



الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

الجمهورية العربية السورية
الجامعة السورية الخاصة
كلية هندسة الذكاء الصناعي
المشروع تخرج واحد

AI PROJECT MANGER

إعداد الطلاب:

أحمد محمد باسل الطباع

محمد عدنان لحام

إشراف:

د. اصف جعفر

2026\2027

شهادة مشرف :

اسم..... :

تاريخ :

التوقيع:

الملخص

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

في هذا المشروع، يتم تطوير وكيل ذكاء صناعي 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-Augmented Generation – RAG)، 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 مقيدة 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 المقترح، وقدرته على دعم قرارات إدارة المشروع بشكل قابل للتفسير.

في RAG يقدّم هذا المشروع مثلاً تطبيقياً على توظيف تقنيات الذكاء الصناعي الحديثة، ولا سيما النماذج اللغوية الكبيرة و بناء أنظمة ذكية لدعم القرار في مجال هندسة البرمجيات، مع استعراض التحديات الحالية والآفاق المستقبلية لتطوير وكلاء ذكاء صناعي 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

المُلخَص.....	3
Abstract.....	4
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 العمل (Scope of Work).....	15
1.5 مفهوم تعريف الوكيل الذكي (Artificial Intelligence Agent).....	15
1.6 الوكيل الذكي في سياق النماذج اللغوية الكبيرة:.....	16
1.7 تعريف وكيل مدير المشروع الذكي المقترح في هذا المشروع.....	16
1.8 خصائص الوكيل الذكي في هذا المشروع.....	16
1.9 AI Agents موقع هذا الوكيل ضمن أطروحات.....	17
الفرق بين توليد النص الحر ودعم القرار الذكي في الوكلاء الذكيين.....	17
(Text Generation) توليد النصوص باستخدام النماذج اللغوية.....	17
(AI Agent-Based Decision Support) دعم القرار الذكي باستخدام الوكلاء الذكيين.....	18
الفارق الجوهرى بين النهجين.....	19
1.10 التحديات المرتبطة بتطوير وكيل مشروع ذكي.....	19
1.11 التحديات المستقبلية في مجال وكلاء الذكاء الصناعي.....	21
لإدارة المشاريع البرمجية.....	21
1.12 خاتمة الفصل.....	22
2 الفصل الثاني: الدراسة المرجعية.....	23
2.1 مقدمة الفصل.....	24
2.2 تصنيف تقنيات بناء الوكلاء الذكيين (AI Agents).....	25
2.2.1 الوكلاء المعتمدون على القواعد (Rule-Based Agents).....	25
2.2.2 الوكلاء المعتمدون على النماذج والاستدلال (Model-Based and Reasoning Agents).....	25
2.2.3 الوكلاء المعتمدون على التعلم (Learning-Based Agents).....	26
2.3 النماذج اللغوية الكبيرة في بناء الوكلاء الذكيين.....	26

2.3.1	الوكلاء المعتمدون على النماذج اللغوية الكبيرة (LLM-Based Agents)	26
2.3.2	الوكلاء المعززون بالمعرفة (RAG-Enhanced Agents).....	27
2.3.3	الوكلاء القائمون على آليات الانتباه والتخطيط السياقي.....	27
2.4	مقاييس تقييم أداء الوكيل الذكي وجودة المخرجات.....	27
2.4.1	دقة الاتساق بين المدخلات والمخرجات (Input-Output Alignment Accuracy)	28
2.4.2	جودة التحليل البنيوي للمشروع (Structural Analysis Quality)	28
2.4.3	دقة التخطيط الزمني وإدارة المخاطر	29
2.4.4	موثوقية القرارات المدعومة بالمعرفة (RAG-Grounded Decision Reliability)	29
2.4.5	قابلية التفسير والوضوح الإداري (Explainability & Managerial Readability)	29
2.4.6	التقييم البشري المقارن (Human-in-the-Loop Evaluation)	30
2.5	استعراض الدراسات المرجعية (Related Work)	30
2.5.1	الدراسة المرجعية الأولى.....	31
2.5.2	الدراسة المرجعية الثانية.....	31
2.5.3	الدراسة المرجعية الثالثة.....	32
2.5.4	الدراسة المرجعية الرابعة.....	32
2.5.5	الدراسة المرجعية الخامسة.....	33
2.6	خلاصة الاستعراض وتحديد الفجوة البحثية.....	33
3	الفصل الثالث: المنهجية المعتمدة.....	34
3.1	مقدمة الفصل.....	35
3.2	بيئة التطوير والأدوات البرمجية المستخدمة.....	35
3.2.1	لغة البرمجة.....	36
3.3	تشغيل النموذج اللغوي الكبير محليًا:.....	36
3.3.1	مكتبة llama_cpp	37
3.3.2	النموذج المستخدم.....	37
3.4	إدارة إعدادات الوكيل باستخدام كائن AgentConfig:.....	38
3.5	قاعدة المعرفة وإدارة السياق المعرفي:.....	39
3.5.1	قاعدة المعرفة النصية للمشروع (Project Textual Knowledge Base).....	39
3.6	آلية الاسترجاع المعرفي (Retrieval-Augmented Generation).....	40
3.6.1	بنية الاسترجاع المعرفي في النظام:.....	41

3.6.2	scikit-learn باستخدام مكتبة TF-IDF الاسترجاع القائم على	41
3.6.3	آلية الاسترجاع القائم على التمثيل الدلالي باستخدام	
	SentenceTransformers : بعد حساب التشابه	42
3.7	هندسة الموجه (Prompt Engineering)	42
3.7.1	تعريف دور النظام	43
3.7.2	تعليمات التحكم السلوكي	43
3.7.3	دمج السياق المعرفي المسترجع	43
3.8	معالجة المخرجات والتحقق البنيوي:	44
3.8.1	المقيّد بنحو رسمي JSON توليد واستخراج	44
3.8.2	(Schema Validation) التحقق البنيوي باستخدام مخطط البيانات	45
3.8.3	آليات الإكمال القسري وإعادة التوليد المنضبط	45
3.8.4	الأثر المنهجي للتحقق البنيوي	46
3.9	تتبع الأداء والشفافية	46
3.10	خاتمة الفصل:	47
4	الفصل الرابع: النتائج والخلاصة	48
4.1	مقدمة شاملة	49
4.2	منهجية التقييم (Evaluation Methodology)	50
4.2.1	(Operational Quantitative Evaluation) التقييم التشغيلي الكمي	51
	(Runtime) زمن التنفيذ الكلي	51
	(Self-Repair Attempts) عدد محاولات الإصلاح الذاتي	51
	(Final Feasibility Assessment) مستوى الجدوى النهائي	52
	(RAG Utilization Indicator) مؤشر استخدام الاسترجاع المعرفي	52
4.3	(Structural Output Validation) التقييم البنيوي للمخرجات	53
4.4	(Behavioral Analysis) تحليل السلوك قبل وبعد التقييم الذاتي	53
4.5	خلاصة منهجية التقييم	54
5	الفصل الخامس: الأعمال المستقبلية والتوصيات	55
5.1	مقدمة شاملة	56
5.2	التوصيات المستقبلية	57
5.2.1	تطوير قاعدة المعرفة وآلية الاسترجاع المعرفي	57
5.2.2	تحسين جودة المخرجات وآليات التحقق	58
5.2.3	تطوير آليات التقييم الذاتي والإصلاح الذاتي	58
5.2.4	تحسين الكفاءة والأداء الحسابي	59

5.2.5	توسيع نطاق الاستخدام والتطبيق	59
5.3	الخلاصة	60
6	الفصل السادس: المراجع	61
	References	62

1 الفصل الأول: مقدمة

1.1 مقدمة عامة

في العصر الرقمي الحديث، أصبحت المشاريع البرمجية جزءًا لا يتجزأ من مختلف جوانب الحياة اليومية والتطبيقات العملية والعلمية، بدءًا من الأنظمة الخدمية والتطبيقات الذكية، مرورًا بمنصات الأعمال والتجارة الإلكترونية، وصولًا إلى الأنظمة الحرجة في مجالات مثل الصحة، والمال، والتعليم، والاتصالات. ومع هذا التوسع المتسارع، أصبحت إدارة المشاريع البرمجية بكفاءة ودقة عاملاً حاسماً في نجاح هذه الأنظمة، لا سيما في ظل التعقيد المتزايد للمتطلبات وتعدد الأطراف المعنية وتسارع دورات التطوير.

وعلى الرغم من التقدم الكبير في أدوات تطوير البرمجيات ومنهجيات إدارتها، لا تزال العديد من المشاريع تعاني من مشكلات جوهرية 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 تعقيداً مع ازدياد حجم المشاريع واعتمادها على تقنيات متقدمة تتطلب تنسيقاً عالي المستوى بين الجوانب التقنية والإدارية.

من هنا، برزت الحاجة إلى حلول ذكية قادرة على دعم عملية إدارة المشاريع البرمجية بشكل منهجي وفعال. ويُعد توظيف تقنيات الذكاء الصناعي في هذا المجال من الاتجاهات البحثية الواعدة، 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 وتفكيكه إلى مراحل واضحة قابلة للتنفيذ.

وقد شهد هذا المجال تطوراً ملحوظاً مع ظهور النماذج اللغوية الكبيرة وتقنيات التوليد الذكي، التي مكّنت من معالجة الأوصاف النصية للمشاريع البرمجية وفهمها دلاليًا. 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 الصناعي بحيث تكون قابلة للتفسير وموثوقة في البيئات العملية.

انطلاقاً من هذه التحديات، يأتي هذا المشروع ليقدم نموذجاً لوكيل ذكاء صناعي واحد يعمل كمدير مشروع برمجي ذكي، يهدف إلى دعم عملية اتخاذ القرار الإداري من خلال تحليل الأوصاف النصية للمشاريع البرمجية، والاستعانة بمعرفة منهجية منظمة، وتوليد مخرجات مقيّدة 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 صناعي واحد يؤدي دور مدير مشروع برمجي. يهدف هذا الوكيل إلى دعم عمليات التحليل والتخطيط واتخاذ القرار الإداري، وذلك بالاعتماد على أحدث تقنيات الذكاء الصناعي 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 حقيقية ذات موارد محدودة.

