International Journal of Scientific Research (IJSR)

المجلة الدولية للبحوث العلمية

Vol. (1), No. (1) November, 2022 الإصدار (1)، العدد (1)

"نهج هجين مقترح لتطوير نظم معلومات الرعاية الصحية"

"A Proposed Hybrid Approach to Developing Health Care Information Systems"

ولاء حسين المصري

ماجستير تكنولوجيا المعلومات، حوسبة الرعاية الصحية، جامعة القاهرة، مصر

welmasry1611@gmail.com

نسمة السكري

ماجستير نظم المعلومات، حوسبة الرعاية الصحية، جامعة القاهرة، مصر
أخصائي بقسم هندسة الإشعاع، المركز الوطني لأبحاث وتكنولوجيا الإشعاع (NCRRT)
هيئة الطاقة الذرية المصرية
nesma.elsokary@gmail.com

ملخص البحث:

برزت أهمية تطوير برمجيات الرعاية الصحية باعتبارها عملية ضرورية لتحويل النظام الورقي للرعاية الصحية إلى نظام معلومات يعتمد على الحاسب الآلي للتأكد من فاعلية وكفاءة وجودة تقديم الخدمات الصحية، وتقليل الأخطاء الطبية والتأكد من سلامة المرضى. كما برزت أهمية تطوير برمجيات الرعاية الصحية نتيجة درجة التعقيد الكبيرة التي تتسم بها خدمات الرعاية الصحية، والدقة المطلوبة لضمان عدم وجود أخطاء. تحتاج برمجيات الرعاية الصحية إلى الامتثال للقوانين واللوائح أثناء تطويرها، وأن تتسم بأن تكون أكثر أماناً واستقراراً عن غيرها من البرمجيات، وأن تكون ذات أداء عالي بدون أعطال. في الواقع، غالباً ما لا تكون هذه هي الحالة الشائعة، حيث ستقوم بإصدار منتج ذي جودة رديئة وتقوم بتحسينه باستمرار - فأنت ترغب في أن يتم الوثوق بمنتجك واستخدامه منذ اليوم الأول. تطوير البرمجيات الرشيقة Agile هي عملية تكرارية تخلق منتج برمجي قابل للاستخدام من المراحل الأولى، مما يتيح التطوير السريع والعالي الجودة والذي سيستجيب بسرعة للمتطلبات المتغيرة. قد يكون نموذج الشلال Waterfall نموذجاً جيداً لتطوير البرامج

في حالة مشروعات البرمجيات ذات الحجم الكبير. بالرغم من ذلك، هناك العديد من المفاضلات بين نهجي تطوير البرمجيات، Agile أو Waterfall. تناولت هذه الورقة منهجيات تطوير البرمجيات الشائعة ومناقشات حول فوائد وعيوب كل منهم، كيف نقوم بتعديلها وفقاً لمتطلبات نظم معلومات الرعاية الصحية. خلال هذا البحث تم اقتراح منهجية تطوير البرمجيات الهجينة (Hybrid Scrum-Fall Model) لتطوير نظام الرعاية الصحية. يجمع النموذج المقترح بين منهجية التطوير الرشيقة Scrum ونموذج الشلال Waterfall لتقديم حلول تتميز بالجودة وتقدم قيمة أكبر للمستخدم النهائي.

الكلمات المفتاحية:

نظم معلومات الرعاية الصحية، تطوير البرمجيات، نموذج الشلال Waterfall، التطوير الرشيق للبرمجيات، Scrum

Abstract:

The importance of developing health care software emerged as a necessary process to transform the paper health care system into a computer-based information system to ensure the effectiveness, efficiency, and quality of health service provision, reduce medical errors and ensure patient safety. The importance of developing health care software also emerged due to the high degree of complexity of health care services, and the accuracy required to ensure that there are no errors. Healthcare software needs to comply with laws and regulations during its development, be more secure and stable than other software, and be high-performance without crashes. In reality, this is often not the common case, as you are going to release a poor-quality product and constantly improve it - you want your product to be trusted and used from day one. Agile software development is an iterative process that creates a usable software product from the early stages, enabling rapid, high-quality development that will respond quickly to changing

requirements. The Waterfall model may be a good software development model in the case of large-scale software projects. However, there are many tradeoffs between the two software development approaches, Agile or Waterfall. This paper covers common software development methodologies and discusses the benefits and drawbacks of each of them, and how we can modify them according to the requirements of healthcare information systems. During this research, a hybrid software development methodology (Hybrid Scrum-Fall Model) was proposed to develop the health care system. The proposed model combines the agile Scrum development methodology with the Waterfall model to deliver quality software solutions that deliver greater value to the end user.

Keywords:

Healthcare Information Systems, Software Development, Waterfall Model, Agile Software Development, Scrum.

1- مقدمة

في السنوات الأخيرة، اتجهت التقنيات الجديدة إلى تطوير وتحسين الرعاية الصحية والخدمات، لتحقيق أهداف مقدمي خدمات الرعاية الصحية، والمشاركة بخلق حلول برمجية لتواجه التحديات والمشاكل في مجال الرعاية الصحية. تتعامل هذه الصناعة مع أخطر الأمور ألا وهي صحة الإنسان، لذلك في معظم الحالات، لا يوجد مجال للفشل والتعلم من خلال الأخطاء. تحتاج حلول الرعاية الصحية إلى العمل بالطريقة التي تحقق توقعات المستخدمين وأداء جميع الوظائف والمهام من البداية. ويجب أن تخضع هذه الحلول الى الامتثال للقوانين واللوائح، وأن تكون أكثر أماناً واستقراراً وأداءً عالياً، كما يجب أن تكون متاحة باستمرار. هذا ليس مجالاً يمكنك من خلاله إصدار منتج ذي جودة رديئة وتحسينه باستمرار - فأنت بحاجة إلى أن يكون منتجك موثوقًا به ويمكنك استخدامه منذ اليوم الأول.

تتميز طرق تطوير البرامج التقليدية بسلسلة من الخطوات المتسلسلة التي يتم فيها تحديد المتطلبات وتوثيقها وتعتمد هذه الطرق التقليدية أيضاً على طرق التنبؤ، حيث تعتمد جميع الأنشطة بشكل صارم على تحليل وتخطيط جميع المتطلبات في البداية. في النهج التقليدي، إذا وجد تغيير في متطلبات البرنامج فيكون ذلك من خلال عملية إدارة تغيير ذات إجراءات محددة مسبقاً وقد تكون معقدة في بعض الأحيان.

