

AI PROJECT MANGER

إعداد الطالب:

أحمد محمد باسل الطباع

محمد عدنان لحام

إشراف:

د. اصف جعفر

2025\2026

شهادة مشرف :

اسم :

تاريخ :

التوقيع:

الملخص

تعدّ أنظمة دعم القرار الذكية أحد المجالات التطبيقية المتقدمة في اختصاص الذكاء الصنعي، لما توفره من إمكانيات لتحليل البيانات غير المهيكلة، وفهم السياق، والمساعدة في اتخاذ قرارات معقدة تحاكي التفكير البشري. وفي ظل التحديات المتزايدة في إدارة المشاريع البرمجية الحديثة، يبرز دور تقنيات الذكاء الصنعي في تحسين التخطيط، وتقدير المخاطر، وتنظيم عملية التطوير وفق منهجيات علمية واضحة.

في هذا المشروع، يتم تطوير وكيل ذكاء صنعي واحد يعمل كمدير مشروع برمجي ذكي، يعتمد على نموذج لغوي كبير لتحليل الأوصاف النصية للمشاريع البرمجية، مثل مستندات المتطلبات أو ملفات README يقوم الوكيل بفهم أهداف المشروع، والتقنيات المتوقعة، والمكونات الأساسية، ومستوى التعقيد، والمخاطر المحتملة، وذلك من خلال معالجة لغوية دلالية متقدمة.

ولتعزيز دقة التحليل وربط القرارات الإدارية بالمعرفة المنهجية، تم دمج آلية التوليد المعزّز بالاسترجاع-Retrieval (Retrieval)، والتي تتيح للوكيل استرجاع مفاهيم Agile و Scrum ذات الصلة واستخدامها لدعم عملية التخطيط واتخاذ القرار. ونتيجة لذلك، يقوم النظام بتوليد خطط سبرنت وتقارير إدارية منظمة تعكس مبادئ إدارة المشاريع البرمجية الحديثة.

يعتمد النظام على مخرجات مقيّدة ببنية ثابتة قابلة للتحقق آلياً، مما يضمن الاتساق المنطقي بين مدخلات المشروع ومخرجات التحليل، ويحدّ من الانحراف الدلالي الشائع في النماذج اللغوية العامة. ويتم تقييم أداء الوكيل، وملاءمة التحليل المقترن، وقدرته على دعم قرارات إدارة المشروع بشكل قابل للتفسير.

في RAG يقدم هذا المشروع مثلاً تطبيقياً على توظيف تقنيات الذكاء الصنعي الحديثة، ولا سيما النماذج اللغوية الكبيرة وبناء أنظمة ذكية لدعم القرار في مجال هندسة البرمجيات، مع استعراض التحديات الحالية والأفاق المستقبلية لتطوير وكلاء ذكاء صنعي أكثر دقة وموثوقية.

Abstract

Intelligent decision-support systems represent an advanced applied research area within the field of Artificial Intelligence, as they enable the analysis of unstructured data, contextual understanding, and support for complex decision-making processes that resemble human reasoning. With the increasing complexity of modern software projects, the application of artificial intelligence techniques has become essential for improving project planning, risk assessment, and development organization using systematic methodologies.

In this project, as part of the Artificial Intelligence specialization, a single AI agent is developed to function as an intelligent software project manager. The agent relies on a large language model to analyze textual software project descriptions, such as requirement documents or README files. It performs semantic understanding of project goals, expected technologies, core system components, complexity level, and potential risks through advanced natural language processing techniques.

To enhance analytical accuracy and ensure that management decisions are grounded in structured knowledge, a Retrieval-Augmented Generation (RAG) mechanism is integrated into the system. This mechanism enables the agent to retrieve relevant Agile and Scrum concepts and apply them to support sprint planning and project decision-making. As a result, the system generates professional sprint plans and structured management reports aligned with established software development practices.

The system produces constrained and machine-parseable outputs with a fixed structure, ensuring logical consistency between project inputs and generated outputs, and reducing semantic drift and hallucination commonly associated with general-purpose language models. The effectiveness of the proposed agent is evaluated, the clarity and relevance of the generated plans, and the interpretability of AI-driven project management decisions.

This project demonstrates a practical application of modern artificial intelligence techniques—particularly large language models and Retrieval-Augmented Generation—in transforming text-generating models into controlled, explainable, and reliable decision-support systems for software project management. It also discusses current challenges and outlines future directions for developing more accurate and trustworthy AI agents within software engineering environments.

جدول المحتويات

3	الملخص	4
Abstract		
1	الفصل الأول: مقدمة	10
1.1	مقدمة عامة	11
1.2	مقدمة عن المشروع	12
1.3	مشكلة البحث وأهميتها	13
1.4	أهداف المشروع ومحاوره التنفيذية	14
1.4.1	❖ أهداف المشروع (Objectives)	14
1.4.2	محاور التنفيذ (Scope of Work)	14
1.5	مفهوم تعریف الوکیل الذکی (Artificial Intelligence Agent)	15
1.6	الوکیل الذکی فی سیاق النماذج اللغویة الكبیرة:	15
1.7	تعریف وكيل مدير المشروع الذكي المقترن في هذا المشروع	15
1.8	خصائص الوکیل الذکی فی هذا المشروع	16
1.9	موقع هذا الوکیل ضمن أطروحت AI Agents	16
	الفرق بين تولید النص الحر ودعم القرار الذكي في الوکلاء الذکین	16
	تولید النصوص باستخدام النماذج اللغویة (Text Generation)	17
	دعم القرار الذكي باستخدام الوکلاء الذکین (AI Agent-Based Decision Support)	17
	الفارق الجوهری بين النهجین	18

1.10	التحديات المرتبطة بتطوير وكيل مشروع ذكي	18
1.11	التحديات المستقبلية في مجال وكلاء الذكاء	
	الصنيعي لإدارة المشاريع البرمجية	19
1.12	خاتمة الفصل	20
2	الفصل الثاني: الدراسة المرجعية	21
2.1	مقدمة الفصل	22
2.2	تصنيف تقنيات بناء الوكلاء الذكين (AI Agents)	23
2.2.1	الوكلاء المعتمدون على القواعد (Rule-Based Agents)	23
2.2.2	الوكلاء المعتمدون على النماذج والاستدلال (Model-Based and Reasoning Agents)	23
2.2.3	الوكلاء المعتمدون على التعلم (Learning-Based Agents)	24
2.3	النماذج اللغوية الكبيرة في بناء الوكلاء الذكين	24
2.3.1	الوكلاء المعتمدون على النماذج اللغوية الكبيرة (LLM-Based Agents)	24
2.3.2	الوكلاء المعزّزون بالمعرفة (RAG-Enhanced Agents).....	25
2.3.3	الوكلاء القائمون على آليات الانتباه والتخطيط السياقي	25
2.4	مقاييس تقييم أداء الوكيل الذكي وجودة المخرجات	25
2.4.1	دقة الاتساق بين المدخلات والمخرجات (Input–Output Alignment Accuracy)	25
2.4.2	جودة التحليل البنائي للمشروع (Structural Analysis Quality)	26
2.4.3	دقة التخطيط الزمني وإدارة المخاطر	26
2.4.4	موثوقية القرارات المدعومة بالمعرفة (RAG-Grounded Decision Reliability)	26
2.4.5	قابلية التفسير والوضوح الإداري (Explainability & Managerial Readability)	27
2.4.6	التقييم البشري المقارن (Human-in-the-Loop Evaluation)	27
2.5	استعراض الدراسات المرجعية (Related Work)	27
2.5.1	الدراسة المرجعية الأولى	28
2.5.2	الدراسة المرجعية الثانية	28
2.5.3	الدراسة المرجعية الثالثة	29
2.5.4	الدراسة المرجعية الرابعة	29
2.5.5	الدراسة المرجعية الخامسة	29
2.6	خلاصة الاستعراض وتحديد الفجوة البحثية	30

3	الفصل الثالث: المنهجية المعتمدة	31
3.1	مقدمة الفصل	32
3.2	بيئة التطوير والأدوات البرمجية المستخدمة	32
3.2.1	تشغيل النموذج اللغوي الكبير محلياً لغة البرمجة Python	32
3.3	تشغيل النموذج اللغوي الكبير محلياً	33
3.3.1	llama_cpp مكتبة	33
3.3.2	النموذج المستخدم	33
3.4	إدارة الإعدادات باستخدام Config Object	33
3.5	قاعدة المعرفة وإدارة السياق المعرفي	34
3.5.1	قاعدة المعرفة (Agile Knowledge Base)	34
3.6	آلية الاسترجاع المعرفي (RAG)	34
3.6.1	scikit-learn مكتبة	34
3.6.2	TF-IDF منهجية	34
3.6.3	آلية التصفية	35
3.7	هندسة الموجه (Prompt Engineering)	35
3.8	معالجة المخرجات والتحقق البنوي	36
3.8.1	استخراج JSON	36
3.8.2	التحقق من صحة البنية	36
3.9	التقييم الذاتي والإصلاح التلقائي	36
3.9.1	التقييم الذاتي	36
3.9.2	آلية الإصلاح	36
3.10	تتبع الأداء والشفافية	37
3.11	خاتمة الفصل:	37
4	الفصل الرابع: النتائج والخلاصة	38
4.1	مقدمة شاملة	39
4.2	منهجية التقييم	39
4.2.1	معايير التقييم الكمية	39
4.2.2	(Structural Evaluation) التقييم البنوي للمخرجات	41
4.3	نتائج خط الأساس (Baseline Behavior)	41
4.4	النتائج النهائية بعد الإصلاح الذاتي	41
4.4.1	(Scope Management) تحسين إدارة النطاق	41
4.4.2	تحسين تحليل المخاطر	42

4.4.3	توزيع منطقي للمهام عبر السيرناتات	42
4.5	مناقشة النتائج	42
4.5.1	فعالية الوكيل الذكي	42
4.5.2	أهمية عدم استخدام أطر جاهزة	42
4.5.3	حدود النظام الحالية	43
5	الفصل الخامس: الأعمال المستقبلية والتوصيات	44
5.1	مقدمة شاملة	45
5.2	التوصيات المستقبلية	45
5.2.1	(Knowledge Base & Retrieval) أولاً: تطوير قاعدة المعرفة وآلية الاسترجاع	45
5.2.2	(Output Quality & Validation) ثانياً: تحسين جودة المخرجات والتحقق	46
5.2.3	(Self-Eval & Self-Repair) ثالثاً: تطوير آلية التقييم الذاتي والإصلاح الذاتي	46
5.2.4	(Efficiency & Performance) رابعاً: تحسين الكفاءة والسرعة	47
5.2.5	(Broader Applications) خامساً: توسيع نطاق الاستخدام والتطبيقات	47
5.3	الخلاصة	48
6	الفصل السادس: المراجع	49
	References	50

الفصل الأول: مقدمة

1.1 مقدمة عامة

في العصر الرقمي الحديث، أصبحت المشاريع البرمجية جزءاً لا يتجزأ من مختلف جوانب الحياة اليومية والتطبيقات العملية والعلمية، بدءاً من الأنظمة الخدمية والتطبيقات الذكية، مروراً بمنصات الأعمال والتجارة الإلكترونية، وصولاً إلى الأنظمة الحرجية في مجالات مثل الصحة، والمال، والتعليم، والاتصالات. ومع هذا التوسع المتزايد، أصبحت إدارة المشاريع البرمجية بكفاءة ودقة عالياً حاسماً فينجاح هذه الأنظمة، لا سيما في ظل التعقيد المتزايد للمتطلبات وتعدد الأطراف المعنية وتتسارع دورات التطوير.

