

#### 5.7 مناقشة النتيجة الثالثة

تتفق هذه النتيجة مع قيمة التعاون مع العميل الواردة في بيان Agile.

كما تتفق مع دليل Scrum ، الذي يجعل مراجعة المنتج مع أصحاب المصلحة جزءاً أساسياً من دورة العمل. وتوضح هذه النتيجة أن الجودة لا تعني فقط أن النظام يعمل من الناحية الفنية، بل يجب أن يقدم الوظائف المناسبة للمستخدم. فقد يكون المنتج خالياً من الأخطاء البرمجية، لكنه لا يحقق القيمة المطلوبة إذا كانت الوظائف غير مناسبة أو معقدة أو لا تلبي احتياجات المستخدمين.

#### 5.8 النتيجة الرابعة: الممارسات الإدارية وحدها لا تكفي لتحقيق الجودة الفنية

أظهر التحليل أن الاجتماعات اليومية، وتخطيط الدورات، وترتيب قوائم المهام، ومراجعة التقدم تساعد على تحسين التنظيم والتواصل، لكنها لا تكفي وحدها لضمان جودة المنتج البرمجي.

وتحتاج الجودة الفنية إلى ممارسات هندسية تدعم عملية التطوير، مثل:

1. الاختبارات الآلية .

2. التكامل المستمر .

3. مراجعة الشفرة البرمجية .

4. إعادة هيكلة الشفرة .

5. الاختبارات الأمنية .

6. مراقبة الأداء .

7. إدارة الدين التقني .

8. توثيق القرارات الفنية المهمة .

9. تحليل أسباب العيوب المتكررة .

وإذا ركز الفريق على سرعة إكمال المهام دون تطبيق هذه الممارسات، فقد تظهر زيادة في عدد الإصدارات، ولكن مع ارتفاع عدد الأخطاء أو صعوبة صيانة النظام مستقبلاً.

#### 5.9 مناقشة النتيجة الرابعة

تتفق هذه النتيجة مع الممارسات المعروفة في البرمجة القصوى و DevOps، التي تركز على دمج الجودة داخل عملية التطوير.

كما تؤكد أن Agile ليست مجموعة من الاجتماعات فقط، بل تحتاج إلى أساس تقني قوي.

ويمكن أن تبدو بعض الفرق ناجحة من حيث عدد المهام المنجزة، بينما تعاني من تراكم الدين التقني، أو ضعف تغطية الاختبارات، أو ارتفاع العيوب بعد النشر.

وتدعم هذه النتيجة الفرضية الثالثة التي تشير إلى أن الممارسات الهندسية تساهم في تحسين موثوقية البرمجيات وقابليتها للصيانة.

#### 5.10 النتيجة الخامسة: تعريف الإنجاز يمثل أداة مهمة لضبط الجودة

أظهر التحليل أن وجود تعريف واضح ومشارك للإنجاز يساعد على منع اعتبار العمل مكتملاً قبل استيفاء متطلبات الجودة. وقد تختلف نظرة أعضاء الفريق إلى معنى اكتمال المهمة. فقد يرى المطور أن المهمة انتهت بمجرد كتابة الشفرة، بينما يرى المختبر أنها لا تزال بحاجة إلى اختبار، وقد يرى مالك المنتج أنها تحتاج إلى مراجعة وقبول. ولذلك يمكن أن يتضمن تعريف الإنجاز الشروط الآتية:

1. اكتمال البرمجة .
  2. مراجعة الشفرة البرمجية .
  3. اجتياز الاختبارات الوظيفية .
  4. اجتياز الاختبارات الآلية .
  5. التحقق من المتطلبات الأمنية .
  6. تحديث التوثيق الضروري .
  7. قبول الوظيفة من مالك المنتج .
  8. جاهزية الوظيفة للنشر أو الاستخدام .
- ويساعد تعريف الإنجاز على توحيد المعايير، وتقليل الخلافات، وتحسين مستوى الجودة في كل دورة.

#### 5.11 مناقشة النتيجة الخامسة

تتفق هذه النتيجة مع دليل Scrum ، الذي يعد تعريف الإنجاز عنصرًا أساسيًا لضمان شفافية جودة المخرجات. ويعمل تعريف الإنجاز بوصفه بوابة للجودة، لأنه يمنع انتقال العمل إلى مرحلة مكتملة إذا لم يحقق المعايير المتفق عليها. كما يساعد الإدارة وأصحاب المصلحة على فهم أن سرعة الإنجاز لا تعني تقليل متطلبات الاختبار أو التحقق.

#### 5.12 النتيجة السادسة: دعم الإدارة العليا يعزز نجاح تطبيق Agile

أظهرت النتائج أن دعم الإدارة العليا يعد من أهم العوامل المؤثرة في نجاح التحول الرشيق وتحسين جودة المخرجات. ويتمثل دعم الإدارة في:

1. توفير الموارد اللازمة .
2. دعم التدريب والتطوير .
3. إزالة العوائق التنظيمية .
4. تمكين الفرق من اتخاذ القرارات .
5. قبول التغيير التدريجي .
6. دعم الاستثمار في أدوات الاختبار والأتمتة .
7. عدم الضغط على الفرق لتسريع التسليم على حساب الجودة .
8. متابعة المؤشرات بصورة متوازنة .
9. تعزيز التعاون بين الإدارات .

وفي غياب الدعم الإداري، قد يجد الفريق صعوبة في تغيير الإجراءات أو الحصول على الأدوات أو معالجة المشكلات التي تقع خارج صلاحياته.

### 5.13 مناقشة النتيجة السادسة

تتفق هذه النتيجة مع دراسة Dikert وزملائه، التي أظهرت أن دعم الإدارة العليا ووضوح الرؤية والتدريب من أهم عوامل نجاح التحولات الرشيقية واسعة النطاق.

وتشير النتيجة إلى أن Agile ليست مسؤولية فريق التطوير فقط، بل تحتاج إلى بيئة تنظيمية تدعم المبادئ الرشيقية. وقد يؤدي التناقض بين مطالب الإدارة وممارسات الفريق إلى فشل التطبيق، مثل أن تطلب الإدارة المرونة، لكنها في الوقت نفسه تفرض موافقات طويلة أو تقيس الأداء بعدد المهام فقط.

وتدعم هذه النتيجة الفرضية الرابعة المتعلقة بأهمية دعم الإدارة وتمكين الفرق.

### 5.14 النتيجة السابعة: تمكين الفريق يحسن سرعة القرار والمسؤولية عن الجودة

أظهر التحليل أن الفرق التي تمتلك مستوى مناسباً من الاستقلالية تكون أكثر قدرة على معالجة المشكلات واتخاذ القرارات الفنية بسرعة. ويساعد تمكين الفريق على:

1. تقليل وقت انتظار الموافقات .