أما في الطرق الرشيقة لتطوير البرمجيات Agile فإن السمة الرئيسية لها هو أنها تعتمد على نهج تكيفي ولا يتم التخطيط بالتفصيل لجميع المتطلبات من البداية، وتعتمد أيضاً على المشاركة المبكرة للعميل [1]. تتضمن الطرق الرشيقة لتطوير البرمجيات Agile على الاختبار التكراري، والتواصل المفتوح والمستمر مع العملاء، والتعاون الجماعي بين فريق العمل، ولكنها تتضمن الحد الأدنى من عملية التوثيق، وتقليل مخاطر الأخطاء الجسيمة في أنشطة التطوير [2].

تقدم هذه الورقة مراجعة للأدبيات الخاصة بتطوير نظام الرعاية الصحية في القسم 2، وتلخص بعض الأعمال ذات الصلة في القسم 3، ثم تستعرض بإيجاز منهجيات تطوير البرامج في القسم 4، وأخيراً تقدم نهج هجين مقترح لتطوير نظم معلومات الرعاية الصحية Hybrid Scrum-Fall model في القسم 5.

2- تطوير نظام الرعاية الصحية

غالباً ما توصف أنظمة الرعاية الصحية بأنها أنظمة معقدة بشكل خاص إذا كانت تقع في بيئات ديناميكية، وتتفاعل مع مجموعة متنوعة من الجهات الفاعلة والاهتمامات والعوامل المختلفة، وتتميز باحتياجات مختلفة ومتغيرة.

وتتأثر نظم الرعاية الصحية بالعديد من العوامل، منها: القوى السياسية المحلية والوطنية والإقليمية، وكذلك أصحاب المصلحة مثل الجهات الأكاديمية والشركات والمنظمات غير الحكومية والمرضى. التغيرات الاجتماعية والديموغرافية، والتقدم السريع في التكنولوجيا الطبية، والاهتمامات المهنية، وتغير المطالب العامة للرعاية الصحية والنظريات الجديدة حول إدارة وتنظيم الرعاية الصحية، كلها تؤثر على أنظمة الرعاية الصحية [3]. يواجه تطوير برامج الرعاية الصحية جميع تحديات عمليات التطوير التقليدية للبرمجيات، بالإضافة إلى عدد من العوامل الحاسمة وغير القابلة للتفاوض. تزيد هذه العوامل من تعقيد تطوير البرمجيات المستخدمة في المجال الطبي.

عند تحديد متطلبات برمجيات الرعاية الصحية، تأخذ عملية تطوير البرمجيات في الاعتبار الأجهزة الطبية المتضمنة في خدمات الرعاية الصحية، ويحتاج فريق التطوير إلى تحديد هذه المتطلبات مسبقاً وسيعتمدون على الإجابة عن الأسئلة التالية: هل يعتبر البرنامج جهازاً طبياً؟ أم أنه جزء من جهاز طبي؟ وهل يتم تخزين معلومات صحية محمية؟

سيحتاج فريق التطوير إلى تحديد جميع المتطلبات التنظيمية التي يجب الالتزام بها، ثم تطوير النهج ومجموعة العمليات والتي بدورها تلبي تلك المتطلبات قبل الشروع في أي تطوير. سواء كان فريق التطوير يستخدم النهج الشلال التقليدي أو النهج الرشيق Agile أو Waterfall لتطوير البرمجيات فالمتطلبات التنظيمية لا تتغير. بمجرد تحديد المتطلبات التنظيمية regulatory requirements، يمكن توثيقها على أنها قصص مستخدم غير وظيفية non-functional user stories. تشير المتطلبات التنظيمية وخصوصية بيانات المريض الي أن أمان أي برنامج تم تطويره للرعاية الصحية يجب تحديده واختباره بشكل مناسب. الآن، على الرغم من كونها مثالية عند تطوير نموذج رعاية صحية لتحديد جميع متطلبات المنتج مسبقاً، إلا أن برامج الرعاية الصحية غالبًا ما تكون معقدة؛ وستتضمن التكامل مع الأنظمة الأخرى وغالبًا ما تكون مهمة جداً [4]. يواجه المطورون العديد من التحديات عند تطوير مشروعات الرعاية الصحية، من أهم هذه التحديات ما يلي:

- سرية البيانات: تطبيقات الرعاية الصحية، غالباً ما تتضمن البيانات الأكثر حساسية. على سبيل المثال، مراقبة معلومات كل شخص وصحته وقراءاته التشخيصية. يضمن قانون التأمين الصحي (HIPPA) مجموعة من الإرشادات لضمان خصوصية المعلومات وأمنها. يعد أمان بيانات ومعلومات المريض مطلباً واضحاً لنظام الرعاية الصحية، لذلك يجب أن يتوافق النظام مع HIPPA. لذا من المهم وضع خطة تصف الإجراءات والخطوات الحاسمة لحماية معلومات المريض وكيفية جمع البيانات، ما هي البيانات التي يحتاجها النظام ليعمل بفعالية كما يجب أن تقصر البيانات على المعلومات الهامة فقط [5].
- إدارة البيانات الضخمة: تساعد البيانات الضخمة في مجال الرعاية الصحية على التنبؤ بالأوبئة، وتحسين مستوى الحياة، وتطوير الأدوية لعلاج الأمراض. زيادة البيانات في المجالات الطبية ومجالات الرعاية الصحية تثرى البيانات الخاصة بالموقع المركزي والتي تتيح استخدامها في المنظمات الأخرى المرتبطة بنظام الرعاية الصحية مثل شركات التأمين وشركات الأدوية والتي تجعل مؤسسات الرعاية الصحية وكل

منظمة مرتبطة بها لديها قدر هائل من المعرفة التي يجب تنظيمها وإدارتها من أجل استخدامها في كثير من المجالات. على الرغم من أن هذه البيانات تعتبر المنفذ الرئيسي لجعل أداء الخدمات في مجال الرعاية الصحية أكثر فاعلية. لكن قضايا الأمان والخصوصية تتزايد مما يعوق منظمة الصحة على الاستفادة منها [6].