وعلى الرغم من التقدم الكبير في أدوات تطوير البرمجيات ومنهجيات إدارتها، لا تزال العديد من المشاريع تعاني من مشكلات جوهريّة تتعلق بسوء التخطيط، وضعف تقدير المخاطر، وعدم وضوح المتطلبات، إضافةً إلى القيود الزمنية والموارد المحدودة. ويؤدي ذلك في كثير من الحالات إلى تأخير التسليم، أو انخفاض جودة المنتج النهائي، أو حتى فشل المشروع بالكامل. وتزداد هذه التحديات تقيداً مع ارتفاع حجم المشاريع واعتمادها على تقنيات متقدمة تتطلب تنسيقاً عالياً المستوى بين الجوانب التقنية والإدارية.

من هنا، برزت الحاجة إلى حلول ذكية قادرة على دعم عملية إدارة المشاريع البرمجية بشكل منهجي وفعال. وينبع توظيف تقنيات الذكاء الصنعي في هذا المجال من الاتجاهات البحثية الواحدة، حيث تتيح هذه التقنيات تحليل كميات كبيرة من البيانات غير المهيكلة، وفهم السياق العام للمشاريع، والمساهمة في اتخاذ قرارات أكثر دقة فيما يتعلق بالتخطيط والتنفيذ وتقييم المخاطر. ولا يقتصر هذا الدور على أتمتة المهام التقليدية، بل يمتد ليشمل محاكاة دور مدير المشروع البشري في فهم المشروع وتقديمه إلى مراحل واضحة قبلة للتنفيذ.

وقد شهد هذا المجال تطوراً ملحوظاً مع ظهور النماذج اللغوية الكبيرة وتقنيات التوليد الذكي، التي مكنت من معالجة الأوصاف النصية للمشاريع البرمجية وفهمها دلائلاً. وأسهم ذلك في بناء أنظمة قادرة على تحليل أهداف المشروع، والتقنيات المتوقعة، والمكونات الأساسية، إضافةً إلى اقتراح خطط تنفيذ مبنية على منهجيات معتمدة مثل Agile و Scrum. ومع ذلك، لا تزال هناك تحديات قائمة، أبرزها ضمان اتساق المخرجات مع المدخلات، وتجنب الانحراف الدلالي، وضبط قرارات الذكاء الصنعي بحيث تكون قابلة للتفسير وموثوقة في البيئات العملية.

انطلاقاً من هذه التحديات، يأتي هذا المشروع ليقدم نموذجاً لوكيل ذكاء صنعي واحد يعمل كمدير مشروع برمجي ذكي، بهدف إلى دعم عملية اتخاذ القرار الإداري من خلال تحليل الأوصاف النصية للمشاريع البرمجية، والاستعانة بمعرفة منهجهة منظمة، وتوليد مخرجات مقيدة ومنضبطة تعكس احتياجات المشروع الواقعية وتحدياته الفعلية.

1.2 مقدمة عن المشروع

في ضوء التحديات والفرص المرتبطة بإدارة المشاريع البرمجية الحديثة، يأتي هذا المشروع لاستكشاف وتطوير نموذج ذكي متقدم يعمل كوكيل ذكاء صنعي واحد يؤدي دور مدير مشروع برمجي. يهدف هذا الوكيل إلى دعم عمليات التحليل والتخطيط واتخاذ القرار الإداري، وذلك بالاعتماد على أحدث تقنيات الذكاء الصنعي ومعالجة اللغة الطبيعية. وينصب التركيز الأساسي للمشروع على تحقيق توازن فعال بين دقة التحليل وجودة المخرجات من جهة، وكفاءة التنفيذ وقابلية التطبيق العملي من جهة أخرى، بما يتيح استخدام النظام في بيئات تطوير حقيقة ذات موارد محدودة.

يسند المشروع إلى دراسة تحليلية لمختلف الأساليب المستخدمة في إدارة المشاريع البرمجية، بدءاً من النماذج التقليدية المعتمدة على الخبرة البشرية، وصولاً إلى التوجهات الحديثة التي توظّف النماذج اللغوية الكبيرة في دعم القرار. كما يتم تحليل كيفية توظيف المعرفة المنهجية المرتبطة بمنهجيات Agile و Scrum ضمن إطار ذكي من خلال آلية التوليد المعزّز بالاسترجاع (Retrieval-Augmented Generation – RAG)، بما يضمن أن تكون القرارات المقترنة مستندة إلى ممارسات تطوير برمجيات معتمدة وليس مجرد توليد لغوي عام.

ومن خلال بناء (Pipeline) متكامل لتحليل أوصاف المشاريع النصية ومعالجتها، يقوم الوكيل بتفكيك مدخلات المشروع إلى عناصر رئيسية تشمل أهداف المشروع، والتقنيات المتوقعة، والمكونات الأساسية، ومستوى التعقيد، والمخاطر المحتملة. ويتم تقييم أداء النظام بناءً على مدى الاتساق المنطقي بين مدخلات المشروع ومخرجات التحليل، إضافةً إلى وضوح خطط السبرنت المقترنة وملاءمتها للقيود الزمنية والبشرية المحددة، إلى جانب قابلية تفسير القرارات الإدارية الناتجة.

كما يسعى المشروع إلى معالجة أبرز التحديات المرتبطة باستخدام النماذج اللغوية الكبيرة في الأنظمة الذكية، وعلى رأسها الحد من الانحراف الدلالي وضمان ارتباط المخرجات بسياق المشروع المدخل، إضافةً إلى ضبط بنية الخرج ضمن قالب منظم وقابل للتحقق آلياً. ويسهم ذلك في تعزيز موثوقية النظام وإمكانية الاعتماد عليه كأداة دعم قرار في بيئات تطوير برمجيات حقيقة.

وتكمّن أهمية هذا المشروع في اتساع نطاق تطبيقاته العملية، حيث يمكن توظيفه في فرق تطوير برمجيات صغيرة ومتوسطة، وشركات تقنية ناشئة، ومراكز تطوير تعتمد بمنهجيات Agile ، إضافةً إلى البيئات التعليمية والتدريبية. ومن المتوقع أن تسهم نتائج هذا المشروع في تقديم حلول ذكية وعملية لتحسين كفاءة إدارة المشاريع البرمجية، ودعم تطوير وكلاء ذكاء صنعي أكثر تقدماً وفاعلية في مجال دعم القرار وإدارة المشاريع المعتمدة على الذكاء الصنعي.

1.3 مشكلة البحث وأهميتها

تتمثل المشكلة الجوهرية التي يعالجها هذا المشروع في التحدي المتعلق بضعف كفاءة وموثوقية إدارة المشاريع البرمجية في ظل ازدياد تعقيد المتطلبات، وضيق الموارد الزمنية والبشرية، والاعتماد المتزايد على الأوصاف النصية غير المهيكلة في توثيق المشاريع. ففي العديد من السيناريوهات الواقعية، يتم التعامل مع أوصاف مشاريع غير دقيقة أو غير مكتملة، مما يحدّ بشكل كبير من القدرة على التخطيط السليم، وتقيير المخاطر، واتخاذ قرارات إدارية فعالة في المراحل المبكرة من المشروع.

وتتجلى هذه المشكلة بوضوح في عدد من السياقات العملية، من أبرزها:

- فرق التطوير الصغيرة والمتوسطة: غالباً ما تعتمد هذه الفرق على مستندات مختصرة أو غير رسمية لوصف المشاريع، مما يؤدي إلى سوء فهم الأهداف أو تضخم نطاق العمل، ويزيد من احتمالية التأخير أو الفشل في التسليم.
- الشركات التقنية الناشئة: تعاني المشاريع الناشئة من قيود زمنية ومالية صارمة، ويؤدي غياب التخطيط الإداري المنهجي إلى قرارات غير مدروسة تتعلق باختيار التقنيات أو ترتيب أولويات التطوير.
- البيانات التعليمية والتدريبية: يواجه الطلاب والمبتدئون في هندسة البرمجيات صعوبة في تحويل أفكار المشاريع النصية إلى خطط تنفيذ واضحة ومنهجية، مما يحدّ من قدرتهم على فهم العلاقة بين المتطلبات التقنية والإدارية.
- المشاريع المعتمدة على منهجيات Agile: رغم اعتماد Agile على المرونة، إلا أن غياب التحليل المنهجي للمشروع في مراحله الأولى قد يؤدي إلى انحراف في السيرنات أو ضعف في إدارة المخاطر.

تكمّن أهمية هذا البحث في أنه لا يقتّم مجرد أداة لتوليد نصوص إدارية، بل يطرح حلّاً ذكيّاً عملياً يعتمد على تقنيات الذكاء الصناعي لدعم عملية إدارة المشاريع البرمجية بشكل منهجي. فمن خلال توظيف النماذج اللغوية الكبيرة وآلية التوليد المعزّز بالاسترجاع(RAG)، يتاح النظام تحليل الأوصاف النصية للمشاريع وتحويلها إلى مخرجات منظمة تشمل أهداف المشروع والمكونات الأساسية، ومستوى التعقيد، والمخاطر المحتملة، وخطط التنفيذ المرحلية.

إن القدرة على تحويل أوصاف المشاريع غير المهيكلة إلى تقارير إدارية دقيقة وخطط Agile واضحة تفتح آفاقاً جديدة لتحسين جودة اتخاذ القرار في إدارة المشاريع البرمجية. وبذلك، يسهم هذا المشروع في سد الفجوة بين المعلومات المتاحة في مرحلة توصيف المشروع والاحتياجات المتزايدة للحصول على تخطيط إداري موثوق وقابل للتفسير، مما يعزّز نجاح المشاريع البرمجية في مختلف القطاعات التقنية والتعليمية.

1.4 أهداف المشروع ومحاور التنفيذية

انطلاقاً من المشكلة البحثية المحددة، يهدف هذا المشروع بشكل أساسي إلى تصميم وتطوير نظام ذكي قائم على تقنيات الذكاء الصناعي يعمل كوكيل واحد لإدارة المشاريع البرمجية. ولا يقتصر الهدف على توليد توصيات عامة أو نصوص إدارية، بل يسعى المشروع إلى تقديم نموذج عملٍ وموثوق قادر على تحليل أوصاف المشاريع البرمجية النصية وتحويلها إلى قرارات إدارية وخطط تنفيذ منهجية مبنية على أساس علمية معتمدة. ويتحقق ذلك من خلال مجموعة من الأهداف المحددة كما يلي:

1.4.1 أهداف المشروع (Objectives)

1. تصميم وتطوير وكيل ذكاء صناعي لإدارة المشاريع البرمجية:
بناء وكيل ذكاء صناعي واحد يعتمد على نموذج لغوي كبير قادر على فهم الأوصاف النصية للمشاريع البرمجية وتحليل أهدافها، والتقنيات المتوقعة، والمكونات الأساسية، ومستوى التعقيد والمخاطر، وإنتاج مخرجات منظمة تعكس دور مدير المشروع البشري.
2. دمج المعرفة المنهجية باستخدام RAG :
توظيف آلية التوليد المعزز بالاسترجاع (Retrieval-Augmented Generation) لدمج مفاهيم Agile ضمن عملية التحليل، بما يضمن أن تكون القرارات الإدارية وخطط السبرنت مستندة إلى معرفة منهجية معتمدة وليس ناتجة عن توليد لغوي حر.
3. ضمان الاتساق بين المدخلات والمخرجات:
تطوير آلية إخراج مقيدة ببنية ثابتة قابلة للتحقق آلياً، بهدف الحد من الانحراف الدلالي وضمان أن تكون مخرجات النظام مرتبطة منطقياً بمحنوي المشروع المدخل وقابلة للتفسير والتحليل.
4. تقييم فعالية النظام من منظور ذكاء صنعي:
تقييم أداء الوكيل الذكي بناءً على مدى تطابق الخرج مع الدخل، ووضوح خطط التنفيذ المقترحة، وملاءمة تحليل المخاطر والتعقيد، إضافةً إلى قابلية استخدام النظام كأداة دعم قرار في بيئة تطوير برمجيات حقيقة ذات موارد محدودة.

