



الجمهورية العربية السورية
وزارة التعليم العالي والبحث العلمي
الجامعة الوطنية الخاصة
كلية الهندسة (هندسة الحاسوب)

(حلقة بحث)

إدارة المشاريع البرمجية وتتبعها باستخدام JIRA

Manage and track projects with JIRA

إشراف الدكتور:

د.م. علي لؤي ياسين

إعداد الطالب:

عبد الهادي محمد براء الكيلاني

2025 - 2024

1. مقدمة:

في عالمنا الرقمي المتسارع، أصبحت المشاريع البرمجية تمثل حجر الزاوية في تطوير التقنيات الحديثة، من تطبيقات الهواتف الذكية إلى أنظمة الذكاء الاصطناعي، ومن مواقع الإنترنت إلى برمجيات المؤسسات. من التعليم والرعاية الصحية إلى التجارة الإلكترونية والحكومة. تتطلب هذه المشاريع تخطيطاً دقيقاً، وتنفيذاً محكماً، ومتابعة مستمرة لضمان نجاحها وتحقيق أهدافها ضمن الوقت والميزانية المحددين.

وفقاً لـ Sommerville (2016)، فإن البرمجيات ليست مجرد أدوات تقنية، بل هي أنظمة معقدة يجب تطويرها وفق منهجيات علمية وإدارية لضمان فعاليتها واستمراريتها. [1]

2. ماهي إدارة المشاريع وتتبعها:

إدارة المشاريع وتتبعها هي منهجية شاملة وحيوية تُستخدم لتوجيه المشاريع نحو النجاح من خلال التخطيط الدقيق، والتنظيم الفعال للموارد، والتنفيذ المنهجي للخطة، والمراقبة المستمرة للأداء، كل ذلك بهدف تحقيق أهداف محددة ضمن قيود الزمن والميزانية المحددة. تُطبق هذه المنهجية في قطاعات متنوعة كالمهندسة، تكنولوجيا المعلومات، التسويق، والبناء. وفقاً لتعريف معهد إدارة المشاريع (PMI)، تتمثل إدارة المشاريع في استخدام المعارف والمهارات والأدوات والتقنيات لتنفيذ أنشطة المشروع وتحقيق متطلباته، وعادةً ما تمر المشاريع بدورة حياة تتضمن مراحل البدء، التخطيط، التنفيذ، المراقبة والتحكم (وهي المرحلة التي يبرز فيها دور تتبع المشاريع)، وأخيراً الإغلاق. يركز تتبع المشاريع بشكل خاص على المراقبة المستمرة لتقدم العمل مقارنةً بالخطة الموضوعية، مع الاهتمام بالجوانب الرئيسية مثل الوقت، التكلفة، استخدام الموارد، والجودة، لضمان بقاء المشروع على المسار الصحيح. ولتحقيق ذلك، تُستخدم أدوات متنوعة مثل مخططات جانث (Gantt Charts) لتصور الجداول الزمنية، ولوحات كانبان (Kanban Boards) لتتبع تدفق المهام (مثل المستخدمة في جيرا (Jira) وتريلو)، ومخططات الاحتراق (Burndown Charts) لقياس معدل الإنجاز، بالإضافة إلى الاعتماد على مؤشرات الأداء الرئيسية (KPIs) لتقييم مدى الالتزام بالجدول الزمني والميزانية ونطاق العمل المحدد، ومدى تحقيق رضا أصحاب المصلحة. تكمن الأهمية الجوهرية لهذه المنهجية المتكاملة في قدرتها على تحسين كفاءة استخدام الموارد المتاحة، وتقليل المخاطر المحتملة والتأخيرات غير المتوقعة، ورفع مستوى جودة المخرجات النهائية، وبالتالي تحقيق رضا العملاء وكافة الأطراف المعنية بالمشروع. [2]

3. أهمية إدارة المشاريع البرمجية:

إدارة المشاريع البرمجية ليست مجرد تنظيم للمهام، بل هي عنصر أساسي في نجاح أي منتج برمجي. من أبرز الفوائد:

1. تحقيق الأهداف بدقة: تساعد الإدارة الجيدة في تحديد نطاق المشروع، المتطلبات، والمخرجات بدقة، مما يضمن تسليم منتج يلي توقعات العميل.
2. ضبط الوقت والميزانية: من خلال استخدام أدوات تتبع المشروع، يمكن تقليل الهدر في الوقت والموارد.
3. تحسين جودة البرمجيات: بالإدارة الفعالة يمكن تقليل الأخطاء وزيادة موثوقية المنتج.
4. تنسيق فرق العمل: غالباً ما يكون المشروع البرمجي نتاج فريق متنوع المهارات، مما يتطلب تنسيقاً عالياً لضمان التكامل بين الأجزاء المختلفة.