2. رفع المسؤولية المشتركة عن الجودة .

3. تحسين التعاون بين التخصصات .

4. زيادة الابتكار في الحلول .

5. معالجة المشكلات داخل الدورة .

6. رفع الدافعية والانتماء .

7. تحسين التعلم المستمر .

ولكن تمكين الفريق لا يعني غياب المسؤولية أو الرقابة، بل يعني منحه صلاحيات واضحة ضمن أهداف ومعايير محددة.

### 5.15 مناقشة النتيجة السابعة

يتفق ذلك مع مبادئ Agile التي تؤكد بناء المشاريع حول أفراد مؤهلين ومنحهم الثقة والدعم.

كما يتفق مع مفهوم الفرق ذاتية التنظيم في Scrum.

ويمكن تفسير هذه النتيجة بأن الأفراد الأقرب إلى المشكلة الفنية يكونون غالباً أكثر قدرة على تحديد الحل المناسب، بشرط توفر المعلومات والمهارات والصلاحيات.

### 5.16 النتيجة الثامنة: الثقافة التنظيمية تؤثر في جودة التطبيق

أظهر التحليل أن الثقافة التنظيمية قد تدعم تطبيق Agile أو تعيقه.

وتشمل الثقافة الداعمة:

1. الشفافية في عرض المشكلات .

2. التعاون بدل المنافسة الداخلية .

3. تقبل الملاحظات .

4. التعلم من الأخطاء .

5. مشاركة المعرفة .

6. الاستعداد للتغيير .

7. الثقة بين الإدارة والفرق .

8. التركيز على قيمة العميل .

أما الثقافة المعيقة فتتسم بإخفاء المشكلات، والخوف من الفشل، واللوم، والمركزية الشديدة، وضعف مشاركة المعلومات. وفي هذه البيئات قد تتحول الاجتماعات الرشيقة إلى تقارير رقابية، ويخشى الفريق من ذكر العوائق أو الأخطاء، مما يضعف التحسين المستمر.

5.17 النتيجة التاسعة: التطبيق الشكلي أو الجزئي يضعف أثر Agile

أظهر التحليل أن بعض الشركات تطبق مظاهر Agile دون مبادئها الأساسية.

ومن صور التطبيق الشكلي:

1. عقد الاجتماع اليومي دون معالجة العوائق .
  2. تقسيم العمل إلى دورات مع استمرار التغييرات العشوائية داخل الدورة .
  3. وجود مالك منتج دون صلاحية اتخاذ القرار .
  4. تنفيذ الاختبارات بعد انتهاء التطوير بدل دمجها داخل الدورة .
  5. قياس الفريق بعدد المهام فقط .
  6. عدم تنفيذ نتائج الاجتماعات الاستيعادية .
  7. اعتبار كل مهمة منتهية بمجرد انتهاء البرمجة .
  8. عدم مشاركة العميل في مراجعة المنتج .
  9. فرض القرارات على الفريق بصورة مركزية .
- وقد يؤدي هذا النوع من التطبيق إلى زيادة الاجتماعات والإجراءات دون تحقيق تحسن في الجودة.

5.18 مناقشة النتيجة التاسعة

تتفق هذه النتيجة مع دراسة Kuhrmann وزملائه، التي أوضحت أن اسم الإطار المستخدم لا يعكس بالضرورة درجة الرقابة الحقيقية. وتدعم هذه النتيجة الفرضية الخامسة التي تشير إلى أن التطبيق الشكلي أو الجزئي يؤدي إلى ضعف أثر Agile في جودة المخرجات. كما تؤكد ضرورة تقييم الممارسات الفعلية بدل الاكتفاء بإعلان الشركة أنها تستخدم Scrum أو Agile.

5.19 النتيجة العاشرة: قد يؤدي التركيز المفرط على السرعة إلى تراجع الجودة

أظهر التحليل أن ضغط التسليم المستمر قد يدفع بعض الفرق إلى تأجيل أعمال التحسين الفني أو تقليل وقت الاختبار. ومن الآثار المحتملة لذلك:

1. تراكم الدين التقني .
2. زيادة الأخطاء بعد النشر .
3. صعوبة تعديل النظام .
4. انخفاض استقرار المنتج .
5. ارتفاع تكلفة الصيانة .
6. إرهاق أعضاء الفريق .
7. انخفاض القدرة على تطوير وظائف جديدة .

## 8. تكرار الأعطال .

ويظهر هذا الخطر عندما تستخدم السرعة بوصفها المؤشر الوحيد للأداء، دون وجود مؤشرات للجودة والاستدامة.

### 5.20 مناقشة النتيجة العاشرة

لا تتعارض Agile مع الجودة الفنية، بل تؤكد أهمية التميز الفني والتصميم الجيد. ولذلك فإن استخدام Agile لتبرير تنفيذ أسرع مع إهمال الجودة يمثل فهمًا غير صحيح للمنهجية. ويجب أن توازن الشركات بين سرعة الوصول إلى السوق والمحافظة على جودة المنتج وقدرته على الاستمرار والتوسع.

### 5.21 النتيجة الحادية عشرة: النهج الهجين مناسب لبعض الشركات والمشاريع

أظهر التحليل أن عددًا من الشركات تستخدم مزيجًا من الأساليب التقليدية والرشيقة.

وقد يكون النهج الهجين مناسبًا في الحالات الآتية:

1. المشاريع الخاضعة لمتطلبات تنظيمية صارمة .
  2. المشاريع التي تتطلب موافقات رسمية .
  3. الأنظمة الحساسة أو عالية المخاطر .
  4. المشاريع المرتبطة بعقود ونطاقات محددة .
  5. المؤسسات الكبيرة متعددة الإدارات .
  6. المشاريع التي تجمع بين أعمال إنشائية أو تشغيلية وتطوير برمجي .
- ويمكن في هذه الحالات استخدام تخطيط عام وميزانية واعتمادات رسمية، مع تنفيذ أعمال التطوير في دورات قصيرة.

### 5.22 مناقشة النتيجة الحادية عشرة

تتفق هذه النتيجة مع دليل الممارسات الرشيقة الصادر عن معهد إدارة المشاريع، الذي يؤكد أهمية اختيار النهج المناسب وفق طبيعة المشروع ومستوى عدم اليقين.

كما تتفق مع دراسة Kuhrmann وزملائه التي أظهرت انتشار الممارسات الهجينة في الواقع العملي. ولا يمثل النهج الهجين ضعفًا في الرقابة إذا كان تصميمه واضحًا ويحقق التوازن بين الحوكمة والمرونة.