1.4.2 محاور التنفيذ (Scope of Work)

- لتحقيق الأهداف المذكورة، يتبع المشروع مسار عمل منظم يشمل المحاور التنفيذية التالية:
- إعداد المعرفة وبناء قاعدة البيانات المرجعية:
تجميع وتنظيم مجموعة من مفاهيم Agile و Scrum الأساسية ضمن قاعدة معرفة تُستخدم لدعم عملية الاسترجاع في نظام RAG .
 - تصميم وبناء خط العمل (Pipeline):
بناء خط متكامل يشمل مراحل إدخال وصف المشروع النصي، استرجاع المعرفة ذات الصلة، بناء المطالبة (Prompt Engineering)، تنفيذ التحليل باستخدام النموذج اللغوي، واستخراج المخرجات المنظمة.
 - تنفيذ التجارب وتحليل النتائج:
اختبار النظام على أوصاف مشاريع مختلفة، وتحليل النتائج الناتجة من حيث الاتساق الدلالي، جودة التخطيط، وملاءمة القرارات الإدارية المقترحة لمتطلبات المشروع.
 - التوثيق الأكاديمي والتقييم:
توثيق جميع مراحل المشروع بشكل أكاديمي منهجي، بدءاً من الإطار النظري، مروراً بمنهجية العمل والتنفيذ، وانتهاءً بعرض النتائج ومناقشتها واقتراح آفاق التطوير المستقبلية لوكالء الذكاء الصناعي في مجال إدارة المشاريع البرمجية.

1.5 مفهوم تعريف الوكيل الذكي (Artificial Intelligence Agent)

يُعرف الوكيل الذكي (Artificial Intelligence Agent) في مجال الذكاء الصناعي على أنه كيان برمجي مستقل يمتلك القدرة على إدراك البيئة المحيطة به، واتخاذ قرارات مبنية على هذا الإدراك، ثم تنفيذ أفعال تهدف إلى تحقيق أهداف محددة ضمن قيود معينة. ويتميز الوكيل الذكي بقدراته على العمل بشكل ذاتي، والتكيف مع المدخلات المتغيرة، واستخدام المعرفة والخبرة المتراكمة لتحسين جودة قراراته مع مرور الوقت.

وفي السياق الحديث لأنظمة الذكاء الصناعي، لا يقتصر مفهوم الوكيل الذكي على التفاعل الحسي المباشر مع البيئة الفизيائية، بل يمتد ليشمل التفاعل مع بيانات معلوماتية، مثل النصوص، والبيانات غير المهيكلة، والأنظمة الرقمية المعقدة. وفي هذه الحالات، يكون الإدراك قائماً على تحليل المحتوى الدلالي للمدخلات، بينما تمثل الأفعال في توليد قرارات أو توصيات أو مخرجات منظمة تدعم عمليات بشرية معقدة.

1.6 الوكيل الذكي في سياق النماذج اللغوية الكبيرة:

مع ظهور النماذج اللغوية الكبيرة، تطور مفهوم الوكيل الذكي ليشمل وكلاء قادرين على الفهم الدلالي العميق للنصوص، والاستنتاج، والتخطيط، بدلاً من الاقتصار على ردود فعل بسيطة. إلا أن هذه النماذج، عند استخدامها بشكل منفصل، تفتقر إلى البنية المنهجية والقيود الالزامية لضمان موثوقية القرارات. لذلك، يتم دمجها ضمن إطار وكيل ذكي يحدد أدوارها، ويضبط سلوكها، ويعيد مخرجاتها بما يخدم هدفاً واضحاً.

1.7 تعريف وكيل مدير المشروع الذكي المقترن في هذا المشروع

في ضوء التعريف السابق، يمكن تعريف وكيل مدير المشروع الذكي المقترن في هذا المشروع على النحو التالي:

وكيل ذكاء صناعي مستقل يعمل ضمن بيئة معلوماتية نصية، يهدف إلى دعم عملية إدارة المشاريع البرمجية من خلال إدراك وتحليل الأوصاف النصية للمشاريع، والاستعانة بمعرفة منهجية منظمة، واتخاذ قرارات تخطيطية وإدارية مقيدة بسياق محدد وبنية إخراج منتظمة.

يقوم هذا الوكيل بإدراك البيئة من خلال تحليل المدخلات النصية غير المهيكلة، مثل مستندات المتطلبات أو ملفات README، ويعالجها باستخدام نموذج لغوي كبير مدمج ضمن بنية وكيل ذكي. أما أفعاله فتمثل في توليد مخرجات منظمة تشمل أهداف المشروع، والتقنيات المتوقعة، والمكونات الأساسية، وتقدير مستوى التعقيد والمخاطر، إضافةً إلى اقتراح خطط سبرنت وتقارير إدارية مدعومة بمفاهيم Agile و Scrum.

1.8 خصائص الوكيل الذكي في هذا المشروع

يتميز وكيل مدير المشروع الذكي المقترن بالخصائص التالية:

- الاستقلالية: يعمل الوكيل دون تدخل بشري مباشر بعد تزويده بوصف المشروع.
- الفهم الدلالي: يعتمد على معالجة لغوية متقدمة لفهم السياق والمعنى وليس الكلمات فقط.
- التوجيه المعرفي: يستعين بمعرفة منهجية مسترجعة باستخدام آلية RAG لضبط قراراته.
- ضبط المخرجات: ينتج مخرجات مقيدة بنية ثابتة قبلة للتحقق، مما يحدّ من الانحراف الدلالي.
- دعم القرار: يعمل كمنظومة دعم قرار نكية، وليس كنظام توليد نصوص حر.

1.9 موقع هذا الوكيل ضمن أطروحتـ AI Agents

يُمثل هذا الوكيل نموذجاً تطبيقياً لوكاء الذكاء الصنعي المعتمدين على النماذج اللغوية الكبيرة، حيث يجمع بين الإدراك النصي، والاسترجاع المعرفي، واتخاذ القرار ضمن إطار واحد متكامل. ويضع هذا المشروع الوكيل ضمن فئة Goal-Oriented Intelligent Agents المصممين لدعم مهام إدارية معقدة في بيئات واقعية، مع قابلية التوسيع والتطوير المستقبلي.

الفرق بين توليد النص الحر ودعم القرار الذكي في الوكاء الذكين

من الضروري التمييز بوضوح بين مفهومي توليد النصوص باستخدام النماذج اللغوية وأنظمة دعم القرار المعتمدة على الوكاء الذكين، لا سيما في سياق تطبيقات الذكاء الصنعي الحديثة. فعلى الرغم من أن كلا النهجين يعتمدان على نماذج لغوية كبيرة، إلا أنهما يختلفان جوهرياً من حيث الهدف، وأآلية العمل، وطبيعة المخرجات.

توليد النصوص باستخدام النماذج اللغوية (Text Generation)

- **الهدف:**
يركز هذا النهج على إنتاج نصوص لغوية متماسكة وسليمة نحوياً بناءً على مدخلات عامة، دون الالتزام بالضرورة بسياق تطبيقي محدد أو هدف إداري واضح.
- **الأالية:**
يعتمد على التوليد الاحتمالي للنصوص استناداً إلى الأنماط اللغوية التي تعلّمها النموذج أثناء التدريب، دون وجود قيود صارمة على بنية المخرجات أو علاقتها المباشرة بالمدخل.
- **العمليات:**
تشمل:
 - إعادة صياغة النصوص.
 - الإجابة على الأسئلة العامة.
 - تأثير المحتوى.
 - توليد محتوى وصفي أو تفسيري.
- **النتيجة:**
نص لغوي مقبول من حيث الصياغة، لكنه قد يعاني من انحراف دلالي أو ضعف في الارتباط بالسياق العملي، ولا يمكن الاعتماد عليه مباشرةً لاتخاذ قرارات حساسة أو تنفيذية.

دعم القرار الذكي باستخدام الوكلاء الذكين (AI Agent-Based Decision Support)

- **الهدف:**
يهدف هذا النهج إلى دعم عملية اتخاذ القرار ضمن سياق محدد، من خلال تحليل المدخلات وفهمها دلالياً، ثم تحويلها إلى مخرجات منظمة وقابلة للتنفيذ.
- **الأالية:**
يعمل النموذج اللغوي ضمن إطار وكيل ذكي مقيد، حيث يتم:
 - تحديد هدف واضح للوكيل.
 - تقدير نوع المدخلات.
 - فرض بنية ثابتة للمخرجات.
 - توجيه القرارات بالاعتماد على معرفة منهجية مسترجعة (RAG).
- **العمليات:**
تشمل:
 - تحليل الأوصاف النسبية غير المهيكلة.
 - استنتاج أهداف المشروع ومكوناته.
 - تقدير مستوى التعقيد والمخاطر.
 - اقتراح خطط تنفيذ وسبرنات مبنية على Agile/Scrum.
- **النتيجة:**
مخرجات منظمة تمثل قرارات إدارية مدروسة، مرتبطة بشكل مباشر بالمدخلات، وقابلة للتحقق والتفسير، مما يجعلها مناسبة للاستخدام في بيئات عملية حقيقية.

الفارق الجوهرى بين النهجين

يمكن تلخيص الفرق الجوهرى بين توليد النص الحر ودعم القرار الذكى في أن الأول يركز على تحسين **الشكل اللغوى للنص** دون ضمان الارتباط بالسياق أو الهدف، في حين يسعى الثاني إلى بناء معنى وظيفي قابل للتنفيذ من خلال ضبط السلوك الإدراكي للنموذج ضمن إطار وكيل ذكى.

وفي هذا المشروع، لا يتم استخدام النموذج اللغوى بوصفه مولداً للتصوص فقط، بل يتم توظيفه كعنصر إدراكي داخل وكيل ذكاء صنعي موجه بالمعرفة، يهدف إلى دعم قرارات إدارة المشاريع البرمجية بشكل منهجي وموثوق.

1.10 التحديات المرتبطة بتطوير وكيل مشروع ذكى

على الرغم من التقدم الملحوظ في تقنيات الذكاء الصناعي والنماذج اللغوية الكبيرة، إلا أن تطوير وكيل ذكى قادر على دعم قرارات إدارة المشاريع البرمجية يواجه مجموعة من التحديات الجوهرية التي تتطلب معالجة منهجية دقيقة. وتتمثل أبرز هذه التحديات فيما يلى:

- **عدم اكتمال وغموض المدخلات النصية:**

غالباً ما تكون أوصاف المشاريع البرمجية غير مكتملة أو مكتوبة بشكل غير منظم، مما يؤدي إلى فقدان معلومات سياقية مهمة تتعلق بالأهداف أو القيود أو نطاق العمل. ويشكل استنتاج هذه المعلومات بدقة دون افتراضات غير مبررة أحد أبرز التحديات في تصميم الوكيل الذكى.
- **الانحراف الدلالي في المخرجات:**

تميل النماذج اللغوية الكبيرة، في حال استخدامها دون قيود، إلى توليد مخرجات لغويًا سليمة لكنها غير مرتبطة دلاليًا بسياق المدخل. ويُعد الحد من هذا الانحراف وضمان الاتساق بين وصف المشروع وقرارات التخطيط الناتجة تحدياً أساسياً في أنظمة الوكالء الذكين.
- **ضبط التوازن بين المرونة والصرامة:**

يتطلب وكيل مدير المشروع الذكى درجة من المرونة للتعامل مع مشاريع متعددة ومختلفة الطبيعة، وفي الوقت ذاته يحتاج إلى صرامة منهجية تمنع العشوائية في اتخاذ القرار. إن تحقيق هذا التوازن بين التكيف مع السياق وضبط السلوك الإداري للوكيل يُعد تحدياً تصميمياً معقداً.
- **الاعتماد على المعرفة المنهجية الموثوقة:**

تستوجب عملية التخطيط الإداري الناجح الاستناد إلى مفاهيم ومعايير واضحة من منهجيات Agile و Scrum. ويمكن التحدى في ضمان أن المعرفة المسترجعة عبر آلية RAG دقيقة، محدثة، ومرتبطة فعلياً بسياق المشروع، وليس مجرد استدعاء عام للمفاهيم النظرية.