- التعامل مع تقلبات / اتجاهات السوق: كل يوم، ندرك الاتجاهات الجديدة في مجال الرعاية الصحية التي تؤدي إلى التقلبات وعدم اليقين والتعقيد والغموض في غالبية قطاعات الرعاية الصحية. يساهم هذا التقلب المستمر في التغيير في الممارسات المعيشية، وتنوع الأجهزة التي يتم تقديمها، والتقنيات الجديدة، والأفكار، والتوافق. يؤدي تحول الرعاية الصحية إلى إحداث ثورة، لذا يتعين على قادة القطاع الصحي التعامل مع جميع السيناريوهات. وبالتالي، فإن مواكبة الاتجاهات والحفاظ على الأداء الصحي في نفس الوقت أمر أساسي للبقاء في المقدمة في المجال [7، 8].
- التكامل مع نظم تكنولوجيا المعلومات: لتحويل نظام الرعاية الصحية إلى نظام تكنولوجي لديه القدرة علي قابلية التشغيل البيني يجب أن يتم من خلال الحدود التنظيمية. على سبيل المثال: يجب أن يرسل نظام المراقبة عن بعد المعلومات إلى نظام السجلات الطبية الإلكترونية، والذي يمكن بعد ذلك التعرف عليه من خلال نظام السجل الصحي الإلكتروني. قابلية التشغيل البيني، يمكن أن تُعرف باسم قدرة النظام الصحي على العمل مع سلسلة معالجة البيانات، على الرغم من أن نظام الرعاية الصحية كفء في العمل، إلا أنه يتطلب الكثير من التكلفة [9].

3- الأعمال ذات الصلة

تطورت صناعة الرعاية الصحية لتقديم أنظمة رعاية صحية إلكترونية ذكية لدعم تقديم الخدمة وكذلك لتحقيق رضا العملاء. قام العديد من الباحثين ببذل الجهود الكثيرة في مجال إمكانيات بناء أنظمة رعاية صحية قادرة على تحسين الخدمات الصحية ومواجهة التحديات والتغيرات في البيئة وإنتاج حل برمجي مناسب، لذلك يتم تطوير البرمجيات باستخدام مجموعة متنوعة من نماذج ونهج التطوير.

في العديد من الدراسات السابقة اقترح الباحثون منهجيات تطوير البرمجيات في مجال الرعاية الصحية. لتقديم أساليب تطوير برمجيات مبتكرة وحديثة، من المفترض أنها ستعزز كفاءة وإنتاجية مطوري البرمجيات.

- قام V. Kadurin بعرض المنهجيات الهامة والشائعة المستخدمة في تطوير البرمجيات مع إبراز الفروق بينهم وعرض مزايا وعيوب كل منهجية وكيف يمكن الاستفادة منها في المواقف المختلفة. اقترح أيضاً منهجية جديدة لتطوير البرمجيات تجمع بين أسلوب ال Agile ومنهجية تقليدية وأكثر تنظيماً مثل ال Waterfall. يتضمن النهج الجديد 4 مراحل مختلفة يتم تنفيذها بالتسلسل، ولكن لها أنماط تكرارية داخلياً، هي: (1) خطة التحليل، (2) الهندسة المعمارية للنظام، (3) التطوير، (4) التحقق من الصحة. وبالتالي فإن هذا الأسلوب يناسب متطلبات مجال الرعاية الصحية المعقدة [4] .
- قام الباحثون Lalband, Neelu, and D. Kavitha بتقديم مناقشة موجزة حول نماذج دورة حياة تطوير البرمجيات المختلفة التي تتبع في صناعة البرمجيات. على الرغم من وجود العديد من طرق التطوير، إلا أنها ركزت فقط على الأساليب الشائعة والأساسية في هذا المجال. وإجراء مقارنة بين هذه الأساليب واختيار منهجية Agile الأكثر ملاءمة لتطوير نظام الرعاية الصحية. كما تم تقديم استبيان تضمن مناقشة حول التوسع والتحول في مجال الرعاية الصحية بالإضافة إلى تحديات الرعاية الصحية الذكية. كما أخذوا في الاعتبار القضايا التي يجب مراعاتها عند تطوير نظام الرعاية الصحية [10].
- قام الباحثون Abbas et al بتوضيح الهدف الرئيسي لتطوير برامج الرعاية الصحية وهو تزويد المرضى بأفضل خدمة ممكنة من خلال جمع وتفسير المعلومات الدقيقة. ركزوا على تعريف أولي لصفات الجدارة بالثقة من الأعمال البحثية السابقة. ثم حاولوا البحث عن العوامل والأسباب التي تحدد الجدارة بالثقة في حلول برمجيات الرعاية الصحية. كما تم اقتراح نموذج برنامج رعاية صحية جدير بالثقة لدعم أصحاب المصلحة في تكنولوجيا الرعاية الصحية، والذي يمكن تطبيقه لتقدير مستوى الجدارة بالثقة المرتبط بحلول برامج الرعاية الصحية[11].
- قام الباحثون Jabbar et al بعرض ومناقشة التحديات التي تواجه مطوري نظام الرعاية الصحية خلال عملية التطوير. ثم اقترحوا نموذجاً قائماً على المنهجية الرشيقة SCRUM لمساعدة فريق تطوير البرمجيات على بناء أنظمة رعاية صحية، يتم استخدامه مع تقنيات إدارة المعرفة لتحقيق تعزيز معايير الجودة المطلوبة والحفاظ عليها [12].
- قام الباحثون McCaffery et al بتسليط الضوء على القضايا في تطوير برامج الأجهزة الطبية التقليدية في قطاع الرعاية الصحية. وناقشوا الفوائد والتحديات التي من المحتمل أن تتم ملاحظتها عند تنفيذ نموذج Waterfall/V-Model في دورة حياة تطوير برامج الأجهزة الطبية. هذه النماذج تم تبنيها على نطاق واسع في تطوير برامج الأجهزة الطبية، وقد حاول المؤلفون التركيز على أهمية استخدام النهج

الرشيق Agile في مجال الأجهزة الطبية لتشجيع الممارسين على تقييم فوائد أداء تطوير البرمجيات التي تمت بهذا النهج. ثم ناقشوا التحديات وركزوا على ممارسات ال Agile التي تم تبنيها بنجاح في صناعة برمجيات الأجهزة الطبية [13]. جدول رقم (1) يعرض مخلص الأعمال ذات الصلة.