### 5.23 ربط ممارسات Agile بأبعاد جودة البرمجيات

يمكن توضيح العلاقة بين أهم الممارسات الرشيقة وأبعاد الجودة كما يأتي:

#### 5.23.1 التسليم التكراري

يسهم في تحسين:

- الملاءمة الوظيفية .
- رضا المستخدم .
- سرعة اكتشاف العيوب .
- تقليل مخاطر عدم توافق المنتج .

#### 5.23.2 التغذية الراجعة المستمرة

تسهم في تحسين:

- اكتمال المتطلبات .
- صحة الوظائف .
- سهولة الاستخدام .
- توافق المنتج مع توقعات العميل .

### 5.23.3 الاختبارات الآلية

تسهم في تحسين:

- الموثوقية .
- استقرار النظام .
- تقليل العيوب المتكررة .
- سرعة التحقق من التعديلات .

### 5.23.4 التكامل المستمر

يسهم في تحسين:

- جودة دمج التعديلات .
- سرعة اكتشاف مشكلات التكامل .
- استقرار البناء البرمجي .
- جاهزية الإصدارات .

### 5.23.5 مراجعة الشفرة البرمجية

تسهم في تحسين:

- القابلية للصيانة .
- جودة التصميم .
- تقليل الأخطاء .
- مشاركة المعرفة .

### 5.23.6 تعريف الإنجاز

يسهم في تحسين:

- توحيد معايير الجودة .
- منع تسليم وظائف غير مكتملة .
- وضوح حالة المنتج .
- رفع مستوى الثقة في المخرجات .

### 5.23.7 الاجتماعات الاستعدادية

تسهم في تحسين:

- جودة عملية العمل .
- معالجة أسباب المشكلات .

- التعلم المستمر .
- رفع كفاءة الفريق.

5.24 مؤشرات قياس أثر Agile على جودة البرمجيات  
أظهر التحليل أهمية استخدام مؤشرات أداء متوازنة لقياس أثر Agile.  
ومن أهم المؤشرات المقترحة:

5.24.1 عدد العيوب البرمجية

يقيس عدد الأخطاء المكتشفة خلال فترة محددة أو في كل إصدار.

5.24.2 كثافة العيوب

تقيس عدد العيوب مقارنة بحجم المنتج أو عدد الوظائف المنفذة.

5.24.3 العيوب المتسربة إلى بيئة الإنتاج

تقيس عدد أو نسبة العيوب التي لم يتم اكتشافها قبل إطلاق المنتج.

ويعد انخفاض هذا المؤشر علامة إيجابية على فعالية الاختبار والجودة.

5.24.4 زمن معالجة العيوب

يقيس الفترة من تسجيل الخطأ إلى إصلاحه وإغلاقه.

5.24.5 تغطية الاختبارات

تقيس نسبة أجزاء النظام أو الوظائف التي تشملها الاختبارات الآلية.

5.24.6 نسبة نجاح الاختبارات

تقيس نسبة الاختبارات التي اجتازها المنتج بنجاح.

5.24.7 زمن الدورة

يقيس الفترة من بدء العمل على المهمة إلى اكتمالها وجاهازيتها للاستخدام.

5.24.8 تكرار النشر

يقيس عدد مرات نشر الإصدارات خلال فترة معينة.

ولا ينبغي تفسير ارتفاع هذا المؤشر بصورة منفصلة عن معدل فشل التغيير.

5.24.9 معدل فشل التغيير

يقيس نسبة الإصدارات التي أدت إلى أعطال أو احتاجت إلى إصلاح عاجل أو تراجع عن النشر.

5.24.10 متوسط زمن استعادة الخدمة

يقيس الوقت المطلوب لإعادة النظام إلى وضعه الطبيعي بعد حدوث عطل.

5.24.11 رضا العميل أو المستخدم

يمكن قياسه من خلال الاستبيانات، والتقييمات، والشكاوى، ونسبة استخدام الوظائف.

5.24.12 الدين التقني

يشير إلى حجم الأعمال الفنية المؤجلة التي تؤثر في سهولة الصيانة والتطوير.

5.24.13 سرعة الفريق

تقيس حجم العمل المنجز خلال كل دورة، ولكن يجب استخدامها لأغراض التخطيط الداخلي، لا للمقارنة المباشرة بين الفرق أو لقياس الجودة وحدها.

## 5.25 مناقشة فرضيات الدراسة

### 5.25.1 الفرضية الأولى

تنص الفرضية الأولى على وجود أثر إيجابي لتطبيق منهجية الإدارة الرشيقة على جودة مخرجات مشاريع البرمجيات. أظهرت نتائج التحليل دعم هذه الفرضية بصورة عامة، حيث تبين أن Agile تساعد على تحسين المراجعة والتعاون واكتشاف الأخطاء والاستجابة للتغيرات.

إلا أن قوة الأثر تعتمد على جودة التطبيق والسياق التنظيمي. وبناءً عليه، تعد الفرضية مقبولة مع التأكيد على أن العلاقة مشروطة بعدد من العوامل.

### 5.25.2 الفرضية الثانية

تنص الفرضية الثانية على أن التسليم التكراري والتغذية الراجعة المبكرة يساهمان في تحسين الملاءمة الوظيفية. أظهرت النتائج أن مشاركة العميل ومراجعة الوظائف بصورة متكررة تساعدان على اكتشاف سوء الفهم وتعديل الأولويات وتحسين توافق المنتج مع احتياجات المستخدم. وبناءً عليه، تدعم النتائج قبول الفرضية الثانية.

### 5.25.3 الفرضية الثالثة

تنص الفرضية الثالثة على أن الممارسات الهندسية تساهم في تحسين الموثوقية والقابلية للصيانة. أظهر التحليل أن الاختبارات الآلية والتكامل المستمر ومراجعة الشفرة وإعادة الهيكلة ترتبط بصورة مباشرة بتحسين الجودة الفنية. وبناءً عليه، تدعم النتائج قبول الفرضية الثالثة.

### 5.25.4 الفرضية الرابعة

تنص الفرضية الرابعة على أن دعم الإدارة وتمكين الفرق يعززان أثر Agile على جودة المخرجات. أظهرت الدراسات السابقة أن دعم الإدارة والتدريب وإزالة العوائق وتمكين الأفراد تعد من أهم عوامل نجاح التحول الرشيق. وبناءً عليه، تدعم النتائج قبول الفرضية الرابعة.

### 5.25.5 الفرضية الخامسة

تنص الفرضية الخامسة على أن التطبيق الشكلي أو الجزئي يضعف أثر Agile. أظهر التحليل أن تطبيق الاجتماعات والمصطلحات دون تغيير حقيقي في الثقافة أو الممارسات الفنية لا يحقق التحسين المتوقع. وبناءً عليه، تدعم النتائج قبول الفرضية الخامسة.