- الكلفة الحسابية وكفاءة التنفيذ:**

تتطلب النماذج اللغوية الكبيرة موارد حوسية وذاكرة مرتفعة نسبياً، مما قد يحدّ من قابلية استخدام الوكيل في البيئات ذات الموارد المحدودة أو في سيناريوهات تتطلب استجابة سريعة. لذلك، يُعد تحسين كفاءة التنفيذ دون التأثير على جودة التحليل من التحديات المهمة في هذا المشروع.
- قابلية التفسير والثقة في القرارات:**

في السياقات الإدارية، لا يكفي أن تكون القرارات صحيحة فحسب، بل يجب أن تكون قابلة للتفسير ومفهومة للمستخدم البشري. ويُعد تقييم مخرجات يمكن تتبع منطقها وربطها بالمدخلات والمعرفة المستخدمة تحدياً أساسياً لتعزيز الثقة في الوكيل الذكي، خاصة في البيئات التعليمية أو المؤسسية.
- التعامل مع السيناريوهات الواقعية المتنوعة:**

تركز العديد من النماذج التجريبية على حالات مثالية أو أوصاف مشاريع مصاغة بعناية، في حين أن المشاريع الواقعية تتسم بتعقيد أعلى وتنوع في الصياغة والمحتوى. وبظل تعليم أداء الوكيل على هذه السيناريوهات الواقعية أحد التحديات المفتوحة التي يسعى هذا المشروع إلى معالجتها تدريجياً.

1.11 التحديات المستقبلية في مجال وكلاء الذكاء الصنعي لإدارة المشاريع البرمجية

مع استمرار تطور تقنيات الذكاء الصنعي، ولا سيما النماذج اللغوية الكبيرة وأنظمة الوكالء الذكين، تبرز تحديات مستقبلية جديدة تتطلب إعادة تقييم مستمرة لأولويات البحث والتطوير، وذلك بهدف تلبية متطلبات التطبيقات الواقعية والمعقدة في مجال إدارة المشاريع البرمجية. وتفرض هذه المتطلبات على الباحثين والمطوروين التفكير في معالجة عدد من التحديات الجوهرية، من أبرزها:

- نحو وكلاء يتعاملون مع الغموض وعدم الاتكمال:** (Blind Project Understanding)

يتوجب تطوير وكلاء ذكين قادرين على تحليل أوصاف مشاريع واقعية غير مكتملة أو غير دقيقة، دون توفر معرفة مسبقة واضحة حول جميع المتطلبات أو القيود. ويعُد هذا التوجه خطوة أساسية نحو بناء وكلاء ذكاء صنعي يمكن الاعتماد عليهم في البيئات العملية الحقيقة، حيث نادرًا ما تكون مدخلات المشاريع مكتملة أو مصاغة بشكل مثالي.
- تحسين كفاءة الوكالء وتقليل الكلفة الحسابية:**

مع ارتفاع تعقيد النماذج اللغوية المستخدمة ضمن الوكالء الذكين، تبرز الحاجة إلى تصميم وكلاء أخف وزناً وأكثر كفاءة من حيث استهلاك الموارد الحوسية والذاكرة. ويهدف هذا التوجه إلى تمكين استخدام وكلاء إدارة المشاريع الذكية في بيئات ذات موارد محدودة، أو ضمن أنظمة تتطلب استجابة شبه فورية، مثل أدوات إدارة الفرق البرمجية التفاعلية.
- تعزيز موثوقية القرارات وتقليل التوليد غير الواقعي:**

يتطلب الاستخدام العملي لوكالء الذكاء الصنعي مستوى عالياً من الثقة في القرارات الإدارية الناتجة. لذلك، يُعد الحد من التوصيات غير الواقعية أو غير القابلة للتطبيق، وتعزيز ارتباط القرارات بسياق المشروع الفعلي، من التحديات المستقبلية الأساسية، خاصة في المشاريع الحساسة التي تعتمد على تخطيط دقيق للموارد والوقت.
- تطوير قابلية التفسير والشفافية:**

مع توسيع استخدام الوكالء الذكين في البيئات المؤسسية، تزداد الحاجة إلى وكلاء قادرين على تفسير قراراتهم بشكل واضح للمستخدم البشري. ويشمل ذلك توضيح أسباب اختيار تقنيات معينة، أو تقسيم السيرنات، أو تقدير مستوى المخاطر، بما يعزز الثقة في النظام ويدعم اتخاذ القرار المشترك بين الإنسان والذكاء الصنعي.
- تكامل الوكالء مع مهام إدارية وتقنية أخرى:**

يتجه البحث الحديث نحو تصميم وكلاء ذكين شاملة لا تقتصر على تخطيط المشاريع فقط، بل تدمج مهام إضافية

ضمن إطار واحد متكامل، مثل تتبع تقدم المشروع، تحليل الأداء، إدارة المخاطر الديناميكية، ودعم اتخاذ القرار أثناء التنفيذ. ويسمم هذا التكامل في بناء منظومات ذكية أكثر شمولية وفاعلية في دعم دورة حياة المشروع البرمجي بالكامل.

1.12 خاتمة الفصل

يختتم هذا الفصل بتقديم الإطار النظري والمنهجي الذي يؤسس لهذا البحث في مجال الذكاء الصنعي، حيث تم استهلال الفصل بتحديد سياق المشكلة المرتبطة بإدارة المشاريع البرمجية الحديثة، مع إبراز الدور المحوري للتخطيط المنهجي واتخاذ القرار الإداري في نجاح المشاريع التقنية، ولا سيما في ظل تزايد التعقيد، وضغط الوقت، وتعدد المتطلبات.

وعلى هذا الأساس، تم بناء إشكالية البحث المتمثلة في تحليل أوصف المشاريع البرمجية النصية غير المهيكلة، وما يترتب على ذلك من ضعف في وضوح الأهداف، وسوء تقدير المخاطر، وعدم دقة خطط التنفيذ. وقد تم توضيح الأثر السلبي لهذه الإشكالية في بيئة تطوير برمجيات واقعية، سواء في الفرق الصغيرة، أو الشركات الناشئة، أو السيارات التعليمية. ومن هذه الإشكالية انبثقت أهداف المشروع ومحاربه التقنية، التي تم تحديدها بدقة لتوجيه مسار البحث نحو تطوير وكيل ذكاء صنعي قادر على دعم القرار الإداري بشكل منهجي وقابل للتفسير.

ولتحقيق الوضوح الاصطلاحي والمفاهيمي، قدم الفصل تعريفاً دقيقاً لمفهوم الوكيل الذكي (AI Agent) ضمن سياق النماذج اللغوية الكبيرة، مع توضيح موقع وكيل مدير المشروع الذي يوصفه منظومة دعم قرار، وليس مجرد نظام توليد نصوص. كما تم التمييز بشكل منهجي بين التوليد اللغوي الحر وأنظمة الوكالء الذكين المقيدة بالسياق والمعرفة، وذلك لتحديد نطاق البحث وحدوده بشكل واضح.

كما ألقى الفصل نظرة نقية على التحديات الراهنة والمستقبلية المرتبطة بتطوير وكلاء ذكاء صنعي لإدارة المشاريع البرمجية، بدءاً من مشكلات الغموض وعدم اكتمال المدخلات، مروراً بالانحراف الدلالي وضبط المخرجات، وصولاً إلى القيود الحسابية وقابلية التفسير، إضافة إلى التوجهات البحثية المستقبلية نحو وكلاء أكثر استقلالية، وكفاءة، وقدرة على التكامل مع مهام إدارية وتقنية أخرى.

وبهذا العرض المنهجي، يكون الفصل الأول قد أنجز وظيفته في بناء قاعدة معرفية متينة وإطار مفاهيمي واضح، يوفر الأساس المنطقي الذي ستبني عليه الفصول اللاحقة. حيث سينتقل البحث من مرحلة التنظير والتحليل المفاهيمي إلى دراسة الأعمال السابقة ذات الصلة، ومن ثم إلى عرض تفاصيل النظام المقترن، ومنهجية التنفيذ، والتقييم العملي لأداء وكيل الذكاء الصنعي المطرّر في هذا المشروع.

2 الفصل الثاني: الدراسة المرجعية

2.1 مقدمة الفصل

يتناول هذا الفصل استعراضًا منهجيًّا وتحليليًّا للأبحاث والدراسات السابقة التي أسهمت في تشكيل مجال الوكلاء الذكين (AI Agents) وتطبيقاتهم في هندسة البرمجيات وإدارة المشاريع البرمجية. وسيتم تتبع المسار التطوري لهذا الحقل، بدءًا من النماذج التقليدية للوكلاء المعتمدين على القواعد والأنظمة التفاعلية البسيطة، وصولًا إلى التركيز بشكل خاص ومفصل على التحول الجذري الذي أحدثه تقنيات الذكاء الصنعي الحديثة، ولا سيما النماذج اللغوية الكبيرة والتعلم العميق، والتي أعادت تعريف حدود الممكن في تصميم أنظمة دعم القرار الذكية.

ولا تمثل هذه الدراسات مجرد عرضٍ تاريخيٍ لتطور المفاهيم، بل تشكل قاعدة معرفية أساسية لا غنى عنها لفهم الاتجاهات البحثية الحديثة، والنماذج الرائدة التي تم ابتكارها بهدف تمكين الوكلاء الذكين من الفهم الدلالي للنصوص، والتخطيط، واتخاذ القرار في بيئات برمجية معقدة. كما توفر هذه الأعمال الخلفية النظرية التي يقوم عليها هذا المشروع، وتسهم في توضيح موقعه ضمن الأدبيات العلمية ذات الصلة.

في هذا السياق، سيتم استعراض المناهج المتتبعة في الأدبيات العلمية بشكل نقي، مع تحليل عميق للبني والآليات الرئيسية التي هيمنت على أبحاث الوكلاء الذكين، بما في ذلك الوكلاء المعرفيون (Cognitive Agents) القادرون على الاستدلال واتخاذ القرار، والوكلاء المعتمدون على النماذج اللغوية الكبيرة الذين يتمتعون بقدرات متقدمة في فهم اللغة الطبيعية، إضافةً إلى تقنيات التوليد المعزز بالاسترجاع (RAG) التي مكّنت هذه الوكلاء من الاستناد إلى معرفة خارجية منظمة أثناء عملية التخطيط واتخاذ القرار.

وسيقتربن هذا التحليل بمناقشة التحديات التقنية والمنهجية التي واجهتها الدراسات السابقة، مثل مشكلة الانحراف الدلالي في مخرجات النماذج اللغوية، وصعوبة ضبط القرارات ضمن سياق تطبيقي محدد، إضافةً إلى القيود المرتبطة بالتكلفة الحسابية وقابلية التفسير في الأنظمة الذكية المعقدة.

علاوةً على ذلك، ونظرًا لأن تقديم الأداء يُعد حجر الزاوية في البحث العلمي، سيتم التركيز على أساليب تقييم أنظمة الوكلاء الذكين في الأدبيات السابقة. ويشمل ذلك مناقشة المعايير المستخدمة لقياس جودة القرارات المولدة، ومدى اتساقها مع المدخلات، وقابليتها للتفسير، إضافةً إلى المقارنات بين أداء الأنظمة الذكية والقرارات البشرية في سياقات إدارة المشاريع البرمجية.