يستند المشروع إلى دراسة تحليلية لمختلف الأساليب المستخدمة في إدارة المشاريع البرمجية، بدءًا من النماذج التقليدية المعتمدة على الخبرة البشرية، وصولاً إلى التوجهات الحديثة التي توظف 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 المعزز بالاسترجاع (Retrieval-Augmented Generation – RAG)، بما يضمن أن تكون القرارات المقترحة مستندة إلى ممارسات تطوير برمجيات معتمدة وليست 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 المقترحة 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 حقيقية.

وتكمن أهمية هذا المشروع في اتساع نطاق تطبيقاته العملية، حيث يمكن توظيفه في فرق تطوير برمجيات صغيرة ومتوسطة، وشركات تقنية ناشئة، ومراكز تطوير تعتمد منهجيات 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 صناعي أكثر تقدمًا وفاعلية في مجال دعم القرار وإدارة المشاريع المعتمدة على الذكاء الصناعي.

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، وتقدير 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:** غالبًا 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.
- **البيئات 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 السبرنتات أو ضعف في إدارة المخاطر.

تكمن أهمية هذا البحث في أنه لا يقدّم مجرد أداة لتوليد نصوص إدارية، بل يطرح حلًا ذكيًا عمليًا يعتمد على تقنيات الذكاء الصناعي لدعم عملية إدارة المشاريع البرمجية بشكل منهجي. فمن خلال توظيف النماذج اللغوية الكبيرة وآلية التوليد المعزّز بالاسترجاع (RAG)، يتيح 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 آفاقًا 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 يعزز نجاح المشاريع البرمجية في مختلف القطاعات التقنية والتعليمية.

1.4 أهداف المشروع ومحاوره 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 توصيفات عامة أو نصوص إدارية، بل يسعى المشروع إلى تقديم نموذج عملي وموثوق قادر على تحليل أوصاف المشاريع البرمجية النصية وتحويلها إلى قرارات إدارية وخطط تنفيذ منهجية مبنية على أسس 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 و Scrum ضمن عملية التحليل، بما يضمن أن تكون القرارات الإدارية وخطط السبرنت مستندة إلى معرفة منهجية معتمدة وليست ناتجة عن توليد لغوي حر.
3. ضمان الاتساق بين المدخلات والمخرجات:
تطوير آلية إخراج مقيّدة ببنية ثابتة قابلة للتحقق آلياً، بهدف الحد من الانحراف الدلالي وضمان أن تكون مخرجات النظام مرتبطة منطقياً بمحتوى المشروع المدخل وقابلة للتفسير والتحليل.
4. تقييم فعالية النظام من منظور ذكاء صناعي:
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 بيئات تطوير برمجيات حقيقية ذات موارد محدودة.

1.4.2 محاور العمل (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 برمجية 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، بدءًا من الإطار النظري، مرورًا بمنهجية العمل والتنفيذ، وانتهاءً بعرض النتائج ومناقشتها واقتراح آفاق التطوير المستقبلية لوكلاء الذكاء الصناعي في مجال إدارة المشاريع البرمجية.

1.5 مفهوم تعريف الوكيل الذكي (Artificial Intelligence Agent)

يُعرّف 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 (15). ويتميّز الوكيل الذكي بقدرته على العمل بشكل ذاتي، والتكيف مع المدخلات المتغيرة، واستخدام المعرفة والخبرة المتراكمة لتحسين جودة قراراته مع مرور الوقت (16،17).

وفي السياق الحديث لأنظمة الذكاء الصناعي، لا يقتصر مفهوم الوكيل الذكي على التفاعل الحسي المباشر مع البيئة الفيزيائية، بل يمتد ليشمل التفاعل مع بيانات معلوماتية، مثل النصوص، والبيانات غير المهيكلة، والأنظمة الرقمية المعقدة. وفي هذه الحالات، يكون الإدراك قائمًا على تحليل المحتوى الدلالي للمدخلات، بينما تتمثل الأفعال في توليد قرارات أو توصيات أو مخرجات منظمة تدعم عمليات بشرية معقدة (16،17).

1.6 الوكيل الذكي في سياق النماذج اللغوية الكبيرة:

مع ظهور النماذج اللغوية الكبيرة، تطوّر 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، تقتصر إلى البنية المنهجية والقيود اللازمة لضمان موثوقية القرارات. لذلك، يتم دمجها ضمن إطار وكيل ذكي يحدد أدوارها، ويضبط سلوكها، ويقيّد مخرجاتها بما يخدم هدفاً واضحاً.

1.7 تعريف وكيل مدير المشروع الذكي المقترح في هذا المشروع

في ضوء التعريف السابق، يمكن تعريف وكيل مدير المشروع الذكي المقترح 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 مقيّدة بسياق محدد وبنية إخراج منظمة.

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 فتتمثل 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

يتميّز وكيل مدير المشروع الذكي المقترح 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 مقيّدة ببنية ثابتة قابلة للتحقق، مما يحدّ من الانحراف الدلالي.
- **دعم القرار:** يعمل كمنظومة دعم قرار ذكية، وليس كنظام توليد نصوص حر.

1.9 موقع هذا الوكيل ضمن أطروحات AI Agents

يُمثل هذا الوكيل نموذجًا تطبيقيًا لوكلاء الذكاء الصناعي 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 التوسع والتطوير المستقبلي.

الفرق بين توليد النص الحر ودعم القرار الذكي في الوكلاء الذكيين

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

توليد النصوص باستخدام النماذج اللغوية (Text Generation)

- **الهدف:**
يركّز هذا النهج على إنتاج نصوص لغوية متماسكة وسليمة نحويًا بناءً على مدخلات عامة، دون الالتزام بالضرورة بسياق تطبيقي محدد أو هدف إداري واضح.
- **الآلية:**
يعتمد على التوليد الاحتمالي للنصوص استنادًا 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.
 - تلخيص 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 مباشرة لاتخاذ قرارات حساسة أو تنفيذية.

دعم القرار الذكي باستخدام الوكلاء الذكيين (AI Agent–Based Decision Support)

- **الهدف:**

يهدف هذا النهج إلى دعم عملية اتخاذ القرار ضمن سياق محدد، من خلال تحليل المدخلات وفهمها دلاليًا، ثم تحويلها إلى مخرجات منظمة وقابلة للتنفيذ.

- **الآلية:**

يعمل النموذج اللغوي ضمن إطار وكيل ذكي مقيّد، حيث يتم:

- تحديد هدف واضح للوكيل.
- تقييد نوع المدخلات.
- فرض بنية ثابتة للمخرجات.
- توجيه القرارات بالاعتماد على معرفة منهجية مسترجعة (RAG).

- **العمليات:**

تشمل:

- تحليل الأوصاف النصية غير المهيكلة.
- استنتاج أهداف المشروع ومكوناته.
- تقدير مستوى التعقيد والمخاطر.
- اقتراح خطط تنفيذ وسبرنتات 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 فى أن الأول يركّز على تحسين الشكل اللغوى للنص دون ضمان الارتباط بالسياق أو الهدف، فى حين يسعى الثانى إلى بناء معنى وظيفى قابل للتنفيذ من خلال ضبط السلوك الإدراكى للنموذج ضمن إطار وكيل ذكى.

وفى هذا المشروع، لا يتم استخدام النموذج اللغوى بوصفه مولدًا للنصوص فقط، بل يتم توظيفه كعنصر إدراكى داخل وكيل ذكاء صناعى موجه بالمعرفة، يهدف إلى دعم قرارات إدارة المشاريع البرمجية بشكل منهجى وموثوق.

1.10 التحديات المرتبطة بتطوير وكيل مشروع ذكى

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

- **عدم اكتمال وغموض المدخلات النصية:**

غالبًا 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. ويشكّل استنتاج هذه المعلومات بدقة دون افتراضات غير مبررة أحد أبرز التحديات فى تصميم الوكيل الذكى.

- **الانحراف الدالالى فى المخرجات:**

تميل النماذج اللغوية الكبيرة، فى 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 تحديًا أساسيًا فى أنظمة الوكلاء الذكيين.

- ضبط التوازن بين المرونة والصرامة:

يتطلب وكيل مدير المشروع الذكي درجة من المرونة للتعامل مع مشاريع متنوعة 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 تحديًا تصميميًا معقدًا.

- الاعتماد على المعرفة المنهجية الموثوقة:

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

- الكلفة الحسابية وكفاءة التنفيذ:

تتطلب النماذج اللغوية الكبيرة موارد حوسبية وذاكرة مرتفعة نسبيًا، 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 تقديم مخرجات يمكن تتبع منطقها وربطها بالمدخلات والمعرفة المستخدمة تحديًا أساسيًا 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 تعميم أداء الوكيل على هذه السيناريوهات الواقعية أحد التحديات المفتوحة التي يسعى هذا المشروع إلى معالجتها تدريجيًا.

1.11 التحديات المستقبلية في مجال وكلاء الذكاء الصناعي لإدارة المشاريع البرمجية

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

- **نحو وكلاء يتعاملون مع الغموض وعدم الاكتمال (Blind Project Understanding) :**
يتوجب تطوير وكلاء ذكيين قادرين على تحليل أوصاف مشاريع واقعية غير مكتملة أو غير دقيقة، دون توفر معرفة مسبقة واضحة حول جميع المتطلبات أو القيود. ويُعد هذا التوجه خطوة أساسية نحو بناء وكلاء ذكاء صناعي يمكن الاعتماد عليهم في البيئات العملية الحقيقية، حيث نادرًا ما تكون مدخلات المشاريع مكتملة أو مصاغة بشكل مثالي.
- **تحسين كفاءة الوكلاء وتقليل الكلفة الحسابية:**
مع ازدياد تعقيد النماذج اللغوية المستخدمة ضمن الوكلاء الذكيين، تبرز الحاجة إلى تصميم وكلاء أخف وزنًا وأكثر كفاءة من حيث استهلاك الموارد الحوسبية 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 لوكلاء الذكاء الصناعي مستوى عاليًا 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 توسع استخدام الوكلاء الذكيين في البيئات المؤسسية، تزداد الحاجة إلى وكلاء قادرين على تفسير قراراتهم بشكل واضح للمستخدم البشري. ويشمل ذلك توضيح أسباب اختيار تقنيات معينة، أو تقسيم السبرنتات، أو تقدير مستوى المخاطر، بما يعزز الثقة في النظام ويدعم اتخاذ القرار المشترك بين الإنسان والذكاء الصناعي.
- **تكامل الوكلاء مع مهام إدارية وتقنية أخرى:**
يتجه البحث الحديث نحو تصميم وكلاء ذكيين 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. ويسهم هذا التكامل في بناء منظومات ذكية أكثر شمولية وفاعلية في دعم دورة حياة المشروع البرمجي بالكامل.