## 5.26 الإجابة عن أسئلة الدراسة

السؤال الأول: ما المقصود بمنهجية Agile ؟

هي منهجية تعتمد على المرونة والتطوير التكراري والتعاون والاستجابة للتغيير والتسليم التدريجي للقيمة.

السؤال الثاني: ما المقصود بجودة مخرجات البرمجيات؟

هي قدرة المنتج على تلبية الاحتياجات وتحقيق مستوى مناسب من الملاءمة والموثوقية والأداء والأمان وسهولة الاستخدام والقابلية للصيانة.

السؤال الثالث: كيف تسهم Agile في تحسين الجودة؟

من خلال الاختبار المتكرر، والتغذية الراجعة المبكرة، والتعاون، واكتشاف الأخطاء، وتحسين الأولويات.

السؤال الرابع: ما الممارسات الأكثر تأثيراً؟

الاختبارات الآلية، والتكامل المستمر، وتعريف الإنجاز، ومراجعة الشفرة، والتسليم التكراري، ومشاركة العميل.

السؤال الخامس: ما عوامل النجاح؟

دعم الإدارة، وتمكين الفرق، والثقافة التنظيمية، والكفاءة الفنية، ومشاركة العميل، ووضوح الأدوار.

السؤال السادس: ما أبرز التحديات؟

مقاومة التغيير، والتطبيق الشكلي، ونقص المهارات، وضعف مشاركة العميل، والضغط على السرعة، وتراكم الدين التقني.

السؤال السابع: ما مؤشرات القياس المناسبة؟

العيوب المتسربة، وزمن المعالجة، وتغطية الاختبارات، ومعدل فشل التغيير، وزمن الدورة، ورضا المستخدم، والدين التقني.

السؤال الثامن: ما النموذج المناسب للتطبيق؟

نموذج يربط بين القيادة وقيمة العميل والجودة المدمجة والأتمتة والقياس والتحسين المستمر، وسيتم عرضه في الفصل السادس.

## 5.27 الاستنتاج العام لنتائج الدراسة

تشير النتائج إلى أن Agile توفر إطاراً مناسباً لتحسين جودة مخرجات مشاريع البرمجيات، ولكن نجاحها يتطلب تطبيقاً متكاملاً يجمع بين الجوانب الإدارية والفنية.

فالجانب الإداري يشمل التعاون، والتخطيط التكراري، وتمكين الفريق، ومشاركة العميل، والتحسين المستمر.

أما الجانب الفني فيشمل الاختبارات الآلية، والتكامل المستمر، ومراجعة الشفرة، والأمن، وإدارة الدين التقني.

ويؤدي الفصل بين الجانبين إلى نتائج محدودة. فقد ينجح الفريق في تحسين التنظيم دون تحسين المنتج، أو يطبق ممارسات تقنية جيدة داخل بيئة إدارية تعيق سرعة القرار والتعاون.

ولذلك ينبغي النظر إلى Agile بوصفها نظام عمل متكاملاً، وليس مجرد إطار لإدارة الاجتماعات أو المهام.

## 5.28 خلاصة الفصل

تناول هذا الفصل نتائج تحليل الدراسات السابقة والمراجع المهنية المرتبطة بأثر Agile على جودة مخرجات مشاريع البرمجيات.

وأظهرت النتائج وجود أثر إيجابي عام للمنهجية في الجودة، خاصة من خلال التسليم التكراري، والتغذية الراجعة، واكتشاف الأخطاء، والتعاون المستمر.

كما أظهرت أهمية الممارسات الهندسية في تحقيق الجودة الفنية، وأهمية دعم الإدارة وتمكين الفريق والثقافة التنظيمية.

وأوضحت النتائج أن التطبيق الشكلي أو التركيز المفرط على السرعة قد يؤديان إلى ضعف الجودة وتراكم الدين التقني.

كما تمت مناقشة فرضيات الدراسة والإجابة عن أسئلتها، وعرض مجموعة من مؤشرات الأداء المناسبة.

وسيتناول الفصل السادس النموذج التطبيقي المقترح، وخارطة التنفيذ، وتوزيع المسؤوليات، والمخاطر، والتوصيات النهائية.

## الفصل السادس: النموذج التطبيقي المقترح والتوصيات

### 6.1 تمهيد

يتناول هذا الفصل النموذج التطبيقي المقترح للاستفادة من نتائج الدراسة في تحسين جودة مخرجات مشاريع البرمجيات داخل الشركات التقنية. وقد أظهرت نتائج الدراسة أن تطبيق منهجية الإدارة الرشيدة لا يحقق أثرًا إيجابيًا بصورة تلقائية، بل يعتمد على مجموعة من العوامل الإدارية والفنية والتنظيمية.

ومن هذا المنطلق، يقدم هذا الفصل نموذجًا عمليًا يساعد الشركات على ربط ممارسات Agile بجودة البرمجيات، إلى جانب توضيح مراحل التنفيذ، والمسؤوليات، ومؤشرات الأداء، والمخاطر المحتملة، والتوصيات النهائية.

### 6.2 النموذج التطبيقي المقترح

تقترح الدراسة نموذجًا تطبيقيًا باسم:

نموذج Agile-Q لحوكمة الجودة الرشيدة

ويرمز الحرف Q إلى كلمة Quality، أي الجودة.

ويهدف النموذج إلى مساعدة الشركات التقنية على تطبيق Agile بطريقة متوازنة تربط بين سرعة الإنجاز وجودة المنتج، وتمنع تحول المنهجية إلى مجموعة من الاجتماعات أو الإجراءات الشكلية. ويتكون النموذج من ستة محاور رئيسية:

1. القيادة والسياق التنظيمي .
2. قيمة العميل .
3. الجودة المدمجة .
4. التدفق والأتمتة .
5. القياس والشفافية .
6. التعلم والتحسين المستمر .

### 6.3 المحور الأول: القيادة والسياق التنظيمي

يبدأ نجاح تطبيق Agile من فهم الإدارة العليا لأهداف التحول، وتحديد المشكلة التي تسعى الشركة إلى حلها. فلا ينبغي تطبيق المنهجية لمجرد أنها شائعة أو مستخدمة في شركات أخرى، بل يجب أن ترتبط بحاجة واضحة، مثل:

1. تقليل تأخر المشاريع .
2. تحسين جودة المنتجات .
3. رفع رضا العملاء .
4. تقليل العيوب بعد النشر .
5. تحسين التعاون بين الإدارات .
6. تسريع الاستجابة للتغيرات .
7. تقليل تكلفة إعادة العمل .