وفي نهاية المطاف، يهدف هذا الفصل إلى تقديم تحليل نقي ومتكملاً لا يقتصر على سرد الدراسات السابقة، بل يسعى إلى استخلاص الدروس المستفادة منها، ورصد الفجوات البحثية التي لا تزال قائمة في مجال الوكلاء الذكين لإدارة المشاريع البرمجية. ومن خلال ذلك، يتم بناء مبرر علمي متين للمنهجية المقترحة في هذا البحث، بما يدعم تحقيق أهدافه ويسهم بفعالية في تطوير أنظمة ذكاء صنعي أكثر موثوقية، وقابلية للتفسير، وملاءمة للتطبيق في البيئات البرمجية الواقعية.

2.2 تصنیف تقنيات بناء الوکلاء الذکيين (AI Agents)

شهدت منهجيات تصميم وبناء الوکلاء الذکيين تطويراً ملحوظاً عبر مراحل تاريخية متعددة، ويمكن تصنیف هذه المنهجيات، من منظور تقني ومنهجي، إلى عدة فئات رئيسية، لكل منها مبادئها الخاصة، ومزاياها، وقيودها. ويُعد فهم هذا التصنیف ضروريًا لاستيعاب التطور الذي قاد إلى بروز الوکلاء الذکيين المعتمدين على النماذج اللغوية الكبيرة بوصفهم النهج الأكثر فاعلية في الوقت الحالي.

2.2.1 الوکلاء المعتمدون على القواعد (Rule-Based Agents)

تُعد هذه الفئة من أقدم وأبسط أنواع الوکلاء الذکيين، حيث تعتمد على مجموعة من القواعد المنطقية المحددة مسبقاً لاتخاذ القرار. يعمل الوکيل في هذا السياق من خلال مطابقة المدخلات مع شروط محددة، ثم تنفيذ أفعال ثابتة بناءً على هذه الشروط.

وعلى الرغم من بساطة هذه المقاربة وسرعة تنفيذها، إلا أنها تعاني من قيود جوهريّة، أبرزها ضعف المرونة وعدم القدرة على التكيف مع السيناريوهات غير المتوقعة أو المدخلات غير المهيكلة. كما أن هذا النوع من الوکلاء غير قادر على التعامل مع الغموض أو استنتاج معلومات غير مصرح بها صراحة، مما يجعله غير مناسب لإدارة المشاريع البرمجية المعقدة.

2.2.2 الوکلاء المعتمدون على النماذج والاستدلال (Model-Based and Reasoning Agents)

تهدف هذه الفئة إلى تجاوز محدودية الوکلاء القائمين على القواعد، من خلال بناء تمثيل داخلي للحالة أو البيئة التي يعمل فيها الوکيل. يعتمد هذا النوع من الوکلاء على نماذج استدلالية ومنهجيات تخطيط تتيح له تحليل الوضع الحالي، وتقدير النتائج المحتملة للأفعال المختلفة، ثم اختيار القرار الأنسب.

تتميز هذه المقاربة بقدرتها على التعامل مع سيناريوهات أكثر تعقيداً مقارنة بالأنظمة القائمة على القواعد، إلا أنها تتطلب نماذج دقيقة ومحدة للبيئة، وتعتمد بشكل كبير على جودة الافتراضات المسبقة. كما أن توسعها لتشمل بيانات غنية بالنصوص غير المهيكلة، مثل أوصاف المشاريع البرمجية، يُعد أمراً محدود الفاعلية.

2.2.3 الوكالء المعتمدون على التعلم (Learning-Based Agents)

تمثل هذه الفئة النقلة النوعية في مجال الوكالء الذكين، وهي النهج الأكثر انتشاراً وفاعلية في الوقت الراهن. بدلاً من الاعتماد على قواعد أو نماذج ثابتة، تعتمد هذه الوكالء على تعلم السلوك واتخاذ القرار من البيانات، باستخدام تقنيات التعلم الآلي والتعلم العميق.

يمكن تقسيم هذا النهج تاريخياً إلى:

- **الوكالء المعتمدون على التعلم الإحصائي:**
الذين يعتمدون على نماذج احتمالية وخوارزميات تعلم تقليدية لاستخلاص الأنماط من البيانات.
- **الوكالء المعتمدون على التعلم العميق:**
الذين يستخدمون الشبكات العصبية العميقه لاكتساب تمثيلات عالية المستوى للبيئة، مما مكّنهم من التعامل مع بيانات معقدة وغير مهيكلة مثل النصوص.

وتعُد هذه الفئة الأساس الذي بُنيت عليه الوكالء الذكين المعتمدون على النماذج اللغوية الكبيرة، والتي تشَكّل محور هذا البحث.

2.3 النماذج اللغوية الكبيرة في بناء الوكالء الذكين

أحدثت النماذج اللغوية الكبيرة ثورة حقيقة في تصميم الوكالء الذكين، حيث تجاوزت قدرتها حدود التفاعل النصي البسيط إلى الفهم الدلالي، والاستدلال، والتخطيط. تعتمد هذه النماذج على تعلم تمثيلات لغوية عميقه من كميات ضخمة من البيانات النصية، مما يتيح لها تحليل السياق واستخلاص المعاني الضمنية.

2.3.1 الوكالء المعتمدون على النماذج اللغوية الكبيرة (LLM-Based Agents)

تُعد النماذج اللغوية الكبيرة المكون الإدراكي الأساسي في هذا النوع من الوكالء، حيث تمكّن الوكيل من فهم أوصاف معقدة مكتوبة بلغة طبيعية، مثل وثائق المتطلبات أو ملفات README. وتسمح هذه القدرة بتحويل المدخلات النصية غير المهيكلة إلى تمثيلات معرفية يمكن استخدامها في التخطيط واتخاذ القرار.

إلا أن استخدام هذه النماذج بشكل مباشر قد يؤدي إلى مخرجات غير منضبطة أو انحراف دلالي، مما يستوجب دمجها ضمن إطار وكيل ذكي مقيد يحدد الأهداف، وينظم عملية اتخاذ القرار.

2.3.2 الوكالء المعزّز بالمعرفة (RAG-Enhanced Agents)

يمثل دمج آلية التوليد المعزّز بالاسترجاع (Retrieval-Augmented Generation – RAG) خطوة محورية في تطوير الوكالء الذكيين. تتيح هذه الآلية لوكيل الوصول إلى معرفة خارجية منظمة، مثل مفاهيم Agile وScrum، واستخدامها لدعم قراراته بدلاً من الاعتماد على المعرفة الضمنية للنموذج فقط.

يسهم هذا الدمج في تقليل التوليد غير الواقعي، وتعزيز موثوقية المخرجات، وتحويل الوكيل من نظام لغوي عام إلى منظومة دعم قرار معرفية موجة.

2.3.3 الوكالء القائمون على آليات الانتباه والتخطيط السياقي

تستفيد الوكالء الحديثة من آليات الانتباه المتقدمة داخل النماذج اللغوية، والتي تمكّنها من التركيز على الأجزاء الأكثر أهمية في المدخلات النصية، مثل القيود الزمنية أو المتطلبات الحرجة للمشروع. وتساعد هذه الآليات في التقاط العلاقات بعيدة المدى بين عناصر المشروع المختلفة، مما يعزّز جودة التخطيط واستنتاج المخاطر.

2.4 مقاييس تقييم أداء الوكيل الذكي وجودة المخرجات

يُعد تقييم أداء الوكالء الذكيين (AI Agents) عنصراً جوهرياً في الأبحاث المعاصرة في مجال الذكاء الاصطناعي، خاصة عند اعتمادهم على نماذج لغوية كبيرة في تحليل المدخلات غير المهيكلة وتوليد قرارات أو تقارير داعمة لصنع القرار. وعلى عكس الأنظمة التقليدية، لا يقتصر تقييم الوكيل الذكي على صحة التنفيذ البرمجي، بل يمتد ليشمل جودة الفهم، ودقة التحليل، واتساق المخرجات مع المدخلات، ومدى توافقها مع المعرفة المرجعية المعتمدة.

في هذا المشروع، تم اعتماد مجموعة من المقاييس الكمية والنوعية لتقييم أداء وكيل إدارة المشاريع الذكي، وذلك بهدف قياس مدى موثوقية المخرجات، ومستوى التحكم في السلوك التوليدى للنموذج، وضمان عدم انحرافه عن سياق المهمة المحددة.

2.4.1 دقة الاتساق بين المدخلات والمخرجات (Input–Output Alignment Accuracy)

يُعد اتساق المخرجات مع المدخلات النصية أحد أهم معايير تقييم الوكالء المعتمدين على النماذج اللغوية الكبيرة. يقيس هذا المعيار مدى التزام الوكيل بالمحظى الفعلي لوصف المشروع البرمجي المدخل، وعدم توليد استجابات عامة أو غير ذات صلة بالسياق.

يتم تقييم هذا المؤشر من خلال:

- مقارنة عناصر المشروع المستخرجة (مثل الهدف، التقنيات، المكونات، المخاطر) مع ما ورد صراحة أو ضمنياً في وصف المشروع.
- قياس نسبة العناصر الصحيحة مقابل العناصر المختلفة أو غير المدعومة بالنص الأصلي.

يعكس هذا المقياس قدرة الوكيل على الفهم الدلالي الحقيقي، ويُعد مؤشراً مباشراً على تقليل ظاهرة الهلوسة في المخرجات (Hallucination).

2.4.2 جودة التحليل البنائي للمشروع (Structural Analysis Quality)

يركز هذا المعيار على تقييم قدرة الوكيل على تحويل النصوص غير المهيكلة إلى بنية منطقية قابلة للتنفيذ، وهو ما يُعد جوهر دور وكيل إدارة المشاريع.

يشمل هذا التقييم:

- وضوح تقسيم المشروع إلى مكونات رئيسية.
- منطقية توزيع المهام على مراحل أو Sprints.
- الترابط بين الأهداف، المخاطر، والخطوة الزمنية المقترنة.

كلما كانت البنية الناتجة أكثر تنظيماً وتماسكاً، دل ذلك على نضج آلية الاستدلال والتخطيط داخل الوكيل.

2.4.3 دقة التخطيط الزمني وإدارة المخاطر

نظراً لأن الوكيل يحاكي دور مدير مشروع بشري، فإن جودة التخطيط تعد معياراً أساسياً للتقييم. يركز هذا المقياس على:

- مدى واقعية تقسيم العمل عبر السبرنات.
- শمولية تحديد المخاطر التقنية والتنظيمية.
- منطقية استراتيجيات التخفيف (Mitigation Strategies) المقترنة.

يتم تقييم هذا الجانب من خلال المقارنة مع مبادئ Agile و Scrum المعتمدة، سواء بشكل بدوبي أو بالرجوع إلى قاعدة المعرفة المستخدمة ضمن آلية RAG.

2.4.4 موثوقية القرارات المدعومة بالمعرفة (RAG-Grounded Decision Reliability)

يُعد دمج آلية التوليد المعزّز بالاسترجاع (RAG) أحد المحاور الجوهرية في هذا المشروع، ولذلك تم تخصيص مقياس مستقل لتقييم أثرها.

يركز هذا المقياس على:

- مدى اعتماد المخرجات على المعرفة المسترجعة بدلاً من التوليد الحر.
- عدد المفاهيم الإدارية المستندة إلى مصادر Agile/Scrum الموثوقة.
- انخفاض التناقضات أو القرارات غير المبررة في التقارير الناتجة.

يعكس هذا المؤشر مدى تحول الوكيل من نموذج لغوي عام إلى نظام دعم قرار معرفي.