1.12 خاتمة الفصل

يختتم هذا الفصل بعرض الإطار المفاهيمي والمنهجي الذي يشكل الأساس العلمي لهذا البحث في مجال الذكاء الصناعي. حيث تم في بدايته تحديد سياق المشكلة المرتبطة بإدارة المشاريع البرمجية الحديثة، مع تسليط الضوء على الأهمية المتزايدة للتخطيط المنهجي واتخاذ القرار الإداري المدعوم بالبيانات في ضمان نجاح المشاريع التقنية، ولا سيما في ظل ازدياد تعقيد الأنظمة البرمجية، وضغط الزمن، وتعدد المتطلبات الوظيفية والتنظيمية.

وانطلاقاً من هذا السياق، تم تحديد إشكالية البحث المتمثلة في محدودية كفاءة الأساليب التقليدية في تحليل أوصاف المشاريع البرمجية المكتوبة بصيغة نصوص غير مهيكلة، وما يترتب على ذلك من غموض في تحديد الأهداف، وضعف في تقدير المخاطر، وعدم دقة في إعداد خطط التنفيذ. كما تم بيان الانعكاسات العملية لهذه الإشكالية في بيئات تطوير برمجيات واقعية، سواء ضمن الفرق الصغيرة، أو الشركات الناشئة، أو السياقات التعليمية. وبناءً على ذلك، تم اشتقاق أهداف المشروع ومحاوره التنفيذية بشكل واضح، لتوجيه مسار البحث نحو تطوير وكيل ذكاء صناعي قادر على دعم القرار الإداري بطريقة منهجية وقابلة للتفسير.

ولضمان الدقة الاصطلاحية والوضوح المفاهيمي، قَدِّم الفصل تعريفاً منهجياً لمفهوم الوكيل الذكي في سياق النماذج اللغوية الكبيرة، مع توضيح موقع وكيل مدير المشروع الذكي بوصفه نظام دعم قرار يعتمد على المعرفة والسياق، وليس مجرد نظام لتوليد النصوص. كما تم التمييز بصورة منهجية بين التوليد اللغوي الحر وأنظمة الوكلاء الذكيين المقيدة بالسياق والمعرفة، بهدف تحديد نطاق البحث وحدوده التطبيقية بشكل دقيق.

كما تناول الفصل بصورة تحليلية التحديات الراهنة والمستقبلية المرتبطة بتطوير وكلاء ذكاء صناعي مخصصة لإدارة المشاريع البرمجية، بدءاً من مشكلات الغموض وعدم اكتمال المدخلات، مروراً بقضايا الانحراف الدلالي وضبط المخرجات، وصولاً إلى القيود الحسابية ومتطلبات القابلية للتفسير. إضافةً إلى ذلك، تم استعراض الاتجاهات البحثية المستقبلية التي تسعى إلى تطوير وكلاء أكثر استقلالية وكفاءة، وقادرين على التكامل مع مهام إدارية وتقنية متعددة ضمن بيئات تطوير معقدة.

وبهذا العرض، يكون الفصل الأول قد أدى دوره في بناء أساس معرفي ومنهجي واضح، يمهد للانتقال إلى الفصول اللاحقة. حيث سيركّز الفصل التالي على دراسة الأعمال السابقة ذات الصلة، تمهيداً لعرض النظام المقترح، ومنهجية تنفيذه، وآليات تقييم الأداء العملي لوكيل الذكاء الصناعي المطور في إطار هذا المشروع.

2 الفصل الثاني: الدراسة المرجعية

2.1 مقدمة الفصل

يتناول هذا الفصل استعراضًا منهجيًا وتحليليًا للأبحاث والدراسات السابقة التي أسهمت في تشكيل مجال الوكلاء الذكيين (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 أحدثته تقنيات الذكاء الصناعي الحديثة، ولا سيما النماذج اللغوية الكبيرة والتعلم العميق، والتي أعادت تعريف حدود الممكن في تصميم أنظمة دعم القرار الذكية.

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

في هذا السياق، سيتم استعراض المناهج المتبعة في الأدبيات العلمية بشكل نقدي، مع تحليل معمّق للبنى والآليات الرئيسية التي هيمنت على أبحاث الوكلاء الذكيين، 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 المعزّز بالاسترجاع (RAG) التي مكّنت هذه الوكلاء من الاستناد إلى معرفة خارجية منظمة أثناء عملية التخطيط واتخاذ القرار.

وسيقترن 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 بالكلفة الحسابية وقابلية التفسير في الأنظمة الذكية المعقّدة.

علاوةً على ذلك، ونظرًا لأن تقييم الأداء يُعد حجر الزاوية في البحث العلمي، سيتم التركيز على أساليب تقييم أنظمة الوكلاء الذكيين في الأدبيات السابقة. ويشمل ذلك مناقشة المعايير المستخدمة لقياس جودة القرارات المولّدة، 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 نقدي ومتكامل لا يقتصر على سرد الدراسات السابقة، بل يسعى إلى استخلاص الدروس المستفادة منها، ورصد الفجوات البحثية التي لا تزال قائمة في مجال الوكلاء الذكيين 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 صناعي أكثر موثوقية، وقابلية للتفسير، وملاءمة للتطبيق في البيئات البرمجية الواقعية.

2.2 تصنيف تقنيات بناء الوكلاء الذكيين (AI Agents)

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

2.2.1 الوكلاء المعتمدون على القواعد (Rule-Based Agents)

تُعد هذه الفئة من أقدم وأبسط أنواع الوكلاء الذكيين، حيث تعتمد على مجموعة من القواعد المنطقية المحددة مسبقًا لاتخاذ القرار. يعمل الوكيل 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 جوهرية، أبرزها ضعف المرونة وعدم القدرة على التكيف مع السيناريوهات غير المتوقعة أو المدخلات غير المهيكلية. كما أن هذا النوع من الوكلاء غير قادر على التعامل مع الغموض أو استنتاج معلومات غير مصرح بها صراحة، مما يجعله غير مناسب لإدارة المشاريع البرمجية المعقدة.

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

تهدف هذه الفئة إلى تجاوز محدودية الوكلاء القائمين على القواعد، من خلال بناء تمثيل داخلي للحالة أو البيئة التي يعمل فيها الوكيل. يعتمد هذا النوع من الوكلاء 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 تعقيدًا مقارنة بالأنظمة القائمة على القواعد، إلا أنها تتطلب نماذج دقيقة ومحددة للبيئة، وتعتمد بشكل كبير على جودة الافتراضات المسبقة. كما أن توسعتها لتشمل بيانات غنية بالنصوص غير المهيكلية، مثل أوصاف المشاريع البرمجية، يُعد أمرًا محدود الفاعلية.

2.2.3 الوكلاء المعتمدون على التعلّم (Learning-Based Agents)

تمثل هذه الفئة النقلة النوعية في مجال الوكلاء الذكيين، وهي النهج الأكثر انتشارًا وفاعلية في الوقت الراهن. بدلًا من الاعتماد على قواعد أو نماذج ثابتة، تعتمد هذه الوكلاء على تعلّم السلوك واتخاذ القرار من البيانات، باستخدام تقنيات التعلّم الآلي والتعلّم العميق.

يمكن تقسيم هذا النهج تاريخيًا إلى:

- **الوكلاء المعتمدون على التعلّم الإحصائي:**
الذين يعتمدون على نماذج احتمالية وخوارزميات تعلّم تقليدية لاستخلاص الأنماط من البيانات.
 - **الوكلاء المعتمدون على التعلّم العميق:**
الذين يستخدمون الشبكات العصبية العميقة لاكتساب تمثيلات عالية المستوى للبيئة، مما مكنهم من التعامل مع بيانات معقدة وغير مهيكلة مثل النصوص.
- وتُعد هذه الفئة الأساس الذي بُنيت عليه الوكلاء الذكيون المعتمدون على النماذج اللغوية الكبيرة، والتي تشكّل محور هذا البحث.

2.3 النماذج اللغوية الكبيرة في بناء الوكلاء الذكيين

أحدثت النماذج اللغوية الكبيرة ثورة حقيقية في تصميم الوكلاء الذكيين، حيث تجاوزت قدرتها حدود التفاعل النصي البسيط إلى الفهم الدلالي، والاستدلال، والتخطيط. تعتمد هذه النماذج على تعلّم تمثيلات لغوية عميقة من كميات ضخمة من البيانات النصية، مما يتيح لها تحليل السياق واستخلاص المعاني الضمنية.

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

تُعد النماذج اللغوية الكبيرة المكوّن الإدراكي الأساسي في هذا النوع من الوكلاء، 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 وكيّل ذكي مقيّد يحدد الأهداف، وينظم عملية اتخاذ القرار.

2.3.2 الوكلاء المعزّزون بالمعرفة (RAG-Enhanced Agents)

يمثل دمج آلية التوليد المعزّز بالاسترجاع (Retrieval-Augmented Generation – RAG) خطوة محورية في تطوير الوكلاء الذكيين. تتيح هذه الآلية للوكيل الوصول إلى معرفة خارجية منظمة، مثل مفاهيم Agile و Scrum، 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 موجهة.

2.3.3 الوكلاء القائمون على آليات الانتباه والتخطيط السياقي

تستفيد الوكلاء 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 يعزز جودة التخطيط واستنتاج المخاطر.