جدول (1): مخلص الأعمال ذات الصلة

الباحثون	ملخص العمل
V. Kadurin 2021	اقترح منهجية جديدة لتطوير البرمجيات تجمع بين أسلوب ال Agile ومنهجية أكثر تنظيماً.
Lalband, Neelu, and D. Kavitha 2019	اشتمل الاستبيان على مناقشة حول التوسع والتحول في مجال الرعاية الصحية بالإضافة إلى تحديات الرعاية الصحية الذكية.
Abbas et al 2017	نموذج برمجيات رعاية صحية جدير بالثقة لدعم أصحاب المصلحة في تكنولوجيا الرعاية الصحية.
Jabbar et al 2019	اقترح نموذجًا قائمًا على SCRUM لمساعدة فريق تطوير البرامج على بناء أنظمة الرعاية الصحية.
McCaffery et al 2016	ناقش الفوائد والتحديات التي من المحتمل ملاحظتها عند تنفيذ نموذج Waterfall/V-Model في دورة حياة تطوير برامج الأجهزة الطبية.

4- منهجيات تطوير البرمجيات

يتم تطوير البرامج باستخدام مجموعة متنوعة من الأساليب. يشار إلى هذه الأساليب بمنهجيات تطوير البرمجيات. يتم التعرف على منهجيات تطوير البرمجيات كمخططات تُستخدم لتخطيط وبناء وتنظيم ومراقبة جميع عمليات تطوير البرنامج [4]. إن منهجيات التطوير هذه متنوعة، لكل منها مزاياها وعيوبها وقد مرت بتطورات على مر السنين. يمكن تعريف منهجية هندسة البرمجيات على أنها مجموعة من التقنيات والأدوات المستخدمة في تطوير البرمجيات. ويجب أن تصف المنهجية كل مرحلة من مراحل دورة حياة تطوير البرمجيات. لا تحدد منهجية البرامج عادة التفاصيل. تُترك هذه التفاصيل عادةً لاحتياجات ومتطلبات المنظمة [14]. يتم تطوير البرمجيات عادة من خلال عدة مراحل وهذه المراحل هي [15]:

1. تحديد المشكلة وتحليلها.
2. استفسارات السوق (البحث)
3. استخلاص المتطلبات وتحصيلها.

4. تصميم.
5. تطبيق.
6. اختبارات.
7. تعيين.
8. صيانة وإصلاح الخلل.

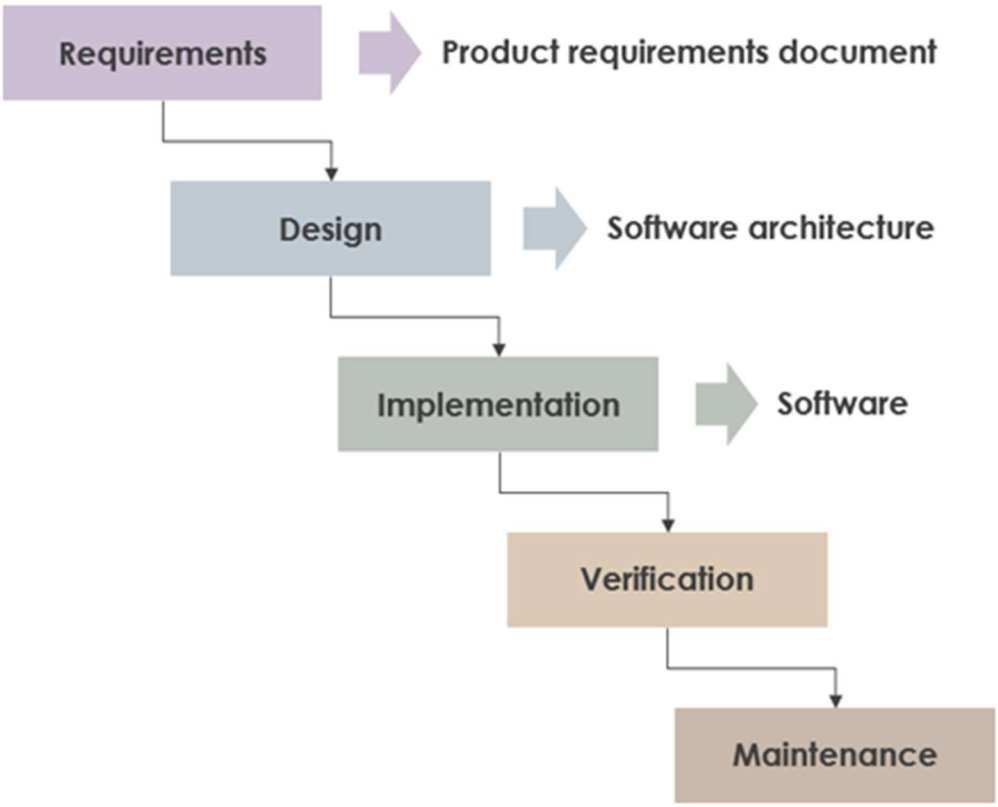
عادة ما تستخدم هذه المراحل كخطوات ضمن منهجيات التطوير. استخدمت بعض المنهجيات هذه الخطوات بطريقة خطية أو متتالية وتستخدم المنهجيات الأخرى بعض هذه الخطوات أو الخطوات الإضافية المضافة إلى تلك المذكورة أعلاه. في هذه الدراسة، سيتم إجراء مناقشة حول منهجيات تطوير البرمجيات ونقاط القوة والضعف فيها، ويتم التركيز على منهجيات تطوير البرمجيات التالية:

(1) نموذج Waterfall

(2) منهجيات Agile

1-4 نموذج Waterfall

كانت منهجية تطويرWaterfall طريقة تطوير معروفة وشائعة الاستخدام منذ تعريفها في عام 1970 بواسطة Winston W. Royce [16]. تستخدم هذه المنهجية النهج المخطط لها وتُعرف أيضًا باسم "نموذج دورة الحياة الخطية المتسلسلة" [4]. المراحل المتسلسلة في نموذج Waterfall كما هو موضح في الشكل رقم (1).