2.4.5 قابلية التفسير والوضوح الإداري(Explainability & Managerial Readability)

نظرًا لأن المخرجات موجهة للاستخدام البشري، فإن قابلية الفهم والتفسير تعد عنصرًا أساسياً في التقييم. يتم التركيز هنا على:

- وضوح اللغة الإدارية المستخدمة في التقارير.
- تسلسل الأفكار والمنطق الإداري في عرض النتائج.
- سهولة تتبع سبب كل قرار أو توصية.

هذا المعيار يضمن أن النظام لا يعمل كـ“صندوق أسود”， بل كوكيل ذكي قابل للفهم والمراجعة.

2.4.6 التقييم البشري المقارن(Human-in-the-Loop Evaluation)

في المراحل المتقدمة من المشروع، يُخطط لإدراج تقييم بشري مقارن، حيث تتم مقارنة مخرجات الوكيل مع:

- خطط مشاريع أعدتها مدير أو مشاريع بشريون.
- أو مخرجات أدوات تخطيط تقليدية.

يساعد هذا النوع من التقييم في قياس مدى قرب أداء الوكيل من التفكير الإداري البشري، ويعزز مصداقية النتائج البحثية.

2.5 استعراض الدراسات المرجعية(Related Work)

بعد عرض الإطار النظري والمفاهيم الأساسية المرتبطة بالوكلاء الذكين والنمذج اللغوية الكبيرة، يركّز هذا القسم على تحليل مجموعة من الدراسات المرجعية التي شكلت نقاط تحول مهمة في توظيف الذكاء الاصطناعي، ولا سيما النماذج اللغوية الكبيرة (LLMs) والأنظمة المعتمدة على الوكلاء(Agentic Systems)، ضمن مجال هندسة البرمجيات وإدارة المشاريع.

تم اختيار هذه الدراسات بعناية لتغطية محاور مكملة لبعضها، تشمل:
أمثلة تخطيط المشاريع البرمجية، تحليل المستودعات البرمجية، دعم منهجيات Agile وScrum، دمج آليات الاسترجاع المعرفي(RAG)، وتقييم موثوقية المخرجات الناتجة عن النماذج اللغوية. وبهدف هذا الاستعراض إلى استخلاص الإسهامات العلمية الرئيسية، وتحديد القيود القائمة، وربطها مباشرة بالمنهجية المقترحة في هذا البحث.

2.5.1 الدراسة المرجعية الأولى

“Large Language Models for Software Engineering: A Systematic Literature Review” (2023)

ملخص الدراسة:

تُعد هذه الدراسة واحدة من أوائل المراجعات المنهجية الواسعة التي تناولت استخدام النماذج اللغوية الكبيرة في هندسة البرمجيات. قامت بتحليل أكثر من 250 دراسة منشورة بين عامي 2020 و2023، وغطّت تطبيقات LLMs في تحليل المتطلبات، توليد الشيفرة، التوثيق، الاختبار، ودعم مهام إدارة المشاريع.

خلصت الدراسة إلى أن النماذج اللغوية الكبيرة تسهم بشكل واضح في رفع الإنتاجية وتقليل الجهد الإداري، إلا أنها تعاني من محدودية الدقة عند التعامل مع معرفة مؤسسية مغلقة أو سياقات خاصة، مما يجعل دمجها مع قواعد معرفة محلية أو تقنيات RAG أمراً ضرورياً لضمان موثوقية النتائج.

أهميةها بالنسبة لهذا المشروع:

تدعم هذه الدراسة الفرضية الأساسية للمشروع، والمتمثلة في إمكانية استخدام LLMs كمساعد ذكي في إدارة المشاريع، مع التأكيد على الحاجة إلى آلية استرجاع معرفي للحد من الانحراف الدلالي والاعتماد الزائد على التوليد الحر.

2.5.2 الدراسة المرجعية الثانية

“Automating Project Planning Using Large Language Models” (2023)

ملخص الدراسة:

تناولت هذه الورقة تطوير نظام يعتمد على النماذج اللغوية الكبيرة لأتمتة عملية تخطيط المشاريع البرمجية انطلاقاً من وثائق المشروع النصية مثل README وملفات المتطلبات. استخدم الباحثون تقنيات هندسة الأوامر (Prompt Engineering) لتحليل المدخلات وتوليد خطط تنفيذ زمنية تتضمن مراحل العمل، المخاطر، والموارد.

أظهرت النتائج أن الخطط المولدة كانت متماسكة ومنطقية، وحققت نسبة تطابق عالية مع الخطط التي أعدها مدير ومشاريع بشريون، مع تقليل زمن إعداد الخطة بشكل ملحوظ.

أهميةها بالنسبة لهذا المشروع:

تمثل هذه الدراسة الأساس المباشر لوظيفة الوكيل المقترن، حيث توكل إمكانية تحويل الوصف النصي للمشروع إلى خطة تنفيذ منظمة. كما تبرز الحاجة إلى تقييم دقيق لمدى تطابق المخرجات مع المدخلات، وهو ما يعالجه هذا البحث بشكل صريح.

2.5.3 الدراسة المرجعية الثالثة

“RepoAgent: LLM-Based Multi-Agent System for GitHub Project Analysis” (2024)

ملخص الدراسة:

تقدم هذه الدراسة نظاماً قائماً على وكلاء متعددين لتحليل مستودعات GitHub ، حيث يتولى كل وكيل مهمة محددة مثل تلخيص التغييرات، تتبع النشاط، أو توليد تقارير دورية. يعتمد النظام على LLMs لفهم محتوى المستودعات وتحويله إلى رؤى تحليلية قابلة للاستخدام.

أهميةها بالنسبة لهذا المشروع:

على الرغم من اعتماد هذا البحث على وكيل واحد (Single Agent) بدلاً من بنية متعددة الوكلاء، إلا أن هذه الدراسة توفر مرجعًا مهمًا فيما يتعلق بتقسيم المهام منطقيًا داخل النظام، وأهمية التقارير المنظمة، وإمكانية التوسيع مستقبلًا نحو معماريات أكثر تعقيدًا.

2.5.4 الدراسة المرجعية الرابعة

“Cognitive Agents for Agile Project Management” (2024)

ملخص الدراسة:

تستكشف هذه الدراسة دور الوكلاء المعرفيين المدعومين بالنمذج اللغوية الكبيرة في تحسين إدارة المشاريع وفق منهجيات Scrum و Agile. طور الباحثون نظاماً أولياً قادرًا على إدارة المجتمعات، تتبع تقدم المهام، وتوليد تقارير السبرنت، مع تكامل مباشر مع أدوات مثل Jira.

أظهرت النتائج تحسناً ملحوظاً في كفاءة فرق العمل وتقليل الزمن المستغرق في الأنشطة الإدارية المتكررة.

أهميةها بالنسبة لهذا المشروع:

تدعم هذه الدراسة رؤية المشروع في التعامل مع الوكيل الذكي ككيان إداري مساعد، وليس مجرد أداة نصية. كما تفتح المجال لتطوير نسخ مستقبلية من النظام تتكامل مع أدوات إدارة المشاريع الفعلية.

2.5.5 الدراسة المرجعية الخامسة

“RAG-Enhanced Language Models for Knowledge-Rich Task Planning” (2024)

ملخص الدراسة:

تركّز هذه الورقة على دمج تقنية التوليد المعزّز بالاسترجاع (RAG) مع النماذج اللغوية الكبيرة في مهام التخطيط المعقّدة التي تتطلب معرفة خارجية. أثبتت النتائج أن استخدام RAG يؤدي إلى تحسين دقة القرارات وتقليل الأخطاء السياقية بشكل واضح.

أهميةها بالنسبة لهذا المشروع:

تُعد هذه الدراسة الداعم العلمي الأقوى لمرحلة RAG في المشروع المقترن، حيث تبرر استخدام قاعدة معرفة Agile/Scrum لتحويل الوكيل من نموذج لغوي عام إلى نظام دعم قرار معرفي موثوق.

2.6 خلاصة الاستعراض وتحديد الفجوة البحثية

من خلال تحليل الدراسات السابقة، يتضح أن النماذج اللغوية الكبيرة تمتلك قدرة عالية على فهم النصوص البرمجية وتوليد خطط وتقديرات مفيدة، إلا أنها تعاني من تحديات تتعلق بالموثوقية، والانحراف الدلالي، وضعف الارتباط الصريح بالمدخلات في بعض الحالات. كما تشير الأدبيات بوضوح إلى أن دمج آليات الاسترجاع المعرفي يمثل الحل الأكثر فاعلية لمعالجة هذه التحديات.

بناءً على ذلك، يتموضع هذا المشروع بوصفه محاولة منهجية لتطوير وكيل ذكاء صنعي واحد لإدارة المشاريع البرمجية، يرتكز على:

- تحقيق تطابق صارم بين المدخلات والمخرجات،
- توليد مخرجات منظمة وقابلة للتقسيم،
- دعم القرارات الإدارية بالمعرفة المسترجعة عبر RAG ،
- وإتاحة مسار واضح للتوسيع المستقبلي من حيث الواجهة، التكامل، ومعايير التقييم.

ويتمثل هذا التوجه مساهمة علمية وتطبيقية ضمن مجال الوكالء الذكين المعتمدين على النماذج اللغوية الكبيرة في بيئات هندسة البرمجيات.

3 الفصل الثالث: المنهجية المعتمدة

3.1 مقدمة الفصل

بعد استعراض الخلفية النظرية المتعلقة بالوكلاء الأذكياء (Artificial Intelligence Agents)، والنماذج اللغوية الكبيرة (Large Language Models)، ومنهجيات إدارة المشاريع الرشيقه (Agile & Scrum) في الفصول السابقة، يرکز هذا الفصل على عرض المنهجية العملية والتقنية للنظام المقترن.

يتمثل النظام في وكيل ذكي ذاتي (**Autonomous AI Agent**) قادر على تحليل وصف مشروع برمجي نصي غير منظم، ثم تحويله إلى خطة تنفيذية منظمة وقابلة للتطبيق، تشمل تحديد هدف المشروع، التقنيات المتوقعة، المكونات الأساسية، مستوى التعقيد، المخاطر المحتملة، وتقسيم العمل إلى سيرنات وفق مبادئ Scrum.

يهدف هذا الفصل إلى توضيح:

- الأدوات البرمجية المستخدمة
- كيفية تكاملها ضمن بنية واحدة
- منطق اتخاذ القرار داخل الوكيل
- آليات الضبط والتحقق والتقييم الذاتي

وذلك بما يضمن الشفافية، القابلية للتكرار، وإمكانية الدفاع الأكاديمي عن كل قرار تقني تم اتخاذه.

3.2 بيئة التطوير والأدوات البرمجية المستخدمة

تم تطوير النظام باستخدام لغة **Python** نظرًا لمرونتها العالية، ودعمها الواسع لأدوات الذكاء الاصطناعي، إضافة إلى سهولة دمج النماذج اللغوية، وخوارزميات المعالجة النصية، وآليات التحقق المنطقي.

3.2.1 لغة البرمجة Python

تم اعتماد Python كلغة رئيسية لتنفيذ النظام، لما توفره من:

- دعم قوي للبرمجة الكائنية والهيكلية
- مكتبات جاهزة لمعالجة اللغة الطبيعية
- سهولة التعامل مع البيانات المهيكلة (JSON)
- توافق ممتاز مع مكتبات تشغيل النماذج اللغوية محلياً

3.3 تشغيل النموذج اللغوي الكبير محليًا

llama_cpp مكتبة 3.3.1

تم استخدام مكتبة `llama_cpp` لتشغيل النموذج اللغوي الكبير محليًا دون الحاجة إلى اتصال خارجي أو واجهات سحابية. يحقق هذا الاختيار عدة أهداف بحثية مهمة:

- التحكم الكامل في عملية التوليد
- ضمان خصوصية البيانات
- تقليل الاعتماد على خدمات خارجية
- إمكانية تحليل سلوك النموذج بدقة

3.3.2 النموذج المستخدم

تم اعتماد النموذج **Qwen2.5-7B-Instruct** (Instruction-tuned) بصيغة GGUF، وهو نموذج موجّه للتعليمات (Instruction-tuned)، مما يجعله مناسًّا لأداء مهام التخطيط واتخاذ القرار بدل التوليد الحر.