2.4 مقاييس تقييم أداء الوكيل الذكي وجودة المخرجات

يُعد تقييم أداء الوكلاء 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 صحّة 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 التوليدي 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 الوكلاء المعتمدين على النماذج اللغوية الكبيرة. يقيس هذا المعيار مدى التزام الوكيل بالمحتوى الفعلي لوصف المشروع البرمجي المُدخل، 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 ضمناً في وصف المشروع.
- قياس نسبة العناصر الصحيحة مقابل العناصر المُختلفة أو غير المدعومة بالنص الأصلي.

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

2.4.2 جودة التحليل البنيوي للمشروع (Structural Analysis Quality)

يركّز 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، والخطة الزمنية المقترحة.

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

2.4.3 دقة التخطيط الزمني وإدارة المخاطر

نظرًا لأن الوكيل يحاكي دور مدير مشروع بشري، فإن جودة التخطيط تُعد معيارًا أساسيًا للتقييم. يركز هذا المقياس على:

- مدى واقعية تقسيم العمل عبر السبرنتات.
- شمولية تحديد المخاطر التقنية والتنظيمية.
- منطقية استراتيجيات التخفيف (Mitigation Strategies) المقترحة.

يتم تقييم هذا الجانب من خلال المقارنة مع مبادئ Agile و Scrum المعتمدة، سواء بشكل يدوي 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.

يركّز هذا المقياس على:

- مدى اعتماد المخرجات على المعرفة المسترجعة بدلًا 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 تحوّل الوكيل من نموذج لغوي عام إلى نظام دعم قرار معرفي.

2.4.5 قابلية التفسير والوضوح الإداري (Explainability & Managerial Readability)

نظرًا لأن المخرجات موجّهة للاستخدام البشري، فإن قابلية الفهم والتفسير تُعد عنصرًا أساسيًا 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 أعدّها مديرو 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 الذكيين والنماذج اللغوية الكبيرة، يركّز هذا القسم على تحليل مجموعة من الدراسات المرجعية التي شكّلت نقاط تحوّل 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:

أتمتة تخطيط المشاريع البرمجية، تحليل المستودعات البرمجية، دعم منهجيات Agile و Scrum، دمج آليات الاسترجاع المعرفي (RAG)، وتقييم موثوقية المخرجات الناتجة عن النماذج اللغوية. ويهدف 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، وغطت 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.

أهميتها 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:

تتناول 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 مديرو مشاريع بشريون، مع تقليل زمن إعداد الخطة بشكل ملحوظ.

أهميتها بالنسبة لهذا المشروع:

تمثل هذه الدراسة الأساس المباشر لوظيفة الوكيل المقترح، حيث تؤكد 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 نظامًا قائمًا 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.

أهميتها 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 التوسع مستقبلاً 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 بالنماذج اللغوية الكبيرة في تحسين إدارة المشاريع وفق منهجيات Agile و Scrum. طوّر الباحثون نظامًا أوليًا قادرًا على إدارة الاجتماعات، تتبع تقدم المهام، وتوليد تقارير السبرنت، مع تكامل مباشر مع أدوات مثل Jira.

أظهرت النتائج تحسّنًا ملحوظًا في كفاءة فرق العمل وتقليل الزمن المستغرق في الأنشطة الإدارية المتكررة.

أهميتها 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.

أهميتها بالنسبة لهذا المشروع:

تُعد هذه الدراسة الداعم العلمي الأقوى لمرحلة RAG في المشروع المقترح، 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 وتقارير 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 صناعي واحد لإدارة المشاريع البرمجية، يركّز على:

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

ويمثّل هذا التوجه مساهمة علمية وتطبيقية ضمن مجال الوكلاء الذكيين المعتمدين على النماذج اللغوية الكبيرة في بيئات هندسة البرمجيات.

3 الفصل الثالث: المنهجية المعتمدة

3.1 مقدمة الفصل

بعد عرض القاعدة المفاهيمية والمنهجية المرتبطة بالوكلاء الأذكاء (Artificial Intelligence Agents)، والنماذج اللغوية الكبيرة (Large Language Models)، ومنهجيات إدارة المشاريع الرشيقة (Agile & Scrum) في الفصول السابقة، يركّز هذا الفصل على تقديم المنهجية التنفيذية والتقنية المعتمدة في تصميم وتطوير النظام المقترح.

يتمثل النظام في وكيل ذكاء صناعي ذاتي (Autonomous AI Agent) صُمم ليعمل كمنظومة دعم قرار مخصصة لإدارة المشاريع البرمجية، حيث يقوم بتحليل أوصاف المشاريع المكتوبة بصيغة نصوص غير مهيكلة، ثم معالجتها وتحويلها إلى مخرجات منظمة وقابلة للتطبيق العملي. وتشمل هذه المخرجات تحديد هدف المشروع، والتقنيات المتوقعة، والمكونات الأساسية، وتحليل المخاطر، إضافةً إلى بناء خطة تنفيذية مقسّمة إلى سباقات عمل (Sprints) متوافقة مع مبادئ إطار Scrum.

يركّز هذا الفصل على توضيح البنية الكاملة للنظام من منظور هندسي، بدءًا من اختيار الأدوات البرمجية والنماذج المستخدمة، مرورًا بآليات التكامل فيما بينها ضمن بنية موحّدة، وصولًا إلى شرح منطق اتخاذ القرار داخل الوكيل الذكي. كما يتناول الفصل آليات الضبط والتحقق من صحة المخرجات، وطرق تقييد النموذج بالسياق والمعرفة، بالإضافة إلى أساليب التقييم الذاتي لضمان جودة النتائج واتساقها.

ويهدف هذا العرض المنهجي إلى توفير توصيف دقيق وقابل للتحقق لجميع القرارات التصميمية والتقنية المتخذة، بما يضمن الشفافية، والقابلية لإعادة التشغيل، وإمكانية الدفاع الأكاديمي عن النظام المقترح من حيث بنيته، ووظيفته، وأدائه العملي.

3.2 بيئة التطوير والأدوات البرمجية المستخدمة

تم تطوير النظام المقترح ضمن بيئة تطوير برمجية تعتمد على لغة Python، وذلك استنادًا إلى ملاءمتها العالية لتطبيقات هندسة الذكاء الاصطناعي، وقدرتها على دعم التكامل بين النماذج اللغوية الكبيرة، وتقنيات الاسترجاع المعرفي، وآليات التحقق النبوي للمخرجات. ويأتي اختيار هذه البيئة نتيجة لمرونتها، وانتشارها الواسع في الأبحاث والتطبيقات العملية المرتبطة بالذكاء الصناعي، إضافةً إلى دعمها القوي للتنفيذ المحلي للنماذج دون الاعتماد على خدمات سحابية خارجية.

توفّر بيئة Python إطارًا مناسبًا لبناء أنظمة هجينة تجمع بين المعالجة الدلالية للنصوص، واتخاذ القرار المنهجي، وضبط المخرجات وفق مخططات محددة، وهو ما يتوافق مع متطلبات النظام المقترح القائم على وكيل ذكاء صناعي ذاتي لإدارة المشاريع البرمجية.

3.2.1 لغة البرمجة

تم اعتماد لغة Python كلغة البرمجة الأساسية لتنفيذ النظام، نظرًا لما تتمتع به من خصائص تجعلها مناسبة لتطوير أنظمة ذكاء صناعي معقدة وقابلة للتوسع. ويمكن تلخيص أسباب هذا الاختيار فيما يلي:

- دعم فعال لمفاهيم البرمجة الكائنية والهيكلية، مما يسهل بناء بنية معيارية للنظام وتقسيمه إلى مكونات مستقلة وقابلة لإعادة الاستخدام.
- توفر منظومة واسعة من المكتبات المتخصصة في معالجة اللغة الطبيعية، والاسترجاع المعلوماتي، وتحليل النصوص، وهو ما يشكل أساسًا لمعالجة أوصاف المشاريع البرمجية غير المهيكلة.
- سهولة التعامل مع البيانات المهيكلة بصيغة JSON ، الأمر الذي يدعم تطبيق آليات التحقق البنيوي (Schema Validation) وضبط المخرجات بشكل صارم.
- توافق مباشر مع مكتبات تشغيل النماذج اللغوية الكبيرة محليًا، مما يتيح التحكم الكامل ببيئة التشغيل، ويعزز من القابلية لإعادة التنفيذ والاستقلالية عن الخدمات الخارجية.

وبناءً على ما سبق، يوفر استخدام لغة Python أساسًا تقنيًا متينًا لتطوير النظام المقترح، ويضمن تحقيق التوازن بين المرونة البرمجية، والدقة المنهجية، وقابلية التوسع والتقييم الأكاديمي.

3.3 تشغيل النموذج اللغوي الكبير محليًا:

يعتمد النظام المقترح على تشغيل نموذج لغوي كبير ضمن بيئة محلية (Local Execution Environment) ، دون الاعتماد على واجهات برمجية سحابية أو خدمات خارجية. يندرج هذا الخيار ضمن التوجهات الحديثة في هندسة أنظمة الذكاء الاصطناعي التي تسعى إلى تعزيز السيطرة على عملية التوليد، وضمان خصوصية البيانات، وتحقيق قابلية أعلى للتحليل والتقييم المنهجي لسلوك النماذج المستخدمة. كما يساهم هذا النهج في تحسين القابلية لإعادة التشغيل، وهو عامل أساسي في السياقات البحثية والأكاديمية.

3.3.1 مكتبة llama_cpp

تم استخدام مكتبة **llama_cpp** لتشغيل النموذج اللغوي الكبير محليًا، نظرًا لقدرتها على تحميل وتشغيل النماذج المحولة إلى صيغة GGUF بكفاءة عالية ضمن بيانات محدودة الموارد نسبيًا. ويحقق هذا الاختيار مجموعة من الأهداف البحثية والتقنية المهمة، من أبرزها:

- إتاحة التحكم الكامل في عملية التوليد اللغوي، بما في ذلك ضبط طول السياق، ومعاملات العشوائية، وآليات التقيد البنيوي للمخرجات.
- ضمان خصوصية البيانات، حيث تتم معالجة أوصاف المشاريع النصية بالكامل ضمن البيئة المحلية دون إرسال أي بيانات إلى أطراف خارجية.
- تقليل الاعتماد على الخدمات السحابية وواجهات البرمجة التجارية، مما يعزز استقلالية النظام وقابليته للاستخدام في بيئات تعليمية أو بحثية مغلقة.
- تمكين تحليل سلوك النموذج اللغوي بدقة، من خلال مراقبة المخرجات، وأزمنة التوليد، وتأثير تغيير المعاملات على جودة النتائج.