ويشمل هذا المحور مجموعة من المتطلبات، منها:

- تحديد رؤية واضحة للتحول .

- دعم الإدارة العليا .
  - توفير الموارد والتدريب .
  - إزالة العوائق التنظيمية .
  - منح الفرق صلاحيات مناسبة .
  - اختيار النهج المناسب لطبيعة المشروع .
  - عدم فرض نموذج واحد على جميع الفرق .
  - ربط التحول بأهداف ومؤشرات واضحة .
- ويعد هذا المحور أساساً لبقية محاور النموذج، لأن غياب دعم الإدارة قد يجعل الفرق غير قادرة على تغيير الإجراءات أو الحصول على الأدوات أو اتخاذ القرارات اللازمة.

#### 6.4 المحور الثاني: قيمة العميل

- يركز المحور الثاني على ضمان أن أعمال التطوير ترتبط بقيمة حقيقية للعميل أو المستخدم. ويتطلب ذلك مشاركة العميل أو ممثله بصورة مستمرة في تحديد الأولويات ومراجعة المخرجات. ويتضمن هذا المحور ما يأتي:
1. تحديد احتياجات المستخدمين بوضوح .
  2. ترتيب قائمة المنتج حسب القيمة والمخاطر .
  3. تطوير الوظائف الأكثر أهمية أولاً .
  4. عرض المخرجات بصورة متكررة .
  5. جمع التغذية الراجعة .
  6. تعديل الأولويات بناءً على النتائج .
  7. قياس رضا المستخدم .
  8. تقليل الوظائف غير الضرورية .
  9. إشراك أصحاب المصلحة في القرارات المهمة .
- ويساعد هذا المحور على تحسين الملاءمة الوظيفية للمنتج، وتقليل احتمال بناء وظائف لا يحتاجها المستخدم أو لا تحقق القيمة المطلوبة.

#### 6.5 المحور الثالث: الجودة المدمجة

- يقصد بالجودة المدمجة أن تكون أنشطة الجودة جزءاً من كل دورة تطوير، بدلاً من تأجيلها إلى نهاية المشروع. ويشمل هذا المحور ما يأتي:
1. وضع تعريف واضح للإنجاز .
  2. إدراج الاختبار ضمن كل دورة .
  3. مراجعة الشفرة البرمجية .
  4. تنفيذ الاختبارات الوظيفية .
  5. تطبيق الاختبارات الآلية .
  6. تنفيذ الاختبارات الأمنية عند الحاجة .

7. مراجعة الأداء .
  8. تحديث التوثيق الضروري .
  9. معالجة العيوب قبل اعتبار المهمة مكتملة .
  10. إدارة الدين التقني بصورة مستمرة .
- ويمكن أن يتضمن تعريف الإنجاز الشروط الآتية:
- اكتمال البرمجة .
  - اجتياز الاختبارات .
  - مراجعة الشفرة .
  - قبول مالك المنتج .
  - عدم وجود عيوب رئيسة .
  - تحديث التوثيق .
  - جاهزية الوظيفة للنشر أو الاستخدام .
- ويضمن هذا المحور عدم التضحية بالجودة من أجل زيادة عدد المهام المنجزة أو تسريع التسليم.

#### 6.6 المحور الرابع: التدفق والأتمتة

يركز هذا المحور على تحسين تدفق العمل وتقليل التوقف والاختناقات، إلى جانب استخدام الأتمتة في عمليات البناء والاختبار والنشر. ويشمل هذا المحور ما يأتي:

1. تصور مراحل العمل بوضوح .
  2. تحديد حجم العمل قيد التنفيذ .
  3. تقليل الانتقال المتكرر بين المهام .
  4. اكتشاف نقاط الاختناق .
  5. تطبيق التكامل المستمر .
  6. أتمتة الاختبارات المتكررة .
  7. أتمتة عمليات البناء .
  8. تحسين عمليات النشر .
  9. مراقبة الأنظمة بعد الإطلاق .
  10. تقليل التدخل اليدوي في العمليات المتكررة .
- وتسهم الأتمتة في تقليل الأخطاء البشرية، وتسريع اكتشاف المشكلات، وتحسين استقرار الإصدارات. ولا يشترط تنفيذ الأتمتة بصورة كاملة منذ البداية، بل يمكن تطبيقها تدريجياً حسب أولويات المنتج والموارد المتاحة.

#### 6.7 المحور الخامس: القياس والشفافية

يتطلب نجاح Agile وجود معلومات واضحة عن تقدم العمل وجودة المنتج. ولا يكفي قياس عدد المهام المنجزة، لأن هذا المؤشر لا يوضح بالضرورة قيمة المخرجات أو جودتها. ويشمل هذا المحور استخدام مجموعة من المؤشرات المتوازنة، مثل:

1. عدد العيوب البرمجية .
  2. العيوب المتسربة إلى بيئة الإنتاج .
  3. زمن معالجة الأخطاء .
  4. تغطية الاختبارات .
  5. نسبة نجاح الاختبارات .
  6. زمن الدورة .
  7. تكرار النشر .
  8. معدل فشل التغيير .
  9. متوسط زمن استعادة الخدمة .
  10. رضا العميل .
  11. الدين التقني .
  12. استقرار النظام .
  13. نسبة تحقيق أهداف الدورة .
  14. الأعمال المتأخرة أو المعادة .
- ويجب عرض المؤشرات بصورة واضحة للفريق والإدارة، مع تجنب استخدامها لمعاقبة الأفراد أو مقارنة الفرق بصورة غير عادلة. والهدف من القياس هو التعلم واتخاذ القرار، لا زيادة الضغط أو إخفاء المشكلات.

#### 6.8 المحور السادس: التعلم والتحسين المستمر

يركز هذا المحور على مراجعة طريقة العمل وتحديد فرص التحسين بصورة مستمرة. ويشمل ذلك:

1. عقد الاجتماعات الاستيعادية .
  2. تحديد الأسباب الجذرية للمشكلات .
  3. اختيار إجراءات تحسين واضحة .
  4. تحديد المسؤول عن كل إجراء .
  5. وضع مدة زمنية للتنفيذ .
  6. متابعة نتائج التحسين .
  7. مشاركة المعرفة بين الفرق .
  8. توثيق الدروس المستفادة .
  9. مراجعة المؤشرات بصورة دورية .
  10. تعديل الممارسات حسب النتائج .
- ويجب ألا تتحول الاجتماعات الاستيعادية إلى مناقشات متكررة دون تنفيذ. بل ينبغي أن ينتج عن كل اجتماع إجراء أو أكثر قابل للمتابعة والقياس.