تم ضبط النموذج باستخدام المعاملات التالية:

- `n_ctx = 4096` لتحديد طول السياق
- `temperature = 0.2` لتقليل العشوائية
- `top_p = 0.85` لتحقيق توازن بين الدقة والتنوع
- `(Reproducibility seed = 42)` لضمان قابلية إعادة النتائج

تعكس هذه الإعدادات توجّه النظام نحو التفكير التحليلي المنطقي بدل الإبداع اللغوي.

3.4 إدارة الإعدادات باستخدام Config Object

تم تصميم كائن إعدادات (`Config`) باستخدام `dataclass` لتجمّع جميع معاملات النظام في بنية واحدة، تشمل:

- إعدادات النموذج
- معاملات التوليد
- حدود الاسترجاع المعرفي
- عدد محاولات الإصلاح
- إعدادات التتبّع

يساهم هذا التصميم في:

- تحسين قابلية الصيانة
- سهولة تعديل التجارب
- تعزيز الشفافية الأكاديمية

3.5 قاعدة المعرفة وإدارة السياق المعرفي

3.5.1 قاعدة المعرفة (Agile Knowledge Base)

تم إنشاء قاعدة معرفة صغيرة ومحددة النطاق تحتوي على مفاهيم أساسية من منهجية Scrum و Agile، مثل:

- أدوار Scrum
- مفهوم السبرنت
- أحداث Scrum
- إدارة المخاطر
- تعريف الإنجاز

تم تمثيل كل عنصر ككائن يحتوي على:

- معرف فريد
- نص معرفي مختصر

هذا التصميم يسمح بتبني مصدر كل قرار إداري لاحقاً.

3.6 آلية الاسترجاع المعرفي (RAG)

3.6.1 scikit-learn مكتبة

تم استخدام مكتبة scikit-learn لتنفيذ الاسترجاع المعرفي، وتحديداً:

- تحويل النصوص إلى متجهات عدديّة用 TfidfVectorizer
- حساب درجة التشابه بين وصف المشروع والنصوص المعرفية用 cosine_similarity

3.6.2 TF-IDF منهجية

تعتمد تقنية TF-IDF على قياس أهمية الكلمات داخل النص نسبةً إلى مجموعة النصوص الكاملة، مما يسمح بتحديد أكثر المفاهيم ارتباطاً بمحتوى المشروع.

3.6.3 آلية التصفية

بعد حساب التشابه:

- يتم اختيار أعلى K مصادر معرفية
- يتم تجاهل أي مصدر نقل درجة تشابهه عن عتبة محددة

يضمن ذلك أن المعرفة المستخدمة:

- ذات صلة مباشرة بالمشروع
- غير مفروضة على النموذج
- قابلة للتقسيم والتحليل

3.7 هندسة الموجه (Prompt Engineering)

تم تصميم الموجه النهائي ليعكس بنية الوكيل الذكي، ويكون من:

1. **تعريف دور النظام**
تحديد أن النموذج يعمل كمدير مشاريع برمجية خبير
2. **تعليمات التحكم السلوكية**
توجيه النموذج لاتخاذ قرارات، وتبريرها، وضمان واقعيتها
3. **السياق المعرفي المسترجع**
إدراج نصوص Agile المسترجعة مع تقيد استخدامها
4. **عقد الإخراج (Output Contract)**
فرض إخراج JSON صارم ببنية ثابتة

هذا التصميم يحول النموذج من مولد نصوص إلى نظام تخطيط قائم على القيود.

3.8 معالجة المخرجات والتحقق البنوي

3.8.1 استخراج JSON

تم تطوير دوال مخصصة لاستخراج JSON من مخرجات النموذج، مع التعامل مع الأخطاء الشائعة مثل:

- وجود نصوص إضافية
- تنسيقات غير صحيحة

3.8.2 التحقق من صحة البنية

يتم التحقق من:

- وجود جميع الحقول الإلزامية
- صحة القيم المسموح بها
- عدد السيرناتات
- وجود ملخص إداري ومصادر معرفية

تمنع هذه الخطوة تمرير مخرجات لغوية غير صالحة كخطط تنفيذية.

3.9 التقييم الذاتي والإصلاح التلقائي

3.9.1 التقييم الذاتي

بعد توليد الخطة، يعاد تمريرها للنموذج ضمن دور مقيم مستقل يقوم بـ:

- تقييم قابلية التنفيذ
- تحديد المخاطر
- اقتراح إجراء تصحيحي

3.9.2 آلية الإصلاح

في حال الحاجة للإصلاح:

- يتم إعادة بناء الخطة
- الحفاظ على نفس البنية
- تعديل القرارات فقط

تم تحديد عدد محاولات الإصلاح لمنع الدوران اللانهائي.

3.10 تبع الأداء والشفافية

يقوم النظام بتسجيل:

- زمن التنفيذ
- المصادر المعرفية المستخدمة
- عدد محاولات الإصلاح
- مستوى القابلية النهائي

توفر هذه البيانات سجلاً شفافاً لعملية اتخاذ القرار داخل الوكيل.

3.11 خاتمة الفصل:

قدم هذا الفصل عرضاً تفصيلياً للمنهجية المعتمدة في بناء وكيل مدير مشاريع ذكي قائم على النماذج اللغوية الكبيرة، مدعوماً بالآلية استرجاع معرفي محدودة، ونظام تحقق بنوي، وتقدير ذاتي، وإصلاح تلقائي.

يُظهر هذا التصميم كيف يمكن تحويل نموذج لغوي إلى وكيل مستقل قادر على التخطيط واتخاذ القرار ضمن قيود واقعية، مما يمهد للفصل التالي الذي يناقش النتائج، التحليل، وسلوك الوكيل في سيناريوهات تطبيقية مختلفة.

4 الفصل الرابع: النتائج والخلاصة

4.1 مقدمة شاملة

يُعد هذا الفصل حجر الزاوية في هذا المشروع البحثي، حيث يتم فيه عرض وتحليل النتائج العملية التي تم التوصل إليها من خلال تنفيذ اختبار وكيل ذكي مستقل لإدارة المشاريع البرمجية. يهدف هذا الفصل إلى تقييم مدى فعالية المنهجية المقترحة في تحويل وصف نصي لمشروع برمجي إلى خطة تنفيذ واقعية تعتمد على مبادئ Agile / Scrum، وذلك باستخدام نموذج لغوي كبير يعمل محلياً ومدعوم بالآلية استرجاع معرفي.(RAG).

يركز هذا الفصل على تقديم:

- نتائج كمية قابلة للقياس (زمن التنفيذ، عدد محاولات الإصلاح، مستوى الجدوى)
- نتائج نوعية تتعلق بجودة الخطط المنتجة
- تحليل دلالات هذه النتائج من منظور هندسة الوكلاء الذكية(AI Agents)

الهدف النهائي هو تحديد مدى نجاح النظام في أداء دور مدير مشروع ذكي قادر على التخطيط، التقييم الذاتي، واتخاذ القرار ضمن قيود واقعية.

4.2 منهجية التقييم

لضمان تقييم دقيق وموثوق لأداء الوكيل الذكي، تم اعتماد منهجية تقييم تجمع بين مقاييس كمية واضحة وتحليل نوعي لبنية المخرجات.

4.2.1 معايير التقييم الكمية

تم اعتماد المعايير التالية، المستخرجة مباشرة من تنفيذ النظام:

زمن التنفيذ الكلي(Runtime)

يمثل الزمن المستغرق منذ لحظة إدخال وصف المشروع وحتى إنتاج الخطة النهائية بعد عمليات التقييم والإصلاح الذاتي.

- متوسط زمن التنفيذ:
(745 ثانية)
- يتضمن:
 - استرجاع المعرفة(RAG)
 - توليد الخطة
 - التقييم الذاتي
 - عمليات الإصلاح(Self-Repair)

يدل هذا الزمن على كلفة حسابية مرتفعة نسبياً، لكنها متوقعة نظراً للتنفيذ المحلي الكامل وعدم الاعتماد على خدمات سحابية.

عدد محاولات الإصلاح الذاتي (Repair Attempts)

يمثل هذا المقياس قدرة الوكيل على تصحيح قراراته ذاتياً.

- عدد المحاولات المنفذة : 2
- الحد الأقصى المسموح : 2

يشير ذلك إلى أن الخطة الأولية لم تكن مثالية، لكن الوكيل تمكّن من تحسينها ضمن القيود المحددة، وهو سلوك متوقع في الأنظمة الذكية المستقلة.

مستوى الجدوى النهائي (Final Feasibility)

بعد آخر عملية تقييم ذاتي، صنف الوكيل الخطة على أنها:

Medium Feasibility •

هذا التقييم يأخذ بعين الاعتبار:

- عدد المطوريين
- الإطار الزمني (6 أسابيع)
- نطاق الميزات المطلوبة

ويمثل هذا المخرج عنصراً مهماً، إذ لا يكتفي النظام بتوليد خطة، بل يُصدر حكماً تحليلياً على قابليتها للتنفيذ.

استخدام المعرفة المسترجعة (RAG Usage)

- عدد عناصر المعرفة المسترجعة : عنصر واحد
- المصدر المستخدم : SCRUM_ROLES
- درجة التشابه : 0.34

يدل ذلك على أن الوكيل:

- لا يفرط في استخدام المعرفة
- يعتمد فقط على ما هو ذو صلة مباشرة بالمشروع

وهو مؤشر إيجابي على كفاءة آلية RAG.

4.2.2 التقييم البنوي للمخرجات (Structural Evaluation)

تم تقييم الخطة الناتجة بناءً على:

- الالتزام الصارم بمخيط JSON
- اكمال جميع الحقول الإلزامية
- ترابط الأهداف، المهام، والمخاطر

أظهرت النتائج أن جميع الخطط:

- مطابقة للمخطط
- قابلة للقراءة الآلية
- قابلة للتحليل والمقارنة

4.3 نتائج خط الأساس (Baseline Behavior)

يتمثل خط الأساس في الخطة الأولية قبل الإصلاح الذاتي.

لوحظ في هذه المرحلة:

- ميل لتوسيع النطاق الوظيفي
- تقليل تقدير المخاطر المتعلقة بالموارد البشرية

وهو سلوك شائع في النماذج اللغوية عند غياب التقييم الذاتي.

4.4 النتائج النهائية بعد الإصلاح الذاتي

بعد تطبيق آلية Self-Evaluation + Self-Repair، أظهرت الخطة النهائية التحسينات التالية:

4.4.1 تحسن إدارة النطاق (Scope Management)

- تحديد وأوضاع الميزات الأساسية
- التركيز على الوظائف الجوهرية فقط
- تقليل احتمالية التمدد غير المنضبط (Scope Creep)

4.4.2 تحسين تحليل المخاطر

أضاف الوكيل:

- مخاطر متعلقة بالموارد البشرية
- استراتيجيات تخفيف واقعية (Mitigation)

وهو عنصر أساسي في إدارة المشاريع الاحترافية.

4.4.3 توزيع منطقي للمهام عبر السيرنات

- تقسيم العمل إلى مرحلتين واضحتين
- مواءمة الأهداف مع قدرات فريق صغير (مطوران فقط)

4.5 مناقشة النتائج

4.5.1 فعالية الوكيل الذكي

أثبتت النتائج أن النظام قادر على:

- فهم وصف مشروع غير مهيكل
- تحويله إلى خطة منظمة
- تقييم جودة قراراته
- تعديلها ذاتياً دون تدخل بشري

وهذا يضع النظام ضمن فئة **Autonomous Decision-Making Agents**.