وبذلك تشكل مكتبة llama_cpp عنصرًا محوريًا في تحقيق المتطلبات الهندسية للنظام المقترح، ولا سيما فيما يتعلق بالتحكم، والشفافية، والقابلية للتقييم.

3.3.2 النموذج المستخدم

تم اعتماد النموذج **Qwen2.5-7B-Instruct** بصيغة **GGUF** كنواة لغوية للنظام، وهو نموذج لغوي كبير موجّه للتعليمات (Instruction-Tuned)، ما يجعله ملائمًا لأداء مهام تتطلب الالتزام بتعليمات محددة، مثل التخطيط المنهجي، واتخاذ القرار، وتوليد مخرجات منظمة، بدلًا من التوليد اللغوي الحر غير المقيد.

تم اختيار هذا النموذج لقدرته على الاستجابة الدقيقة للتعليمات المعقدة، وتوليد مخرجات متسقة بنيويًا عند دمجها مع آليات التقيد البنيوي والتحقق الصارم للمخرجات المعتمدة في النظام.

ولضمان استقرار الأداء وتوازن جودة النتائج، تم ضبط النموذج باستخدام مجموعة من المعاملات التجريبية المدروسة، كما يلي:

- **n_ctx = 4096**

لتحديد الحد الأقصى لطول السياق المدخل إلى النموذج، بما يسمح بدمج وصف المشروع، وسياق الاسترجاع المعرفي، وتعليمات التوليد ضمن نافذة سياقية واحدة.

• **temperature = 0.2**

لتقليل درجة العشوائية في التوليد، وتعزيز الاتساق والدقة في المخرجات، وهو أمر ضروري في تطبيقات التخطيط واتخاذ القرار.

• **top_p = 0.85**

لتحقيق توازن مدروس بين تنوع المخرجات ودقتها، مع الحفاظ على جودة لغوية ومنهجية مقبولة.

• **seed = 42**

لضمان قابلية إعادة إنتاج النتائج (Reproducibility)، وهو شرط أساسي في التقييم الأكاديمي والتحقق التجريبي.

يساهم هذا الإعداد في توفير بيئة تشغيل مستقرة وقابلة للتحكم، ويشكل الأساس اللغوي الذي يعتمد عليه وكيل الذكاء الصناعي في تنفيذ مهامه التخطيطية والإدارية ضمن النظام المقترح.

3.4 إدارة إعدادات الوكيل باستخدام كائن AgentConfig:

اعتمد النظام المقترح على تصميم كائن إعدادات مخصص للوكيل الذكي، يُشار إليه باسم **AgentConfig**، وذلك لتجميع وضبط جميع المعاملات التشغيلية والتجريبية المرتبطة بسلوك وكيل الذكاء الصناعي ضمن بنية موحدة. تم تنفيذ هذا الكائن باستخدام آلية **dataclass** في لغة Python، بما ينسجم مع مبادئ التصميم البرمجي المعياري وفصل الاهتمامات (Separation of Concerns).

يتضمن كائن **AgentConfig** مجموعة من المعاملات التي تتحكم مباشرة في آلية عمل الوكيل، وتشمل إعدادات تشغيل النموذج اللغوي الكبير، ومعاملات التوليد، وحدود الاسترجاع المعرفي، وعدد محاولات إعادة التوليد والإصلاح، إضافةً إلى إعدادات التتبع والتقييم الزمني لسير التنفيذ.

يساهم هذا التصميم في تحقيق ضبط منهجي لسلوك الوكيل الذكي، ويتيح تعديل الإعدادات التجريبية بشكل صريح ومنفصل عن منطق التنفيذ الأساسي. كما يوفر إطارًا واضحًا لإدارة التجارب المقارنة، من خلال تغيير قيم الإعدادات وملاحظة تأثيرها على جودة المخرجات دون المساس ببنية النظام.

ومن الناحية الأكاديمية، يعزز استخدام كائن **AgentConfig** من الشفافية وقابلية التتبع، حيث تصبح جميع القرارات التجريبية والمعاملات المستخدمة موثقة بشكل مباشر داخل بنية النظام، مما يدعم قابلية إعادة التنفيذ (Reproducibility) ويسهل عملية التقييم والتحقق العلمي من النتائج.

وبذلك لا يقتصر دور كائن **AgentConfig** على كونه أداة ضبط تقنية، بل يشكل عنصرًا منهجيًا أساسيًا في تصميم وكيل الذكاء الصناعي، يضمن الانضباط التجريبي والوضوح الهندسي للنظام المقترح.

3.5 قاعدة المعرفة وإدارة السياق المعرفي:

اعتمد النظام المقترح في مرحلته الأولى على تحليل التوصيف النصي للمشروع البرمجي بوصفه المدخل الأساسي الذي يحدّد سياق المشكلة ومتطلبات الحل. حيث يتم تمرير وصف المشروع كنص غير مهيكّل يتضمّن الأهداف العامة، والقيود الزمنية والتنظيمية، والمتطلبات الوظيفية، إضافةً إلى الإشارة الصريحة إلى اعتماد منهجيات Agile و Scrum في إدارة التنفيذ.

وبالرغم من أن النماذج اللغوية الكبيرة تمتلك معرفة عامة مسبقة بمنهجيات Agile و Scrum نتيجة تدريبها على نطاق واسع من النصوص التقنية والإدارية، فقد أظهرت التجارب الأولية أن الاعتماد على التوليد اللغوي الحر دون تقييد سياقي واضح يؤدي إلى زيادة احتمالية الهلوسة الدلالية، وإلى إنتاج مخرجات تتسم بالتعميم أو بعدم الاتساق مع طبيعة المشروع الفعلي وقيوده الواقعية. ويُعزى ذلك إلى أن المعرفة العامة للنموذج، في غياب توجيه سياقي صريح، قد لا تكون كافية لضمان تطبيق منهجي دقيق لمبادئ إدارة المشاريع الرشيقة ضمن سياق محدد.

وبناءً على ذلك، تم اعتماد نهج قائم على توجيه المعرفة العامة للنموذج وربطها بسياق المشروع المحدد، من خلال تقييد عملية التوليد بالمعلومات المستخلصة مباشرة من توصيف المشروع، بدل الاعتماد على توليد عام غير مرتبط بمصدر معرفي واضح. ويهدف هذا التقييد إلى تحقيق توازن منهجي بين الاستفادة من قدرات النموذج اللغوي الكبيرة، وضمان اتساق المخرجات مع متطلبات المشروع الفعلية ومبادئ Agile و Scrum المعتمدة.

3.5.1 قاعدة المعرفة النصية للمشروع (Project Textual Knowledge Base)

تمثّل قاعدة المعرفة في النظام المقترح ملفاً نصياً يحتوي على توصيف المشروع البرمجي ومتطلباته الأساسية، ويُستخدم هذا الملف كمصدر سياقي موجّه يتم تحميله ومعالجته باستخدام آلية الاسترجاع المعرفي (Retrieval-Augmented Generation – RAG). ويتم تقسيم النص إلى مقاطع معرفية (Knowledge Chunks)، بحيث يُمثّل كل مقطع كوحدة مستقلة تتضمن:

- معرفاً فريداً (Document ID) يتيح الإشارة الدقيقة إلى المقطع.
- محتوى نصي يعبر عن جزء محدد من توصيف المشروع أو أحد قيوده أو متطلباته.

يتيح هذا التمثيل البنوي ربط قرارات الوكيل الذكي — مثل تحديد أهداف المشروع، والتقنيات المتوقعة، وتحليل المخاطر، وبناء خطة السبرنت — بمقاطع نصية محددة من توصيف المشروع. وبذلك تصبح المخرجات ناتجة عن تحليل موجّه للسياق المعرفي، وليس عن توليد لغوي حر غير مقيّد.

كما يساهم هذا النهج في تعزيز قابلية التفسير والتتبع، حيث يصبح من الممكن إرجاع كل قرار إداري أو تخطيطي إلى جزء محدد من توصيف المشروع النصي الذي استند إليه النظام أثناء التوليد. ويتوافق ذلك مع متطلبات الشفافية وقابلية الدفاع الأكاديمي عن النظام المقترح، ولا سيما في سياقات التقييم والتحقق العلمي.

وبناءً على ما سبق، لا تعمل قاعدة المعرفة في هذا النظام كمرجع نظري مستقل لمفاهيم Agile و Scrum، وإنما كآلية لتوجيه المعرفة العامة المتضمنة في النموذج اللغوي وربطها بسياق المشروع المحدد. ويتم دمج هذا السياق المعرفي لاحقاً مع منطق الوكيل وآليات التقييد البنيوي والتحقق، بما يضمن إنتاج مخرجات متوافقة مع مبادئ Agile و Scrum بصورة منضبطة وقابلة للتحقق.

3.6 آلية الاسترجاع المعرفي (Retrieval-Augmented Generation)

يعتمد النظام المقترح على آلية استرجاع معرفي مدمجة ضمن بنية الوكيل الذكي، مصممة لتنظيم استخدام التوصيف النصي للمشروع والتحكم في الأجزاء التي يتم الاعتماد عليها أثناء عملية التوليد. وفي هذا الإطار، تم اعتماد نهج الاسترجاع المعرفي ذاتي الارتكاز (Self-Grounded RAG)، حيث يُستخدم نفس ملف توصيف المشروع البرمجي كمصدر للمعرفة وكمدخل سياقي في الوقت ذاته.