الشكل (1): المراحل المتسلسلة في نموذج Waterfall

- جمع وتحليل المتطلبات: تشمل جمع كل المتطلبات الممكنة للنظام المطلوب تطويره.
- تصميم النظام: يتم دراسة مواصفات المتطلبات التي تم تجميعها من المرحلة الأولى، بحيث يصبح تصميم النظام جاهزاً. يساعد تصميم النظام في تحديد بنية النظام الشاملة، مثل تحديد متطلبات الأجهزة والنظام.
- تطبيق: باستخدام مخرجات المرحلة السابقة (تصميم النظام) كمدخلات في المرحلة الحالية، يتم تطوير النظام أولاً في برامج صغيرة تسمى الوحدات، والتي يتم دمجها في المرحلة اللاحقة.

- التكامل والاختبار: يتم دمج جميع الوحدات المطورة ضمن مرحلة التنفيذ في نظام واحد وذلك بعد اختبار كل وحدة أولاً.
- انتشار النظام: بمجرد إجراء الاختبار الوظيفي وغير الوظيفي؛ يتم نشر المنتج في بيئة العميل أو طرحه في السوق.
- أعمال صيانة: هناك بعض المشكلات التي تظهر أثناء التشغيل الفعلي في بيئة العميل، لإصلاح هذه المشكلات، يتم القيام التصحيحات أو التحسينات، ثم يتم إجراء الصيانة لتقديم هذه التغييرات داخل بيئة العميل.

فيما يلي بعض المزايا الرئيسية لنموذج Waterfall [17]:

- نموذج ذو عمليات، ومراحل بسيطة، وسهلة الفهم، والاستخدام.
- سهل الإدارة.
- تتم معالجة المراحل وإكمالها واحدة تلو الأخرى.
- المراحل محددة بوضوح.
- مفهومة جيداً موثقة جيداً.
- سهولة إعادة ترتيب المهام.

العيوب الرئيسية لنموذج Waterfall هي كما يلي [17]:

- لا يتم إنتاج أي برنامج عمل حتى وقت متأخر خلال دورة الحياة.
- كميات عالية من المخاطر وعدم اليقين.
- ليس نموذجاً مثالياً للمشروعات المعقدة الموجهة للكائنات object-oriented.
- نموذج ضعيف للمشاريع الطويلة والجارية.
- غير مناسب للمشاريع التي تكون فيها المتطلبات معرضة لخطر تعديل متوسط إلى مرتفع.
- من الصعب قياس التقدم خلال المراحل.
- لا يمكن استيعاب المتطلبات المتغيرة.

2-4 منهجيات التطوير الرشيقة Agile

يعتبر نهج Agile هو القدرة على خلق التغيير والاستجابة له. فهو عبارة عن طريقة للعمل والنجاح في وسط بيئة مضطربة ديناميكية في شركات الأعمال تتسم بالتغييرات الكثيرة. لقد تم تصميم نهج Agile خصيصاً لدعم تطوير تطبيقات الأعمال حيث إن متطلبات هذا النظام عادة ما تتغير بشكل سريع أثناء عملية التطوير [18]. يعد تطوير برمجيات Agile في الأساس هو نهج متكرر لإدارة المشروعات وتطوير البرمجيات، ويقوم علي: المواصفات التدريجية أو المتزايدة Incremental أثناء التطوير، والتصميم، والبرمجة. كما يتطلب التكامل الشامل لكل من التقييم والتطوير لمساعدة فرق العمل على تقديم خدمة ذات قيمة عالية لعملائهم بشكل أسرع ومجهود أقل [19].

ويتم تقييم كلا من المتطلبات والخطط والنتائج باستمرار مما يساعد في تكوين آلية تلقائية لدى فرق العمل تجاه الاستجابة للتغيير بشكل سريع من خلال التعاون بين فرق العمل ذاتية التنظيم ومتعددة الوظائف. لقد تحولت منهجيات Agile وأحدثت تغييرات فريدة من نوعها في مجال تطوير البرمجيات من خلال التأكيد بقوة على تعاون الفريق ومشاركة العميل وتحمل تائج التغيير .[20] تقوم منهجيات Agile بتعزيز كلا من التخطيط التكيفي، التطوير والتنفيذ التدريجي، نهج متكرر ذو إطار زمني. كما أنه يقوم بالتشجيع على الاستجابة السريعة والمرنة للتغيير [21]. حيث تشتمل كل عملية تكرارية ذو إطار زمني على تخطيط، تحليل المتطلبات وعمل تصميم لها، تشفير وعمل اختبار.

أصبحت تقنيات تطوير برمجيات Agile أكثر شيوعاً في الآونة الأخيرة. أما بالنسبة لإدارة المشروعات التقليدية، يتم استخدام منهجيات Agile على نطاق واسع في تطوير البرمجيات من أجل تحسين جودة المشروع وتعزيز إرضاء العميل [22]. لم تعد الطرق التقليدية مثل (Waterfall, Unified Process, Prototyping Model and Spiral Model) مناسبة لتطوير البرمجيات في الوقت الحالي بسبب متطلبات السوق. بينما تشتمل تقنيات تطوير البرمجيات الحديثة Agile علي: XP, Scrum, Crystal, FDD, DSDM, and ASD [23].

تعد مساعدة فرق عمل البرمجة على التفكير بشكل مختلف والعمل بشكل متكرر هما الهدف من منهجيات Agile. من أجل الحصول على هذا الهدف، تم اقتراح أربعة قيم أساسية تابعة لنظام Agile تؤدي إلى اثني

عشر مبدأ. تتكون القيم الخاصة بنظام Agile مما يلي: الكفاءة، وتقديم الخدمة في الوقت المحدد، والحفاظ على التعلم وإعادة التعلم من السابق [24]. يسلط بيان Agile الضوء على أربع قيم [25]:

1. الأفراد والتفاعلات تجاه العمليات والأدوات.
2. برمجيات العمل حول وثائق شاملة.
3. تعاون العملاء حول التفاوض على العقد.
4. الاستجابة للتغيير طبقا للخطة.