## 6.9 مراحل تنفيذ النموذج المقترح

يمكن تنفيذ نموذج Agile-Q من خلال خمس مراحل رئيسية:

### 6.9.1 المرحلة الأولى: التشخيص

تهدف هذه المرحلة إلى فهم الوضع الحالي للشركة أو الفريق. وتشمل:

1. تقييم طريقة إدارة المشاريع الحالية .
  2. قياس مستوى تطبيق Agile.
  3. تحديد مؤشرات الجودة الحالية .
  4. تحليل العيوب المتكررة .
  5. تقييم الثقافة التنظيمية .
  6. تحديد مهارات الفريق .
  7. تحديد الأدوات المستخدمة .
  8. تحديد نقاط القوة والضعف .
  9. معرفة مستوى مشاركة العميل .
  10. تحديد العوائق التنظيمية .
- ويكون المخرج الرئيس لهذه المرحلة تقريرًا يوضح خط الأساس والفجوات الحالية.

### 6.9.2 المرحلة الثانية: التصميم

تهدف هذه المرحلة إلى إعداد خطة التطبيق. وتشمل:

1. اختيار فريق أو مشروع تجريبي .
  2. تحديد أهداف التحسين .
  3. اختيار الممارسات المناسبة .
  4. تحديد الأدوار والمسؤوليات .
  5. إعداد تعريف الإنجاز .
  6. اختيار مؤشرات الأداء .
  7. تحديد خطة التدريب .
  8. تحديد الأدوات المطلوبة .
  9. وضع خطة إدارة المخاطر .
  10. تحديد مدة التجربة .
- ويجب أن تكون الخطة واقعية ومتناسبة مع حجم الشركة وخبرة الفريق.

### 6.9.3 المرحلة الثالثة: التجربة

يتم في هذه المرحلة تطبيق النموذج على نطاق محدود.

ويمكن أن تستمر التجربة من ثمانية إلى اثني عشر أسبوعاً، حسب طبيعة المشروع. وتشمل:

1. تنفيذ دورات قصيرة .
  2. إشراك العميل .
  3. تفعيل تعريف الإنجاز .
  4. تطبيق الاختبارات تدريجياً .
  5. استخدام التكامل المستمر عند الإمكان .
  6. متابعة المؤشرات .
  7. عقد مراجعات دورية .
  8. تنفيذ الاجتماعات الاستيعادية .
  9. تسجيل العوائق .
  10. مقارنة النتائج بخط الأساس .
- وتساعد المرحلة التجريبية على اكتشاف المشكلات قبل تعميم التطبيق.

#### 6.9.4 المرحلة الرابعة: التقييم والتوسع

بعد انتهاء التجربة، يتم تقييم النتائج من خلال مقارنة الوضع قبل التطبيق وبعده. ويشمل التقييم:

1. مقارنة عدد العيوب .
2. مقارنة زمن الدورة .
3. قياس رضا العميل .
4. قياس استقرار الإصدارات .
5. تقييم مشاركة الفريق .
6. تحديد الممارسات الناجحة .
7. تحديد الممارسات التي تحتاج إلى تعديل .
8. مراجعة تكلفة التطبيق .
9. توثيق الدروس المستفادة .
10. إعداد خطة التوسع .

وفي حال نجاح التجربة، يمكن تطبيق النموذج على فرق أو منتجات إضافية بصورة تدريجية.

#### 6.9.5 المرحلة الخامسة: التحسين المستمر

لا ينتهي تطبيق النموذج بعد التوسع، بل يجب أن تستمر الشركة في مراجعة النتائج وتطوير الممارسات. وتشمل هذه المرحلة:

1. مراجعة المؤشرات شهرياً أو ربع سنوياً .
2. تحديث تعريف الإنجاز .

3. تحسين أدوات الأتمتة .
4. تطوير مهارات الفريق .
5. معالجة الدين التقني .
6. تحسين تجربة العميل .
7. مشاركة أفضل الممارسات .
8. مراجعة المخاطر .
9. تحديث الأدوار والصلاحيات عند الحاجة .
10. المحافظة على ثقافة التعلم .

#### 6.10 الأدوار والمسؤوليات

يتطلب تنفيذ النموذج توزيعاً واضحاً للمسؤوليات.

##### 6.10.1 الإدارة العليا

تتولى الإدارة العليا ما يأتي:

- تحديد أهداف التحول .
- توفير الموارد .
- دعم التدريب .
- إزالة العوائق .
- متابعة المؤشرات .
- دعم التعاون بين الإدارات .
- حماية الجودة من ضغوط التسليم غير الواقعية .
- تمكين الفرق من اتخاذ القرارات .

##### 6.10.2 مالك المنتج

يتولى مالك المنتج:

- ترتيب الأولويات .
- توضيح متطلبات العميل .
- مراجعة المخرجات .
- قبول الوظائف المنجزة .
- التواصل مع أصحاب المصلحة .
- تحديد قيمة كل وظيفة .
- اتخاذ القرارات المتعلقة بقائمة المنتج .

##### 6.10.3 قائد الفريق أو Scrum Master

يتولى:

- تيسير الاجتماعات .
- دعم تطبيق الممارسات .

- إزالة العوائق أو تصعيدها .
- حماية الفريق من التدخلات غير الضرورية .
- دعم التحسين المستمر .
- تعزيز الشفافية والتعاون .

#### 6.10.4 فريق التطوير

يتولى:

- تحليل المتطلبات .
- تصميم الحلول .
- تطوير البرمجيات .
- اختبار الوظائف .
- مراجعة الشفرة .
- معالجة العيوب .
- تحسين التصميم .
- إدارة الدين التقني .
- المشاركة في التخطيط والتحسين .

#### 6.10.5 فريق الجودة

يتولى:

- إعداد استراتيجية الاختبار .
- تنفيذ الاختبارات .
- تطوير الاختبارات الآلية .
- تحليل اتجاهات العيوب .
- التحقق من معايير الإنجاز .
- دعم تحسين العمليات .
- المشاركة منذ بداية التطوير .

#### 6.10.6 فريق التشغيل وDevOps

يتولى:

- تجهيز بيئات العمل .
- دعم التكامل المستمر .
- أتمتة البناء والنشر .
- مراقبة الأنظمة .
- تحليل الأعطال .
- تحسين استقرار الإصدارات .
- تقليل زمن استعادة الخدمة .

#### 6.10.7 فريق الأمن السيبراني

يتولى:

- تحديد المتطلبات الأمنية .
- إجراء المراجعات الأمنية .
- اختبار نقاط الضعف .
- دعم ممارسات الأمن المبكر .
- متابعة المخاطر الأمنية .
- التأكد من حماية البيانات والصلاحيات.