يُحمل توصيف المشروع النصي مرة واحدة عند تهيئة النظام، ثم تتم معالجته وتقسيمه إلى وحدات نصية منظمة تُستخدم لاحقاً خلال مختلف مراحل تحليل المشروع واتخاذ القرار. ولا تهدف آلية الاسترجاع في هذا النظام إلى إدخال مصادر معرفية خارجية أو توسيع نطاق معرفة النموذج، وإنما إلى تنظيم الوصول إلى محتوى توصيف المشروع نفسه، عبر اختيار المقاطع الأكثر ارتباطاً بسياق المهمة الحالية وتمريرها إلى النموذج اللغوي ضمن سياق مضبوط.

يساهم هذا النهج في تقليل الاعتماد على التوليد العام غير الموجه، ويعزز اتساق المخرجات مع متطلبات المشروع الفعلية وقيوده، مع الحفاظ على قابلية التتبع والتحليل لكل قرار يتم اتخاذه داخل النظام.

3.6.1 بنية الاسترجاع المعرفي في النظام:

تم تصميم مكوّن الاسترجاع المعرفي في النظام ليعمل وفق نمطين مختلفين يمكن التبديل بينهما عبر إعدادات التشغيل، وتشمل هاتان الطريقتان:

- الاسترجاع القائم على التشابه اللفظي باستخدام تقنية TF-IDF .
- الاسترجاع القائم على التشابه الدلالي باستخدام نماذج تمثيل الجمل (Sentence Embeddings) .

يؤقّر هذا التصميم مرونة منهجية تسمح بتقييم تأثير كل أسلوب استرجاع على جودة المخرجات، كما يتيح اختيار آلية خفيفة وقابلة للتفسير أو آلية دلالية أعمق، تبعًا لطبيعة التجربة ومتطلبات التحليل.

3.6.2 الاسترجاع القائم على TF-IDF باستخدام مكتبة scikit-learn

في نمط الاسترجاع القائم على TF-IDF ، تم استخدام مكتبة **scikit-learn** لتنفيذ تحويل النصوص إلى تمثيلات عددية وحساب درجات التشابه. ويعتمد هذا النمط على الأدوات التالية:

- **TfidfVectorizer**

لتحويل المقاطع النصية المستخلصة من توصيف المشروع إلى متجهات عددية تعبّر عن الأهمية النسبية للمصطلحات داخل كل مقطع مقارنةً ببقية المقاطع النصية.

- **cosine_similarity**

لحساب درجة التشابه بين تمثيل الاستعلام الحالي وتمثيلات المقاطع النصية، وتحديد مستوى الارتباط بينها.

يتميّز هذا الأسلوب بكونه بسيطاً من حيث البنية، قابلاً للتفسير، وخفيفاً من حيث المتطلبات الحسابية، مما يجعله مناسباً لتحليل العلاقة اللفظية المباشرة بين نص المهمة ومحتوى توصيف المشروع.

3.6.3 آلية الاسترجاع القائم على التمثيل الدلالي باستخدام Sentence Transformers بعد حساب

التشابه:

إضافةً إلى TF-IDF ، يدعم النظام نمطًا بديلًا للاسترجاع يعتمد على التمثيل الدلالي للنصوص باستخدام نماذج **Sentence Transformers**. في هذا النمط، يتم تحويل المقاطع النصية لتوصيف المشروع إلى متجهات دلالية عالية الأبعاد تعبر عن المعنى العام للنصوص، بدل الاعتماد على التكرار اللفظي للكلمات فقط.

يتم حساب درجة التشابه بين الاستعلام والمقاطع النصية باستخدام الضرب الداخلي (Dot Product) بين المتجهات بعد تطبيعها، وهو ما يعادل حساب التشابه الكوني (Cosine Similarity) في هذا السياق. يتيح هذا الأسلوب التقاط علاقات دلالية أعمق بين النصوص، حتى في حال اختلاف الصياغة أو المفردات المستخدمة.

يوفر هذا النمط قدرة أعلى على استرجاع مقاطع نصية مرتبطة بالمعنى العام للمهمة، ويستخدم في الحالات التي تتطلب فهمًا دلاليًا أوسع لتوصيف المشروع، مع الحفاظ على الارتباط الحصري بمحتوى المشروع النصي ذاته.

3.7 هندسة الموجه (Prompt Engineering)

تمثل هندسة الموجه أحد العناصر الجوهرية في النظام المقترح، حيث تلعب دورًا حاسمًا في ضبط سلوك النموذج اللغوي الكبير وتحويله من مولّد نصوص عام إلى وكيل ذكي موجه لأداء مهام تخطيط وإدارة مشاريع برمجية بشكل منهجي. وقد تم تصميم الموجه النهائي ليعكس بنية الوكيل الذكي، ويتكامل مع آليات الاسترجاع المعرفي والتقييد البنيوي لضمان اتساق المخرجات وقابليتها للتحقق.

يعتمد تصميم الموجه في النظام المقترح على عدة مكونات مترابطة، يتم دمجها ضمن نص توجيهي واحد يُعاد استخدامه وتعديله بشكل منهجي أثناء مراحل التوليد المختلفة.

3.7.1 تعريف دور النظام

في بداية الموجّه، يتم تحديد دور النموذج بشكل صريح على أنه مدير مشاريع برمجية خبير يمتلك معرفة بمنهجيات Agile وScrum، وقادر على تحليل توصيفات المشاريع واتخاذ قرارات تخطيطية مبنية على هذا التحليل. يهدف هذا التحديد إلى تقييد منظور النموذج ومنع إنتاج مخرجات لا تتوافق مع الدور الوظيفي المطلوب، مثل الإجابات العامة أو السرد غير المرتبط بإدارة المشاريع.

3.7.2 تعليمات التحكّم السلوكي

يتضمّن الموجّه مجموعة من التعليمات السلوكية التي توجّه النموذج إلى:

- اتخاذ قرارات تخطيطية واضحة بدل الاكتفاء بالوصف.
- تبرير هذه القرارات ضمن حدود السياق متاح.
- مراعاة القيود الواقعية للمشروع، مثل حجم الفريق والإطار الزمني.
- الالتزام بمنهجية منظمة عند تقسيم العمل وتحليل المخاطر.

تُستخدم هذه التعليمات للحد من النزعة التوليدية الحرة للنموذج، ودفعه نحو إنتاج مخرجات ذات طابع تحليلي ومنهجي، تتوافق مع طبيعة دور الوكيل الذكي.

3.7.3 دمج السياق المعرفي المسترجع

يتم تضمين المقاطع النصّية المسترجعة عبر آلية الاسترجاع المعرفي ضمن الموجّه بوصفها سياقًا معرفيًا موجّهًا. ويُفرض على النموذج استخدام هذا السياق كمصدر أساسي للتحليل، دون افتراض معلومات غير مدعومة به. يساهم هذا الدمج في ربط عملية التوليد مباشرة بمحتوى توصيف المشروع، ويحدّ من الانحرافات الدلالية أو التعميم غير المبرّر.

3.8 معالجة المخرجات والتحقق البنيوي:

تُعد معالجة مخرجات النموذج والتحقق البنيوي منها مرحلة حاسمة في النظام المقترح، إذ تهدف إلى ضمان أن تكون النتائج المتولدة قابلة للاستخدام كخطط تنفيذية فعلية، وليس مجرد مخرجات لغوية غير منضبطة. وبما أن النماذج اللغوية الكبيرة قد تنتج أحيانًا نصوصًا تحتوي على انحرافات شكلية أو بنيوية، فقد تم تصميم سلسلة من آليات المعالجة والتحقق الصارمة لضبط هذه المخرجات قبل اعتمادها داخل منطق الوكيل الذكي.

3.8.1 توليد واستخراج JSON المقيّد بنحو رسمي

يعتمد النظام على توليد المخرجات بصيغة JSON مقيّدة مسبقًا، وذلك من خلال استخدام قواعد نحوية رسمية (Grammar-Based Constraints) مشتقة مباشرة من مخطط البيانات المعتمد للنظام. ويتم تمرير هذا المخطط إلى النموذج اللغوي أثناء عملية التوليد، مما يفرض عليه الالتزام ببنية JSON محددة منذ مرحلة التوليد نفسها، بدل الاكتفاء بالتحقق اللاحق.

يساهم هذا الأسلوب في الحد من الأخطاء الشائعة المرتبطة بمخرجات النماذج اللغوية، مثل:

- إدراج نصوص تفسيرية خارج البنية المطلوبة.
- توليد تنسيقات JSON غير مكتملة أو غير صالحة.
- كسر البنية الهرمية المتوقعة للمخرجات.

وبذلك تصبح عملية استخراج JSON عملية مباشرة، حيث يتم التعامل مع ناتج التوليد بوصفه كيانًا منظمًا، وليس نصًا يحتاج إلى تنظيف أو قصّ يدوي.

3.8.2 التحقق البنيوي باستخدام مخطط البيانات (Schema Validation)

بعد توليد المخرجات بصيغة JSON ، يخضع الناتج لمرحلة تحقق بنيوي صارمة باستخدام مخطط بيانات معرّف مسبقًا، يمثل المواصفات الرسمية لبنية الخرج المتوقع من الوكيل الذكي. ويشمل هذا التحقق التأكد من:

- وجود جميع الحقول الإلزامية ضمن الخرج.
- توافق أنواع البيانات والقيم مع القيود المحددة (مثل حدود القيم العددية).
- صحة عدد السبرنتات وبنيتها الداخلية.
- احتواء كل سبرنت على مهام محددة ومعايير قبول واضحة.
- وجود عناصر التحليل الداعم، مثل المخاطر، ومعايير التقييم، والمصادر المعرفية عند الاقتضاء.

تمنع هذه المرحلة تمرير أي مخرجات لغوية غير مكتملة أو غير متوافقة بنيويًا على أنها خطط تنفيذية صحيحة، وتحول عملية التوليد إلى عملية مقيدة بقواعد تحقق رسمية.

3.8.3 آليات الإكمال القسري وإعادة التوليد المنضبط

في حال اجتياز المخرجات التحقق البنيوي الأساسي مع وجود نواقص منطقية (مثل غياب عناصر إلزامية أو حقول فارغة)، يعتمد النظام آلية إكمال منضبطة (Completion Passes) يتم فيها إعادة توجيه النموذج لتعبئة الأجزاء الناقصة فقط، مع الحفاظ على المحتوى الصحيح الموجود مسبقًا.