تبدأ عملية تطوير برمجيات Agile دائماً بتحديد المستخدمين وتوثيق الرؤية الحالية حول نطاق المشاكل والفرص والقيم التي يجب معالجتها. ومن ثم يستخدم مالك المنتج تلك الرؤية ويعمل مع فريق متعدد التخصصات (أو فرق) لتقديمها. الأدوار في هذه العملية هي: المستخدم ومالك المنتج وفريق تطوير البرمجيات. المزايا الرئيسية التي تحققت عن طريق تطوير البرمجيات القائمة على Agile [26] [27] هي:

- تحسين الاتصال والتنسيق بين أعضاء الفريق.
- الإصدارات السريعة.
- مرونة التصميم.
- عملية أكثر منطقية.
- مرنة بما يكفي لتحديث دورة حياتها اعتمادًا على متطلبات المستخدمين المتغيرة.
- يعود الفضل في رضا العملاء إلى الإصدارات المتصاعدة.
- تساعد آراء العملاء بعد كل إصدار على إنتاج منتجًا ذو جودة عالية.
- يوفر الوقت بسبب الوثائق الموجزة.
- أقل خطورة بسبب آراء العملاء المتكررة.
- يبحث بسرعة عن الأخطاء ويقوم بإصلاحها.

تتمثل التحديات الرئيسية لتطوير البرمجيات القائمة على Agile فيما يلي:

- يجب أن يكون العميل واضحاً بشأن المنتج ومواصفاته.
- من الصعب على المطورين الجدد في فريق العمل فهم الوثائق الموجزة.
- إذا لم يكن الإصدار مناسباً لتوقعات العميل فهذا يعد إضاعة الوقت والموارد والمجهود أيضاً.

توجد أنواع مختلفة من الأساليب Agile. تشترك هذه الأساليب في الأفكار والقيم المشتركة، ولكن مع خصائص دقيقة بينها [2].

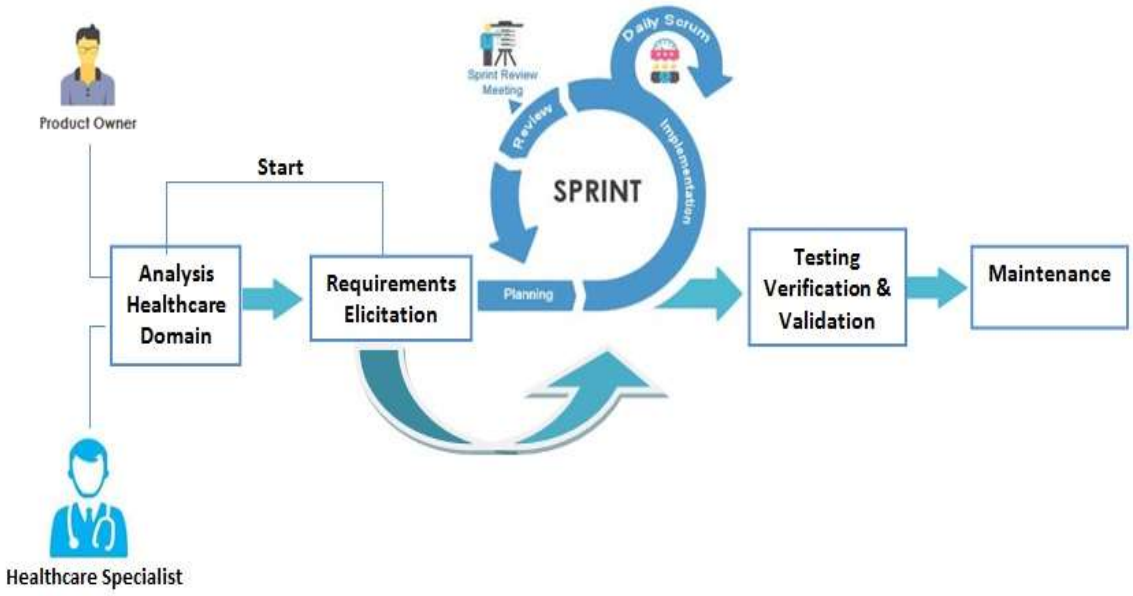
5- نموذج Hybrid SCRUM-FALL

تم تشكيل نموذج Scrum-Fall الهجين بناءً على تكامل نموذج تطوير البرمجيات Waterfall ونهج Agile مع مراعاة النقاط الرئيسية الحاسمة لنجاح نظام الرعاية الصحية. الهدف الرئيسي لتطبيق تكنولوجيا المعلومات في الرعاية الصحية هو تحسين جودة الرعاية الطبية. لذلك، يجب أن يتميز نظام الرعاية الصحية بنظام عالي الموثوقية ومستقر وآمن وعالي الأداء وتوافر عالي.

يمكن أن يكون نموذج Hybrid Scrum-Fall مفيداً في مشروعات برمجيات برامج الرعاية الصحية لأنه يستخدم مزايا كل من نماذج Waterfall ونهج agile باستخدام Scrum. يتم تضمين بعض الخطوات الجديدة في النموذج المقترح بالإضافة إلى دمج بعض هذه الخطوات معاً. بالنسبة لمشروعات البرمجيات الكبيرة التي تتطلب وقتاً طويلاً للتطوير، فإن هذا النموذج مناسب لمثل هذه المشروعات [28].

يجمع نموذج Hybrid Scrum-Fall بين الخصائص الرئيسية لنموذج Waterfall وAgile. هكذا، يمكن أن يعمل نموذج الشلال Waterfall بشكل جيد مع مجال واسع النطاق مثل أنظمة الرعاية الصحية، ويحقق أماناً عالياً، وتحدث الخطوات بالتسلسل، وسهلة التنفيذ، والاستخدام والإدارة بالإضافة إلى تحديد المتطلبات وتوثيقها بوضوح [29]. تعتبر أساليب العمل بنهج Agile تكرارية وتعاونية ومحددة زمنياً وسهلة التغيير، كما يمكن رؤيتها كنتيجة للحاجة إلى تطوير منتج أسرع وأكثر تركيزاً على المخرجات وهي الآن جزء من استراتيجيات إدارة تحول المؤسسة الرقمية للرعاية الصحية. نهج Agile كأسلوب عمل، يلعب scrum دوراً

مركزياً في تطوير الرعاية الصحية. يوضح الشكل رقم (2) نظرة شاملة لنموذج التطوير المقترح **Error! Reference source not found.**.