#### 6.11 مؤشرات الأداء المقترحة

يمكن استخدام مؤشرات الأداء الآتية لتقييم نجاح النموذج:

##### مؤشرات الجودة

1. انخفاض عدد العيوب بعد النشر .
2. انخفاض نسبة العيوب الحرجة .
3. ارتفاع تغطية الاختبارات .
4. تحسين استقرار النظام .
5. انخفاض معدل فشل التغيير .
6. انخفاض زمن معالجة الأخطاء .
7. تحسين القابلية للصيانة .

##### مؤشرات التدفق

1. انخفاض زمن الدورة .
2. تقليل الأعمال المتأخرة .
3. تقليل العمل قيد التنفيذ .
4. رفع نسبة إنجاز أهداف الدورة .
5. تقليل زمن انتظار الموافقات .

##### مؤشرات العميل

1. ارتفاع رضا العميل .
2. ارتفاع معدل قبول الوظائف .
3. تقليل طلبات إعادة العمل .
4. زيادة استخدام الوظائف الجديدة .
5. انخفاض الشكاوى .

##### مؤشرات الفريق

1. تحسين مستوى التعاون .
2. ارتفاع وضوح الأدوار .
3. زيادة مشاركة المعرفة .

4. انخفاض الإرهاق .
5. ارتفاع تنفيذ إجراءات التحسين .
6. تحسن رضا أعضاء الفريق.

#### 6.12 المخاطر المحتملة وطرق معالجتها

##### 6.12.1 مقاومة التغيير

قد يرفض بعض الموظفين الأسلوب الجديد بسبب الخوف من زيادة المسؤولية أو فقدان السيطرة.  
طريقة المعالجة:

- توضيح أهداف التحول .
- إشراك الموظفين .
- تقديم تدريب عملي .
- تطبيق التغيير تدريجيًا .
- دعم الإدارة المباشر .

##### 6.12.2 التركيز على السرعة فقط

قد تركز الإدارة على زيادة عدد الإصدارات دون الاهتمام بالجودة.  
طريقة المعالجة:

- اعتماد مؤشرات جودة واضحة .
- تطبيق تعريف الإنجاز .
- مراقبة العيوب والدين التقني .
- ربط الأداء بالقيمة والجودة .

##### 6.12.3 نقص المهارات الفنية

قد لا يمتلك الفريق الخبرة اللازمة في الاختبارات الآلية أو التكامل المستمر.  
طريقة المعالجة:

- تنفيذ برامج تدريبية .
- الاستعانة بخبراء .
- البدء بأدوات بسيطة .
- تطبيق الأتمتة تدريجيًا .
- مشاركة المعرفة بين الفرق .

##### 6.12.4 ضعف مشاركة العميل

قد لا يكون العميل متاحًا للمراجعة أو اتخاذ القرار.  
طريقة المعالجة:

- تعيين ممثل مخول .
- تحديد مواعيد مراجعة ثابتة .
- توضيح أثر التأخير .

- استخدام قنوات تواصل واضحة .

#### 6.12.5 تراكم الدين التقني

قد تؤدي ضغوط التسليم إلى تأجيل التحسينات الفنية.  
طريقة المعالجة:

- قياس الدين التقني .
- تخصيص نسبة من وقت كل دورة لمعالجته .
- ترتيب الأولويات حسب المخاطر .
- منع إضافة وظائف جديدة فوق مكونات غير مستقرة .

#### 6.12.6 تضخم الاجتماعات

قد يزيد عدد الاجتماعات دون تحقيق قيمة.  
طريقة المعالجة:

- تحديد هدف كل اجتماع .
- الالتزام بمدة محددة .
- إلغاء الاجتماعات غير المفيدة .
- تسجيل القرارات والإجراءات .
- مراجعة فعالية الاجتماعات .

#### 6.12.7 غموض الأدوار

قد يحدث تداخل بين مالك المنتج والمدير وقائد الفريق.  
طريقة المعالجة:

- توثيق الصلاحيات .
- توضيح المسؤوليات .
- تدريب أصحاب الأدوار .
- معالجة التعارضات مبكرًا .

#### 6.12.8 استخدام المؤشرات بصورة خاطئة

قد تستخدم المؤشرات لمقارنة الأفراد أو الضغط على الفرق.  
طريقة المعالجة:

- استخدام المؤشرات للتحسين .
- عدم مقارنة سرعة فرق مختلفة .
- تفسير الأرقام ضمن سياقها .
- الجمع بين المؤشرات الكمية والنوعية .

#### 6.13 التوصيات النهائية للدراسة

في ضوء نتائج الدراسة، توصي الدراسة بما يأتي:

1. ضرورة عدم تطبيق Agile بوصفها مجموعة من الاجتماعات فقط، بل كنظام عمل متكامل .

2. ربط التحول الرشيق بمشكلة أعمال وأهداف واضحة .
3. تطبيق المنهجية تدريجيًا على فريق أو مشروع تجريبي قبل التوسع .
4. إدراج الجودة والاختبار داخل كل دورة تطوير .
5. إعداد تعريف واضح ومشارك للإنتاج .
6. دعم الاختبارات الآلية والتكامل المستمر .
7. تعزيز مشاركة العميل في مراجعة المخرجات وترتيب الأولويات .
8. تمكين فرق العمل ومنحها صلاحيات مناسبة .
9. توفير دعم واضح ومستمر من الإدارة العليا .
10. تطوير مهارات العاملين في Agile والجودة و DevOps .
11. استخدام مؤشرات متوازنة تجمع بين السرعة والجودة والقيمة .
12. مراقبة العيوب المتسربة والدين التقني بصورة مستمرة .
13. عدم استخدام سرعة الفريق بوصفها مؤشرًا وحيدًا للنجاح .
14. تعزيز ثقافة الشفافية والتعلم من الأخطاء .
15. تنفيذ الاجتماعات الاستيعادية وربطها بإجراءات تحسين فعلية .
16. اختيار النهج الرشيق أو الهجين وفق طبيعة المشروع .
17. الاستفادة من معيار ISO/IEC 25010 في تحديد خصائص الجودة المهمة .
18. دمج الأمن السيبراني في مراحل التطوير المبكرة .
19. توثيق الدروس المستفادة ومشاركتها بين الفرق .
20. إجراء دراسات ميدانية مستقبلية داخل الشركات التقنية العربية.