كما يتضمن النظام آلية إكمال قسري خاصة ببنية خطة التنفيذ، تضمن عدم قبول أي مخرج يخلو من خطة سبرنتات مكتملة. وفي هذه الحالة، يتم تفعيل موجه مخصص يُجبر النموذج على توليد خطة تتضمن عددًا ثابتًا من السبرنتات، مع أهداف ومهام ومعايير قبول لكل سبرنت، دون تعديل بقية مكونات الخرج.

تضمن هذه الآليات أن يكون الخرج النهائي:

- مكتملاً بنيويًا.
- متماسكًا منطقيًا.
- قابلاً للاستخدام الفعلي ضمن سياق إدارة المشاريع.

3.8.4 الأثر المنهجي للتحقق البنيوي

من خلال دمج التوليد المقيّد نحوياً، والتحقق البنيوي الصارم، وآليات الإكمال المنضبط، يتحول النموذج اللغوي الكبير من أداة توليد نصوص إلى مكوّن ضمن نظام هندسي محكوم بقواعد واضحة. وتمنع هذه المنهجية تمرير مخرجات لغوية غير صالحة كخطط تنفيذية، كما ترفع من موثوقية النظام وقابليته للتقييم والدفاع الأكاديمي.

وبذلك تتشكّل معالجة المخرجات والتحقق البنيوي حجر الأساس في ضمان أن يكون الوكيل الذكي المقترح نظام دعم قرار فعلي، وليس مجرد واجهة لغوية متقدمة.

3.9 تتبع الأداء والشفافية

يشتمل النظام المقترح على آلية متكاملة لتتبع الأداء وتوثيق مسار اتخاذ القرار داخل الوكيل الذكي، وذلك بهدف تعزيز الشفافية وقابلية التحليل والتحقق. ويُعد هذا التتبع عنصراً أساسياً في تحويل النظام من نموذج توليدي مغلق إلى منظومة هندسية يمكن تقييم سلوكها وفهم مخرجاتها بصورة منهجية.

يقوم النظام بتسجيل مجموعة من المؤشرات التشغيلية والزمنية أثناء تنفيذ كل طلب، من بينها:

- **زمن التنفيذ**، بما يشمل زمن التوليد وزمن التنفيذ الكلي، مما يتيح تقييم الكلفة الزمنية لكل عملية تخطيط.
- **المقاطع المعرفية المستخدمة**، عبر تسجيل معرفات الوثائق (Document IDs) التي تم استرجاعها واستخدامها ضمن السياق، الأمر الذي يوفّر إمكانية تتبع الأساس النصّي لكل قرار ناتج.
- **عدد محاولات الإصلاح والإكمال**، بما في ذلك عدد مرات تفعيل آليات الإكمال المنضبط أو الإكمال القسري لخطة السبرنت، وهو مؤشر مباشر على استقرار المخرجات وجودتها الأولية.
- **معاملات الضبط المستخدمة**، مثل عتبات التشابه وحدود الاسترجاع، بما يسمح بتحليل تأثير هذه المعاملات على النتائج المتولدة.

تُجمّع هذه المعلومات ضمن سجل تتبّع (Trace) منظم يُعاد مع الخرج النهائي، إلى جانب تقرير تفسيري (Explainability Report) يوضّح المنهجية العامة المتبعة في التوليد والاسترجاع والتحقق. ويسمح هذا السجل بإعادة بناء مسار اتخاذ القرار خطوة بخطوة، بدءاً من استرجاع السياق المعرفي، مروراً بمرحلة التوليد، وانتهاءً بالتحقق البنيوي واعتماد الخرج النهائي.

توفّر هذه المنهجية مستوى عالٍ من الشفافية التشغيلية، حيث لا يُنظر إلى مخرجات الوكيل الذكي على أنها نتائج نهائية غير قابلة للتفسير، بل كحوصلة سلسلة واضحة من القرارات المؤثقة والقابلة للتحليل. ويُعد ذلك عاملاً جوهرياً في دعم قابلية التقييم الأكاديمي للنظام، وفي تمكين استخدامه كنظام دعم قرار موثوق في سياقات إدارة المشاريع البرمجية.

3.10 خاتمة الفصل:

قدّم هذا الفصل عرضًا تفصيليًا للمنهجية المعتمدة في بناء وكيل مدير مشاريع ذكي قائم على النماذج اللغوية الكبيرة، مدعومًا بآلية استرجاع معرفي محدودة، ونظام تحقق بنيوي، وتقييم 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

يمثل هذا الفصل المرحلة التحليلية الحاسمة في هذا المشروع البحثي، حيث يتم فيه عرض ومناقشة النتائج العملية المتحققة من تنفيذ واختبار النظام المقترح، والمتمثل في وكيل ذكي مستقل لإدارة المشاريع البرمجية. ويهدف هذا الفصل إلى تقييم فعالية المنهجية المعتمدة في تحويل توصيف نصي غير مهيكّل لمشروع برمجي إلى خطة تنفيذ منظمة وقابلة للتطبيق، تستند إلى مبادئ Agile و Scrum، وذلك بالاعتماد على نموذج لغوي كبير يعمل محليًا ومدعوم بآليات استرجاع معرفي وتقييد بنيوي.

يركّز هذا الفصل على تحليل أداء النظام من منظور هندسي تطبيقي، من خلال الجمع بين مؤشرات كمية قابلة للقياس، مثل زمن التنفيذ، وعدد محاولات الإكمال والإصلاح البنيوي، ومستوى اكتمال المخرجات، وبين مؤشرات نوعية تتعلق بجودة الخطط المنتجة، واتساقها مع متطلبات المشروع، وقدرتها على تمثيل قيود واقعية مثل حجم الفريق والإطار الزمني.

كما يتم تحليل هذه النتائج في سياق هندسة الوكلاء الذكية (AI Agents)، مع التركيز على كيفية تفاعل مكونات النظام المختلفة مثل الاسترجاع المعرفي، هندسة الموجه، التقييد البنيوي، وآليات التتبع لإنتاج سلوك تخطيطي منضبط وقابل للتفسير. ويهدف هذا التحليل إلى إبراز مدى نجاح النظام في تجاوز حدود التوليد اللغوي العام، والتحول إلى نظام دعم قرار قادر على التخطيط، والتقييم الذاتي، واتخاذ قرارات تخطيطية ضمن قيود محددة وواضحة.

ويمهّد هذا الفصل، من خلال عرض النتائج وتحليلها، لتقديم خلاصة شاملة حول القيمة العلمية والعملية للنظام المقترح، ومدى قابليته للتطوير والتطبيق في سياقات واقعية لإدارة المشاريع البرمجية.

4.2 منهجية التقييم (Evaluation Methodology)

نظرًا لأن النظام المقترح لا يُصنّف كنظام تعلّم إشرافي تقليدي، بل كوكلاء ذكي مستقل (Autonomous AI Agent) قائم على نموذج لغوي كبير ويتخذ قرارات تخطيطية، فقد تم اعتماد منهجية تقييم هندسية-سلوكية بدلًا من المقاييس الإحصائية الشائعة في أنظمة التصنيف أو التنبؤ.

يعتمد هذا الأسلوب على تقييم سلوك الوكيل أثناء التنفيذ، وجودة المخرجات الناتجة عنه، ومدى التزامها بالقيود البنيوية والمنهجية المفروضة، وذلك باستخدام بيانات يتم توليدها وتسجيلها تلقائيًا أثناء تشغيل النظام.

تقوم منهجية التقييم على ثلاث ركائز رئيسية:

1. تقييم تشغيلي كمي (Operational Quantitative Evaluation)
2. تقييم بنيوي للمخرجات (Structural Validation)
3. تحليل تطوّر السلوك بعد التقييم الذاتي (Behavioral Improvement Analysis)

ويتم تنفيذ هذه المراحل استنادًا مباشرة إلى منطق النظام المطبّق في الكود، دون أي تدخل بشري يدوي في عملية التقييم.

4.2.1 التقييم التشغيلي الكمي (Operational Quantitative Evaluation)

يهدف هذا النوع من التقييم إلى قياس الأداء الفعلي للنظام أثناء التشغيل، اعتمادًا على مؤشرات يتم استخراجها من سجل التنفيذ (Execution Trace) الذي يولده النظام تلقائيًا.

زمن التنفيذ الكلي (Runtime)

يُقاس زمن التنفيذ باستخدام مؤقتات برمجية مدمجة في منطق التشغيل الرئيسي، ويشمل جميع مراحل المعالجة، وهي:

- تحميل النموذج اللغوي محليًا
- تنفيذ آلية الاسترجاع المعرفي (RAG)
- توليد الخطة الأولية
- التقييم الذاتي
- عمليات الإصلاح البنيوي (Self-Repair)

تم اعتماد هذا المقياس لكونه يعكس الكلفة الحسابية الحقيقية للنظام، خصوصًا في ظل التشغيل المحلي الكامل باستخدام مكتبة llama.cpp دون أي اعتماد على بنى سحابية.

عدد محاولات الإصلاح الذاتي (Self-Repair Attempts)

يمثل هذا المعيار مؤشرًا مباشرًا على قدرة الوكيل على:

- اكتشاف عدم كفاية المخرجات الأولية
- إعادة توليدها ضمن قيود محددة
- تحسين الجودة دون تدخل خارجي

يتم تسجيل هذا المؤشر تلقائيًا من خلال عدّاد داخلي في منطق الوكيل، مع تحديد حد أعلى للمحاولات، مما يمنع الدخول في حلقات توليد غير منضبطة.

ويُعد هذا المعيار أساسيًا لتقييم درجة الاستقلالية (Autonomy Level) للوكيل.

مستوى الجدوى النهائي (Final Feasibility Assessment)

بعد الانتهاء من آخر دورة تقييم ذاتي، يُصدر الوكيل تصنيفًا تحليليًا لمستوى جدوى الخطة المنتجة. يعتمد هذا التصنيف على تحليل داخلي يأخذ بعين الاعتبار:

- عدد السبرنتات
- حجم المهام
- قيود الفريق الزمني والبشري
- مدى اتساق الخطة مع نطاق المشروع

ويمثل هذا التقييم انتقال النظام من مجرد مولّد نصوص إلى نظام دعم قرار تحليلي.