الشكل (2): نظرة شاملة لنموذج التطوير المقترح**Error! Reference source not found.**

- **تحليل مجال الرعاية الصحية Analysis Healthcare Domain** يحاول فريق تطوير الرعاية الصحية فهم المجال العام حيث يتم تطبيق النظام. كما يتم تحديد العناصر ذات الصلة بمجال تطبيق الرعاية الصحية. يدرك فريق التطوير مجال الرعاية الصحية.
- **استنباط المتطلبات Requirements Elicitation:** هذه هي المرحلة الأكثر أهمية حيث إن هدفها الرئيسي هو ربط المعرفة الخاصة بالرعاية الصحية بالتنفيذ الفني للمنتج النهائي. يقوم فريق تطوير المنتج بتحليل المتطلبات بناءً على مشاركة أخصائي الرعاية الصحية، ويفهم المشكلات تماماً. في مجال الرعاية الصحية، تعتبر الدقة والجودة العالية في غاية الأهمية لأنها تتعلق بحياة الإنسان وصحته. ستنشئ هذه المرحلة أيضاً قنوات الاتصال المناسبة لفريق الرعاية الصحية الشخصي وفريق التطوير من أجل حل أي مشكلات قد تظهر في المستقبل. سيكون الناتج في هذه المرحلة وثيقة مفصلة لجميع المتطلبات

المكتوبة بطريقة واضحة بما يكفي ليسهل فهمها من قبل الأشخاص غير الأطباء. يحاول الفريق طرح جميع الأسئلة وتأمين جميع الإجابات التي يحتاجونها لبناء متطلبات المنتج [4]. في هذه المرحلة، يجب على الفريق:

1. تحديد رؤية المنتج والمتطلبات الوظيفية.
2. القيام بميزة العصف الذهني وتحديد الأولويات وتخطيط العمل.
3. فهم تفاصيل مشكلة عميل الرعاية الصحية المحددة.
4. فهم كيفية تفاعل الأنظمة والمساهمة في أهداف العمل العامة.
5. فهم احتياجات وقيود أصحاب المصلحة في النظام.
6. فهم الاحتياجات المحددة للأشخاص الذين يحتاجون إلى دعم النظام في عملهم.

- **Sprint:** Scrum هو إطار عمل agile قائم على التجربة، منظم على أساس نظرية التحكم في التجريبية تحت مظلة Agility، مما يساعد في معالجة العمليات المعقدة وزيادة الإنتاجية وتحسين جودة تطوير منتجات البرمجيات. يعتمد نهج scrum على المبادئ والقيم الخاصة بال Agile، والتي تعزز أسلوباً مختلفاً لإدارة مهام تطوير البرمجيات، والتعاون في التفاوض على العقود، والاستجابة للتغييرات على الخطة الشاملة [30]. تعد مدة sprint مهمة جداً بحيث تكون قصص المستخدم صغيرة قدر الإمكان. متوسط فترة sprint النموذجية تستغرق حوالي أسبوعين. إذا كانت فترة sprint أصغر فهي ميزة، حيث يمكن تلقي المزيد من تعليقات المستخدمين ويمكن معالجة الأعطال والأخطاء ومعالجتها في وقت مبكر قدر الإمكان. إذا كانت فترة sprint أطول فإنه يسمح للمطور بالعمل بشكل كامل. تتمثل المرحلة التالية في إنشاء Sprint backlog حيث يجب على فريق scrum اختيار قصص العملاء المهمة وتحويلها إلى مهام أصغر. إنهم بحاجة إلى التخطيط لكيفية إنجاز المهمة. يتم تحويل قصص المستخدم الفعلية كمهام صغيرة في Sprint backlog حيث يبدأ العمل الفعلي. في هذه الخطوة، تعتبر اجتماعات scrum مهمة لأنها تتم لتتبع حالة العمل ومن يقوم بأي حالة. يجب التحقيق في المهام المكتملة كمنتج عامل مع اختبار دورة الحياة الكاملة [31].
- **الاختبار - المصادقة والتحقق Testing-Verification and Validation:** في هذه المرحلة، يتم إجراء المصادقة والتحقق من قبل فريق مشترك من المطورين ومهندسي ضمان الجودة والمستخدمين

النهائيين وموظفي الرعاية الصحية. هذا يضمن مراجعة جميع جوانب المتطلبات - من الناحية الفنية والطبية.

- **أعمال صيانة Maintenance:** المرحلة الأخيرة من نموذج Scrum-Fall هي الصيانة. هذا للتأكد من أن البرنامج يعمل بمستويات الأداء المثلى. يمكن أن يتعلق ذلك بالأخطاء الموجودة في البرنامج أو مشكلات الأمان أو لمعالجة المشكلات التي ظهرت فقط عند تنفيذ النظام الجديد. تشمل الأنشطة في هذه المرحلة الحصول على طلبات الصيانة من المستخدمين النهائيين، وتحويل هذه الطلبات إلى تغييرات، وتصميم التغييرات وتنفيذ التغييرات المذكورة أخيراً.

6- الملخص والأعمال المستقبلية

تسعي منظمات تطوير برنامج الرعاية الصحية إلى الوصول للحد الأمثل من تطوير إدارة هندسة البرمجيات. يتجه نموذج Scrum في تنمية وإدارة مشروعات البرمجة إلي منهجيات Agile.لقد تناول هذا البحث نموذج هجين لتطوير البرمجيات (Hybrid Scrum-Fall model). يعد نموذج Scrum-Fall تكاملاً لكلاً من نموذج waterfall و Scrum في إطار عمل هجين للتغلب علي الحدود التي تم وضعها بواسطه كلا النموذجين بالإضافة إلي اعتباره بمثابة تمكين أفضل لمطوري برمجيات الرعاية الصحية. مقارنة بإطار العمل القائم بذاته، فالنهج المقترح Scrum-Fall هو الأكثر قوة وقدرة على مواجهة تحديات تطوير البرمجيات حيث إنه يتيح الفرصة للتعاون مع متخصص في مجال الرعاية الصحية. يشتمل النهج المقترح Scrum-Fall على خمس مراحل محددة هي: تحليل مجال الرعاية الصحية، استنباط المتطلبات، السرعة Sprint، والاختبار (التحقق والمصادقة)، وأعمال الصيانة. يتمتع ذلك النظام الهجين بمميزات وفوائد منهجيات كلا من Waterfall وAgile لذا فهو الأمثل لنظم الرعاية الصحية المعقدة. من الممكن أن يكون البحث التالي في ذلك النطاق يتناول طبيعة وخصائص نظم الرعاية الصحية والتعاون بين كلا من أخصائيين الرعاية الصحية ومهندسي البرمجيات في إطار الوقت المحدد لإدارة مشروع الرعاية الصحية.