#### 6.14 مقترحات لدراسات مستقبلية

نقترح الدراسة إجراء البحوث الآتية مستقبلاً:

1. دراسة ميدانية تقيس أثر Agile على جودة البرمجيات في الشركات السعودية .
2. مقارنة أثر Agile بين الشركات الناشئة والشركات الكبيرة .
3. دراسة أثر الاختبارات الآلية في تقليل العيوب بعد النشر .
4. دراسة العلاقة بين الدين التقني وسرعة الإصدارات .
5. تحليل أثر الثقافة التنظيمية على نجاح التحول الرشيق .
6. دراسة أثر القيادة الرشيقة على أداء فرق البرمجيات .
7. مقارنة بين Scrum و Kanban في جودة المخرجات .
8. دراسة أثر DevOps في استقرار الأنظمة وسرعة النشر .
9. تحليل استخدام الذكاء الاصطناعي في دعم الاختبارات الرشيقة .
10. تطوير مقياس عربي لنضج الجودة الرشيقة.

## 6.15 خلاصة الفصل

قدم هذا الفصل نموذج Agile-Q المقترح لحوكمة الجودة الرشيقة، والذي يهدف إلى مساعدة الشركات التقنية على تحقيق التوازن بين سرعة التطوير وجودة المنتجات البرمجية. ويتكون النموذج من ستة محاور تشمل القيادة والسياق التنظيمي، وقيمة العميل، والجودة المدمجة، والتدفق والأتمتة، والقياس والشفافية، والتعلم والتحسين المستمر. كما تم عرض مراحل تنفيذ النموذج، وتوزيع الأدوار والمسؤوليات، ومؤشرات الأداء، والمخاطر المحتملة وطرق معالجتها. وقدم الفصل مجموعة من التوصيات العملية التي يمكن للشركات التقنية الاستفادة منها عند تطبيق Agile.

## الخاتمة

تناولت هذه الدراسة أثر تطبيق منهجية الإدارة الرشيقة Agile على جودة مخرجات مشاريع البرمجيات في الشركات التقنية، وذلك من خلال تحليل المفاهيم الأساسية للمنهجية، واستعراض أهم الأطر والممارسات المرتبطة بها، ودراسة العلاقة بينها وبين أبعاد جودة البرمجيات.

وقد أوضحت الدراسة أن الإدارة الرشيقة لا تمثل مجرد أسلوب لتنظيم الاجتماعات أو تقسيم العمل إلى دورات قصيرة، بل هي منهج متكامل يقوم على التعاون، والمرونة، والتسليم التكراري، والتغذية الراجعة المستمرة، وتمكين فرق العمل، والتحسين المستمر. كما بينت الدراسة أن جودة مخرجات البرمجيات لا تقتصر على خلو المنتج من الأخطاء، وإنما تشمل مجموعة من الخصائص، مثل الملاءمة الوظيفية، والموثوقية، وكفاءة الأداء، وسهولة الاستخدام، والأمان، والقابلية للصيانة، والتوافق، والمرونة. وأظهرت نتائج التحليل أن تطبيق Agile بصورة صحيحة يمكن أن يساهم في اكتشاف الأخطاء في مراحل مبكرة، وتحسين فهم متطلبات العميل، وتقليل عمليات إعادة العمل، ورفع مستوى التعاون بين أعضاء الفريق، وزيادة قدرة المشروع على الاستجابة للتغيرات.

كما تبين أن الممارسات الإدارية وحدها لا تكفي لتحقيق جودة فنية مرتفعة، بل يجب دعمها بممارسات هندسية، مثل الاختبارات الآلية، والتكامل المستمر، ومراجعة الشفرة البرمجية، وإعادة الهيكلة، وإدارة الدين التقني، وتطبيق تعريف واضح للإنجاز. وأوضحت الدراسة كذلك أن نجاح Agile يعتمد بدرجة كبيرة على دعم الإدارة العليا، وتمكين فرق العمل، ووضوح الأدوار، ومشاركة العميل، ووجود ثقافة تنظيمية تشجع على الشفافية والتعاون والتعلم من الأخطاء. وفي المقابل، أظهرت النتائج أن التطبيق الشكلي أو الجزئي للمنهجية قد يؤدي إلى زيادة سرعة التسليم دون تحقيق تحسن فعلي في جودة المنتج، خاصة عند التركيز على عدد المهام المنجزة، أو إهمال الاختبارات، أو تأجيل معالجة المشكلات الفنية. وقدمت الدراسة نموذج Agile-Q لحوكمة الجودة الرشيقة، الذي يربط بين القيادة، وقيمة العميل، والجودة المدمجة، والتدفق والأتمتة، والقياس والشفافية، والتعلم والتحسين المستمر.

ويهدف هذا النموذج إلى مساعدة الشركات التقنية على تطبيق الإدارة الرشيقة بصورة متوازنة تحقق السرعة والجودة في الوقت نفسه، مع الاعتماد على مؤشرات أداء واضحة ومراحل تنفيذ تدريجية. وبناءً على ما سبق، يمكن القول إن Agile قادرة على تحسين جودة مخرجات مشاريع البرمجيات، إلا أن هذا التحسن لا يتحقق بمجرد تبني اسم المنهجية، وإنما يتطلب تطبيقاً فعلياً ومتكاملاً يجمع بين الجوانب الإدارية والفنية والتنظيمية. وتؤكد الدراسة أهمية مواصلة البحث في هذا المجال، خاصة من خلال إجراء دراسات ميدانية داخل الشركات التقنية العربية، بهدف قياس أثر الممارسات الرشيقة بصورة كمية، والتحقق من فاعلية النموذج المقترح في بيئات العمل المختلفة.

## المراجع

Agile Alliance. (2001). *Manifesto for Agile Software Development*.

Agile Alliance. (2001). *Principles behind the Agile Manifesto*.

Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119, 87–108. <https://doi.org/10.1016/j.jss.2016.06.013>

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

International Organization for Standardization. (2023). *ISO/IEC 25010:2023: Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Product quality model*. ISO.

Kuhrmann, M., Tell, P., Hebig, R., Klünder, J., Münch, J., Linssen, O., Pfahl, D., Felderer, M., Prause, C. R., MacDonell, S. G., Nakatumba-Nabende, J., Raffo, D., Beecham, S., Tüzün, E., López, G., Paez, N., Fontdevila, D., Licorish, S. A., Küpper, S., & Keil, P. (2021). What makes agile software development agile? *IEEE Transactions on Software Engineering*.

Project Management Institute. (2021). *A guide to the project management body of knowledge (PMBOK Guide)* (7th ed.). Project Management Institute.

Project Management Institute, & Agile Alliance. (2017). *Agile practice guide*. Project Management Institute.

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum*.

Serrador, P., & Pinto, J. K. (2015). Does Agile work? A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051.

<https://doi.org/10.1016/j.ijproman.2015.01.006>

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

Stavru, S. (2014). A critical examination of recent industrial surveys on agile method usage. *Journal of Systems and Software*, 94, 87–97. <https://doi.org/10.1016/j.jss.2014.03.041>