مؤشر استخدام الاسترجاع المعرفي (RAG Utilization Indicator)

يتم تقييم آلية الاسترجاع المعرفي من خلال:

- عدد المقاطع النصية المسترجعة
- درجة التشابه المحسوبة
- مدى استخدامها فعليًا في تبرير القرارات

ويتم احتساب ذلك باستخدام:

- `TfidfVectorizer` و `cosine_similarity` في النمط اللفظي
- `SentenceTransformer` في النمط الدلالي

ويُعد مؤشرًا على كفاءة ضبط السياق المعرفي.

4.3 التقييم البنيوي للمخرجات (Structural Output Validation)

نظرًا لاعتماد النظام على إخراج منظم بصيغة JSON ، تم اعتماد تقييم بنيوي صارم للتحقق من صلاحية المخرجات.

يعتمد هذا التقييم على:

- التحقق الآلي من مطابقة مخطط البيانات (Schema Validation)
- التأكد من وجود جميع الحقول الإلزامية
- فحص القيود على القيم (مثل Story Points ، مستويات المخاطر)

ويتم تنفيذ هذا التحقق باستخدام نماذج تحقق بنيوية (Schema-Enforced Output) ، مما يمنع تمرير مخرجات لغوية غير صالحة على أنها خطط تنفيذ.

4.4 تحليل السلوك قبل وبعد التقييم الذاتي (Behavioral Analysis)

لإظهار أثر آليات التقييم الذاتي والإصلاح، تم اعتماد مقارنة منهجية بين:

- الخطة الأولية (Baseline Output)
- الخطة النهائية بعد الإصلاح الذاتي

يركّز هذا التحليل على:

- تغيير إدارة النطاق
- تحسّن تحليل المخاطر
- إعادة توزيع المهام عبر السبرنتات

ويمثّل هذا الأسلوب تقييمًا نوعيًا لسلوك الوكيل، وليس فقط لمخرجاته النصّية.

4.5 خلاصة منهجية التقييم

تعكس هذه المنهجية توجهًا هندسيًا قائمًا على تحليل السلوك واتخاذ القرار بدل الاعتماد على مقاييس تعلم آلي تقليدية لا تنطبق على طبيعة النظام. كما تضمن أن تكون جميع نتائج التقييم:

- قابلة للتفسير
- قابلة لإعادة التكرار
- مستندة مباشرة إلى التنفيذ الفعلي للنظام

5 الفصل الخامس: الأعمال المستقبلية والتوصيات

5.1 مقدمة شاملة

أثبت هذا المشروع إمكانية تصميم وتنفيذ وكيل ذكي مستقل (Autonomous AI Agent) يعمل محليًا لدعم تخطيط وإدارة المشاريع البرمجية وفق مبادئ Agile و Scrum، وذلك بالاعتماد على نموذج لغوي كبير (LLM) مقيد بنيويًا، ومدعوم بآلية استرجاع معرفي ذاتي (Self-Grounded RAG)، وطبقة تحقق بنيوي صارمة للمخرجات، إضافة إلى آليات تقييم ذاتي وإصلاح ذاتي.

أظهرت النتائج العملية أن النظام قادر على تحويل توصيف نصي غير مهيكّل إلى خطة تنفيذ منظمة تتضمن الأهداف، المكونات الأساسية، تحليل المخاطر، وتقسيم العمل إلى سبرنتات، مع الحفاظ على قابلية التفسير وتتبع مصادر القرار. ويُعد هذا الإنجاز خطوة متقدمة نحو نقل النماذج اللغوية من دور التوليد النصي العام إلى دور الوكلاء التخطيطيين القائمين على القيود.

ومع ذلك، فإن مجال الوكلاء الذكية لإدارة المشاريع لا يزال في مرحلة بحثية نشطة، ويقدم فرصًا واسعة للتطوير، سواء على مستوى تعميق المعرفة، تحسين آليات الاسترجاع، رفع دقة التقييم الذاتي، تحسين الكفاءة الحسابية، أو تعزيز قابلية التعميم على أنواع مشاريع أكثر تعقيدًا. ويهدف هذا الفصل إلى تقديم خارطة طريق بحثية واضحة لتطوير النظام المقترح نحو نموذج أكثر نضجًا، تفسيرية، وقابلية للتطبيق في بيئات واقعية.

5.2 التوصيات المستقبلية

5.2.1 تطوير قاعدة المعرفة وآلية الاسترجاع المعرفي

(Knowledge Base & Retrieval Enhancement)

تُعد قاعدة المعرفة وآلية الاسترجاع المرافقة لها حجر الأساس في ضبط سلوك الوكيل الذكي وتقليل الاعتماد على التوليد الحر. ومن هذا المنطلق، يمكن اقتراح التوجهات التالية:

توسيع قاعدة المعرفة المنهجية

يوصى بتوسيع نطاق المعرفة لتشمل مفاهيم أكثر عمقاً في إدارة المشاريع البرمجية، مثل:

- تخطيط سعة الفريق (Capacity Planning)
- تفكيك المتطلبات (Epics, User Stories, Backlog Grooming)
- تقدير الجهد باستخدام نماذج منهجية (Story Points, Planning Poker)
- إدارة الاعتماديات بين المهام
- تعريفات موسّعة لـ Definition of Ready و Definition of Done

الانتقال نحو استرجاع هجين (Hybrid Retrieval)

بينما يوفّر TF-IDF تفسيرية عالية، إلا أنه محدود في النقاط التشابه الدلالي العميق. يمكن تحسين ذلك عبر:

- دمج التمثيلات اللفظية مع تمثيلات دلالية (Sentence Embeddings) محلياً
- استخدام آلية إعادة ترتيب النتائج (Re-Ranking)
- تحسين تغطية المعرفة عبر تنظيمها في طبقات مفاهيمية

تعزيز الربط الصارم بين القرار والمصدر (Grounded Decision Making)

يوصى بفرض قيود أشد، بحيث يُربط كل قرار تخطيطي (خطر، اختيار تقني، تقدير زمني) بمقطع معرفي محدد، مما يعزز الشفافية ويخدم هدف التفسيرية المتقدمة.

5.2.2 تحسين جودة المخرجات وآليات التحقق

(Output Quality & Validation)

الانتقال من التحقق البنوي إلى التحقق الدلالي

في حين يضمن التحقق البنوي الحالي سلامة الشكل، يمكن تطويره ليشمل:

- كشف المهام العامة أو غير القابلة للتنفيذ
- التأكد من أن أهداف السبرنت قابلة للقياس
- التحقق من ملائمة التقنيات المقترحة لطبيعة المشروع
- ربط المخاطر فعليًا بقيود الوقت والموارد

بناء مُقيّم خارجي مستقل (External Evaluator)

بدل الاعتماد الكامل على التقييم الذاتي للنموذج، يمكن إدخال طبقة تقييم مستقلة قائمة على:

- قواعد هندسية ثابتة
- Checklists احترافية لمدير مشروع
- كشف التناقضات التخطيطية

يمثل هذا التوجه خطوة مهمة نحو أنظمة هجينة تجمع بين الذكاء التوليدي والمنطق القاعدي.

5.2.3 تطوير آليات التقييم الذاتي والإصلاح الذاتي

(Self-Evaluation & Self-Repair Evolution)

تقييم متعدد المراحل (Multi-Stage Evaluation)

بدل تقييم شامل واحد، يمكن تقسيم التقييم إلى مراحل مستقلة:

1. تقييم النطاق
2. تقييم الجدولة
3. تقييم المخاطر
4. تقييم الملاءمة التقنية

يسمح ذلك بإصلاح موجّه يقلل الكلفة الحسابية ويرفع دقة التحسين.

التحكم في التدهور الناتج عن الإصلاح (Repair Regression Control)

قد يؤدي الإصلاح إلى تحسين جانب على حساب آخر. يوصى بإدخال:

- تتبع نسخ الخطط (Plan Versioning)
- اختيار أفضل نسخة وفق معايير واضحة (Best-Plan Selection)

5.2.4 تحسين الكفاءة والأداء الحسابي

(Efficiency & Performance)

نظرًا لاعتماد النظام على تشغيل محلي كامل للنموذج اللغوي، يمكن تحسين الأداء عبر:

- تقليل عدد استدعاءات النموذج
- دمج بعض مراحل التوليد والتقييم
- تحسين إعدادات التشغيل
- اعتماد التخزين المؤقت (Caching) لنتائج الاسترجاع المتكررة

5.2.5 توسيع نطاق الاستخدام والتطبيق

(Broader Applications)

يمكن توسيع نطاق النظام ليشمل:

- مشاريع ذكاء اصطناعي (ML Pipelines)
- تطبيقات موبايل
- أنظمة إنترنت الأشياء
- مشاريع مؤسسية متعددة الفرق

كما يمكن إضافة مخرجات احترافية مثل:

- Backlog تفصيلي
- مصفوفة مخاطر (Likelihood / Impact)
- Roadmaps متعددة المراحل
- Deliverables رسمية لكل Sprint

5.3 الخلاصة

قدّم هذا المشروع إطارًا هندسيًا متينًا لبناء وكيل ذكي مستقل لتخطيط المشاريع البرمجية، يجمع بين التشغيل المحلي، الاسترجاع المعرفي القابل للتفسير، التقييد البنيوي، وآليات التقييم الذاتي. وتمثل التوصيات المقترحة خارطة طريق بحثية واضحة لتطوير النظام نحو جيل أكثر نضجًا من وكلاء إدارة المشاريع الذكية.

إن التركيز المستقبلي على التفسيرية المتقدمة، التقييم المستقل، والتحكم السلوكي يفتح المجال أمام أنظمة لا تكتفي بإنتاج خطط، بل تبرّر قراراتها وتدافع عنها، مما يجعلها أقرب إلى طريقة تفكير مدير مشروع بشري محترف، وأكثر ملاءمة للاستخدام الأكاديمي والتطبيقي.

6 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/>