سوف نسعى لتفعيل واختبار هذا النموذج المقدم علي حالة دراسية حقيقية. حيث ستحل وسيلة أخري فعالة محل المستخدمة حالياً في عملية تطوير البرمجيات والتي بدورها تقدم حلول أكثر دقة وذو أداء أعلي للنظام بأكمله.

المراجع

[1] Y. B. Leau, W. K. Loo, W. Y. Tham, and S. F. Tan, "Software development life cycle Agile vs Traditional approaches."

[2] P. K. Yeng, S. D. Wolthusen, and B. Yang, "Comparative Analysis of Software Development Methodologies for Security Requirement Analysis: Towards Healthcare Security Practice," 2020.

[3] S. Tolf, M. E. Nyström, C. Tishelman, M. Brommels, and J. Hansson, "Agile, A Guiding Principle for Health Care Improvement?", International Journal of Health Care Quality Assurance, 2015.

[4] V. Kadurin, "A new proposed software development methodology for healthcare industry", in AIP Conference Proceedings, 2021, Vol. 2333, No. 1, p. 030010: AIP Publishing LLC.

[5] K. Abouelmehdi, A. Beni-Hessane, and H. Khaloufi, "Big healthcare data: preserving security and privacy", Journal of Big Data, Vol. 5, No. 1, pp. 1-18, 2018.

[6] K. Abouelmehdi, A. Beni-Hssane, H. Khaloufi, and M. Saadi, "Big data security and privacy in healthcare: a review", Procedia Computer Science, Vol. 113, pp. 73-80, 2017.

[7] P. Turner, "The ecology of healthcare", Leadership in Healthcare, Springer, 2019, pp. 17-43.

[8] A. Kumari, S. Tanwar, S. Tyagi, and N. Kumar, "Fog computing for Healthcare 4.0 environment: Opportunities and challenges", Computers & Electrical Engineering, Vol. 72, pp. 1-13, 2018.

[9] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications", 2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC), 2017, pp. 1-5: IEEE.

[10] N. Lalband and D. Kavitha, "Software engineering for smart healthcare applications", International Journal of Innovative Technology and Exploring Engineering, Vol. 8, pp. 325-331, 2019.

[11] R. M. Abbas, N. Carroll, I. Richardson, and S. Beecham, "The Need for Trustworthiness Models in Healthcare Software Solutions", in HEALTHINF, 2017, Vol. 5, pp. 451-456.

[12] T. Jabbar, Y. Hafeez, A. A. Kiani, N. Anwar, and J. Javaid, "Use of Knowledge Management and SCRUM techniques to increase the reusability in software development", 2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), 2019, pp. 1-5: IEEE.

[13] F. McCaffery, K. Trektere, and O. Ozcan-Top, "Agile–Is it Suitable for Medical Device Software Development?", International Conference on Software Process Improvement and Capability Determination, 2016, pp. 417-422: Springer.

[14] A. Al-Dahmash and S. El-Masri, "A New Proposed Software Engineering Methodology for Healthcare Applications Development", Int. J. Mod. Eng. Res, Vol. 3, No. 3, pp. 1566-1570, 2013.

[15] C. Jones, A Guide to Selecting Software Measures and Metrics. CRC Press, 2017.

[16] B. Davis, Agile Practices for Waterfall Projects: Shifting Processes for Competitive Advantage. J. Ross Publishing, 2012.

[17] S. Shylesh, "A Study of Software Development Life Cycle Process Models", in National Conference on Reinventing Opportunities in Management, IT, and Social Sciences, 2017, pp. 534-541.

[18] D. Mohamed and N. R. Darwish, "Extracting CRM requirements–Waterfall or Agile: A Comparative Study", International Research Journal of Advanced Engineering and Science, Vol. 4, pp. 1-5, 2019.

[19] F. Lonetti and E. Marchetti, "Emerging Software Testing Technologies", Advances in computers, Vol. 108: Elsevier, 2018, pp. 91-143.

[20] Ö. Uludag, M. Kleehaus, C. Caprano, and F. Matthes, "Identifying and Structuring Challenges in Large-Scale Agile Development based on a Structured Literature Review", 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC), 2018, pp. 191-197: IEEE.

[21] S. Singh, I. J. I. J. o. u.-a. e.-S. Chana, Science, and Technology, "Introducing Agility in Cloud-Based Software Development Through ASD", Vol. 6, No. 5, pp. 191-202, 2013.

[22] S. A. Butt, "Study of Agile Methodology with The Cloud", Pacific Science Review B: Humanities and Social Sciences, Vol. 2, No. 1, pp. 22-28, 2016.

[23] P. Mani and S. Deebitha, "Analysis of Agile Software Development Utilizing Cloud Computing Capabilities", ed: Nr, 2014.

[24] A. Manifesto, "Manifesto for Agile Software Development, 2001", URL: http://agilemanifesto. org, 2010.

[25] P. Rodríguez, M. Mäntylä, M. Oivo, L. E. Lwakatare, P. Seppänen, and P. Kuvaja, "Advances in Using Agile and Lean Processes for Software Development", Advances in Computers, Vol. 113: Elsevier, 2019, pp. 135-224.

[26] S. Al-Saqqa, S. Sawalha, and H. AbdelNabi, "Agile Software Development: Methodologies and Trends", International Journal of Interactive Mobile Technologies, Vol. 14, No. 11, 2020.

[27] D. Jagli and S. Yeddu, "CloudSDLC: Cloud Software Development Life Cycle", International Journal of Computer Applications, Vol. 168, No. 8, pp. 6-10, 2017.

[28] M. S. J. D. S. J. Gharajeh, "Waterative model: An integration of the waterfall and iterative software development paradigms," Vol. 10, pp. 75-81, 2019.

[29] A. Lawal and R. C. J. B. J. O. G. S. Ogbu, "A Comparative Analysis of Agile and Waterfall Software Development Methodologies", Vol. 11, No. 2, pp. 1-2, 2021.

[30] N. R. Darwish and S. Megahed, "Requirements Engineering in Scrum Framework", International Journal of Computer Applications, Vol. 149, pp. 24-29, 2016.

[31] K. Schwaber and J. Sutherland, "The Scrum Guide. Scrum. org. 28.08. 2017", ed, 2016