4.5.2 أهمية عدم استخدام أطر جاهزة

أظهر التنفيذ اليدوي لجميع المراحل أن:

- منطق الوكيل شفاف بالكامل
- كل قرار قابل للتفسير
- لا يوجد سلوك “صندوقي أسود”

وهو عنصر حاسم في الأبحاث الأكاديمية.

4.5.3 حدود النظام الحالية

على الرغم من النتائج الإيجابية، لا تزال هناك بعض القيود:

- زمن التنفيذ مرتفع نسبياً
- تقييم الجدوى يعتمد على النموذج اللغوي فقط
- قاعدة المعرفة Agile محدودة الحجم

5 الفصل الخامس: الأعمال المستقبلية والتوصيات

5.1 مقدمة شاملة

أثبتت هذا المشروع فعالية بناء وكيل ذكي مستقل (Autonomous AI Agent) يعمل محلياً لتخفيط المشاريع البرمجية وفق مبادئ Agile / Scrum اعتماداً على نموذج لغوي كبير (LLM) مع آلية استرجاع معرفي بسيطة (RAG) مبنية على آلية JSON Schema Validation ، إضافة إلى طبقة تحقق بنوي للمخرجات (Self-Evaluation / Self-Repair).

أظهرت النتائج أن النظام قادر على تحويل وصف مشروع غير مهيكل إلى خطة منظمة / Sprints / Risks / Components مع تبرير قراراته ضمن عقد مخرجات ثابت. ومع ذلك، فإن مجال الوكالء الذكية لتخفيط المشاريع لا يزال في تطور سريع، ويزخر بفرص كبيرة للتحسين، سواء على مستوى جودة الاسترجاع المعرفي، أو دقة التقييم الذاتي، أو سرعة التنفيذ، أو التعليم على مشاريع متعددة ومعقدة.

يهدف هذا الفصل إلى تقديم خارطة طريق واضحة للأعمال المستقبلية التي يمكن أن تبني على ما تم إنجازه، وتدفع هذا النظام نحو مستوى أكثر احترافية وقابلية للتطبيق في بيئات واقعية (Real-World Project Management).

5.2 التوصيات المستقبلية

5.2.1 أولاً: تطوير قاعدة المعرفة وآلية الاسترجاع (Knowledge Base & Retrieval)

تُعد قاعدة المعرفة (AGILE_KB) حجر الأساس في رفع جودة قرارات الوكيل.

❖ توسيع نطاق قاعدة المعرفة (KB Expansion) يوصى بتوسيع محتوى المعرفة ليشمل مفاهيم أكثر عمقاً مثل:

- تحديد سعة الفريق (Capacity Planning)
- إدارة المتطلبات وتقسيمها (User Stories, Epics, Backlog Grooming)
- تقدير الجهد (Story Points, Planning Poker)
- إدارة الاعتمادات (Dependencies)
- تعريفات أكثر تفصيلاً لـ DoD ومعايير الجودة

❖ تحسين الاسترجاع من TF-IDF إلى استرجاع هجين (Hybrid Retrieval) TF-IDF فعال في التقاط التطابقات اللغوية، لكنه قد يفشل في فهم التشابه الدلالي العميق. يمكن تحسين ذلك عبر:

- دمج TF-IDF مع تمثيلات دلالية (Embeddings) محلياً
- اعتماد أسلوب إعادة ترتيب النتائج (Re-ranking) بعد الاسترجاع الأولي
- رفع جودة "التغطية" بإضافة مصادر معرفية متعددة ومنظمة

❖ ربط المعرفة بمخرجات محددة (Grounded Outputs) يوصى بجعل الوكيل ملزماً بإسناد قرارات محددة إلى مصادر مسترجعة (RAG Sources) بشكل أكثر صرامة، مثل: كل "خطر" أو "اختيار منهجي" يجب أن يرتبط بعنصر معرفي أو قاعدة.

5.2.2 ثانياً: تحسين جودة المخرجات والتحقق (Output Quality & Validation)

❖ توسيع التحقق البنوي إلى تحقق دلالي (Semantic Validation) التحقق الحالي يضمن وجود الحقول، لكنه لا يضمن جودة محتواها. يمكن إضافة قواعد تحقق مثل:

- ألا تكون المهام عامة جدًا (مثل "do testing" فقط)
- أن يكون لكل Sprint هدف واضح قابل للقياس
- أن تتوافق التكنولوجيا المقترنة مع طبيعة المشروع (Web/ML/Mobile)
- أن تكون المخاطر مرتبطة فعليًا بقيود المشروع (وقت/فريق/تعقيد)

❖ بناء "مُقيم خارجي" Rule-Based بدلاً من الاعتماد الكامل على تقييم النموذج لنفسه، يمكن بناء مُقيم مستقل:

- قواعد هندسية ثابتة (Best Practices)
- Checklists لمدير مشروع محترف
- كشف التناقضات (مثلاً: 6 أسباب + ميزات كثيرة + فريق صغير)

هذا يرفع موثوقية النظام أمام لجان المناقشة أو الاستخدام الواقعي.

5.2.3 ثالثاً: تطوير آلية التقييم الذاتي والإصلاح الذاتي (Self-Eval & Self-Repair)

❖ تقييم متعدد المراحل (Multi-Stage Evaluation) بدلاً من تقييم واحد شامل، يمكن تقسيم التقييم إلى:

1. تقييم النطاق (Scope)
2. تقييم الجدولة (Timeline)
3. تقييم المخاطر (Risk)
4. تقييم التقنية (Tech Fit)

ثم الإصلاح يتم بناءً على القسم الضعيف فقط، مما يقلل وقت التنفيذ ويزيد الدقة.

❖ التحكم في الإصلاح لمنع "التدحرج" (Repair Regression Control) قد يؤدي الإصلاح أحياناً إلى فقدان جودة جزء آخر. يوصى بتطبيق:

- مقارنة بين النسخ (Plan Versioning)
- حفظ أفضل نسخة وفق معايير واضحة (Best-Plan Selection)

5.2.4 رابعاً: تحسين الكفاءة والسرعة (Efficiency & Performance)

أظهرت التجربة أن زمن التنفيذ مرتفع نسبياً بسبب تعدد استدعاءات النموذج.

❖ تقليل عدد الاستدعاءات (Fewer LLM Calls)
يمكن دمج بعض الخطوات:

- توليد الخطة + تقييم أولي داخل استدعاء واحد
- أو استخدام تقييم مختصر قبل الدخول في الإصلاح الكامل

❖ تحسين إعدادات التشغيل المحلي
مثل:

- ضبط n_ctx وفق الحاجة الفعلية
- رفع كفاءة عدد الخيوط n_threads حسب العتاد
- تقليل max_tokens في التقييم الذاتي

❖ التخزين المؤقت (Caching)

تخزين نتائج الاسترجاع أو أجزاء من الاستجابة عند تكرار نفس نوع المشروع، لتقليل زمن التنفيذ.

5.2.5 خامساً: توسيع نطاق الاستخدام والتطبيقات (Broader Applications)

❖ دعم مشاريع متعددة الأنواع
حالياً الاختبار تم على مشروع ويب (عيادة). يوصى باختبار النظام على:

- مشاريع ذكاء اصطناعي (ML Pipelines)
- تطبيقات موبайл
- أنظمة IoT
- مشاريع مؤسساتية كبيرة متعددة الفرق

❖ إضافة مخرجات احترافية لبيانات العمل
مثل:

- إنشاء backlog مفصل (User Stories + Acceptance Criteria)
- خطة مخاطر مصنفة (Likelihood/Impact)
- على مراحل Roadmap (Milestones)
- لكل Sprint بشكل رسمي Deliverables

5.3 الخلاصة

قدم هذا المشروع أساساً متيّزاً لبناء وكيل ذكي مستقل لتخطيط المشاريع البرمجية محلّياً، مع دمج استرجاع معرفي قابل للتفسير، وعقد مخرجات منظم، وأالية تقييم ذاتي وإصلاح ذاتي. ومع ذلك، فإن التطوير المستقل يجب أن يركز على خمس ركائز رئيسية: توسيع المعرفة، تحسين الاسترجاع، رفع جودة التحقيق، تعزيز التقييم المستقل، وتحسين الكفاءة الحسابية.

إن الالتزام بهذه التوصيات يفتح الطريق نحو تطوير جيل جديد من أنظمة إدارة المشاريع الذكية: أنظمة أكثر موثوقية، أسرع، وأقرب لطريقة تفكير مدير مشروع بشري محترف، وقدرة على إحداث أثر تطبيقي واضح في بيئات العمل الحقيقة.

٦ الفصل السادس: المراجع

References

1. T. Brown et al., “Language Models are Few-Shot Learners,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.
2. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.
3. A. Vaswani et al., “Attention Is All You Need,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
4. S. Minaee et al., “Large Language Models: A Survey,” *arXiv preprint arXiv:2402.06196*, 2024.
5. P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.
6. C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.
7. G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
8. J. Ramos, “Using TF-IDF to Determine Word Relevance in Document Queries,” *Proceedings of the First Instructional Conference on Machine Learning*, 2003.
9. A. Singhal, “Modern Information Retrieval: A Brief Overview,” *IEEE Data Engineering Bulletin*, vol. 24, no. 4, pp. 35–43, 2001.
10. S. Deerwester et al., “Indexing by Latent Semantic Analysis,” *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
11. G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network,” *arXiv preprint arXiv:1503.02531*, 2015.
12. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
13. G. Marcus and E. Davis, “GPTs Are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models,” *arXiv preprint arXiv:2303.10130*, 2023.
14. H. Chen et al., “Evaluating Large Language Models Trained on Code,” *arXiv preprint arXiv:2107.03374*, 2021.
15. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2021.
16. M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed., Wiley, 2009.
17. S. Franklin and A. Graesser, “Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents,” *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, 1996.
18. D. Silver et al., “Mastering the Game of Go with Deep Neural Networks and Tree Search,” *Nature*, vol. 529, pp. 484–489, 2016.
19. K. Schwaber and J. Sutherland, *The Scrum Guide*, Scrum.org, 2020.
20. M. Cohn, *Agile Estimating and Planning*, Prentice Hall, 2005.
21. R. Pressman and B. Maxim, *Software Engineering: A Practitioner’s Approach*, 9th ed., McGraw-Hill, 2019.

22. B. Boehm, “A Spiral Model of Software Development and Enhancement,” *IEEE Computer*, vol. 21, no. 5, pp. 61–72, 1988.
23. M. Fowler, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 2018.
24. OpenAI, “GPT-4 Technical Report,” *arXiv preprint arXiv:2303.08774*, 2023.
25. Qwen Team, “Qwen2.5: Technical Report,” Alibaba DAMO Academy, 2024.
26. G. Gerganov, “llama.cpp: Port of Facebook’s LLaMA Model in C/C++,” GitHub Repository, 2023.
Available: <https://github.com/ggerganov/llama.cpp>
27. Python Software Foundation, *Python Language Reference*, version 3.10, 2024.
Available: <https://www.python.org>
28. F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
29. Project Jupyter, “Jupyter Notebook Documentation,” Accessed: Jul. 2025.
Available: <https://jupyter.org>
30. JSON Schema Organization, “JSON Schema: A Media Type for Describing JSON Documents,” 2023.
Available: <https://json-schema.org>
31. E. Gamma et al., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
32. The Turing Way Community, *The Turing Way: A Handbook for Reproducible, Ethical and Collaborative Research*, The Alan Turing Institute, 2022.
Available: <https://the-turing-way.netlify.app/>