4. أهمية تتبع المشاريع البرمجية:

تتبع المشروع يعني مراقبة تقدمه وتسجيل مراحل ومشاكله، ويُستخدم لتحقيق:

- مراقبة الأداء مقارنة بالخطة الأصلية.
- اكتشاف الانحرافات مبكرًا ومعالجتها.
- تقييم أداء الفريق وتحديد مكان التحسين.
- تعزيز الشفافية بين الفرق وأصحاب المصلحة.
- إدارة المخاطر

5. صعوبة المشاريع البرمجية مقارنة بالمشاريع العادية:

رغم وجود مبادئ إدارية مشتركة، إلا أن المشاريع البرمجية تواجه تحديات فريدة تجعلها أكثر تعقيدًا:

1. **عدم وضوح المتطلبات:** في كثير من الأحيان، لا تكون المتطلبات واضحة أو ثابتة، بل تتغير خلال مراحل التطوير، مما يستلزم أساليب مرنة مثل Agile أو Scrum.
2. **الطبيعة غير الملموسة:** البرمجيات ليست شيئًا ماديًا يمكن قياسه بسهولة، مما يجعل من الصعب تتبع التقدم أو اكتشاف الأخطاء مبكرًا.
3. **التعقيد التقني:** تعدد اللغات والمنصات والتكاملات يجعل المشروع أكثر عرضة للمشكلات التقنية المفاجئة.
4. **اعتماد عالي على الأفراد:** أداء المشروع قد يتأثر بشكل كبير بخبرة وكفاءة المبرمجين، أكثر من المشاريع التقليدية.
5. **الصيانة المستمرة:** لا ينتهي المشروع البرمجي عند التسليم، بل يتطلب تحديثات وتحسينات دائمة، عكس بعض المشاريع العادية التي تنتهي بمجرد إنجازها.
6. **صعوبة التقدير الزمني والمالي:** يصعب تقدير الوقت والتكلفة في المشاريع البرمجية بدقة مقارنة بالمشاريع العادية، بسبب العوامل غير الملموسة وعدم اليقين المرتفع.

إدارة المشاريع البرمجية وتتبعها بشكل فعال يمثلان عنصرًا حاسمًا في ضمان النجاح في عالم يتجه بقوة نحو الرقمنة. ومع أن التحديات التي تواجه هذه المشاريع كبيرة، إلا أن استخدام أدوات الإدارة الحديثة، واتباع منهجيات مرنة، يساعد على التغلب عليها وتحقيق نتائج عالية الجودة. ومن الأدوات المستخدمة مثل Jira، Trello، GitHub Projects. أصبحت من الضروريات لمتابعة هذه المشاريع وسنأخذ Jira كمثال في هذا المقال كمونه يعد من أكثر الأدوات استخدامًا وشهرة.

6. ما هو جيرا (Jira):

جيرا (Jira) هي أداة برمجية قوية مصممة لإدارة المهام والمشاريع، خاصة في مجال تطوير البرمجيات، ولكنها تتعدى ذلك لتشمل إدارة المشاريع في مجالات متنوعة مثل التسويق والموارد البشرية، تتميز جيرا (Jira) بمرونتها وقدرتها على التكيف مع مختلف أنواع المشاريع وأحجام الفرق. تم تطويرها من قبل شركة أتلاسيان (Atlassian) الأسترالية في العام 2002م وما زالت تعمل على ذلك.

رغم أن هذا البرنامج قد أعد بداية لمتابعة مشاكل تطوير البرمجيات، إلا أنه تطور حتى أصبح يمكن استعماله لإدارة المهام بل لإدارة المشاريع بشكل عام، ومن ثم إلى منصة شاملة لإدارة دورة حياة التطوير بأكملها، توفر جيرا (Jira) ميزات لإنشاء المهام وتعيينها، وتتبع حالتها، وتنظيمها في قوائم وألواح مراقبة، مما يساعد الفرق على التنسيق وزيادة الإنتاجية في عملياتها، يوفر جيرا (Jira) بيئة تعاونية تساعد الفرق على تنظيم أعمالها، وزيادة الإنتاجية، وتحسين إدارة المشاريع، يتميز جيرا (Jira) بمجموعة متنوعة من الأدوات والميزات التي تتبع للمستخدمين إدارة المشاريع بطرق مختلفة وفقًا لاحتياجاتهم الفردية.

اسم جيرا (Jira) مأخوذ من كلمة gojira، وهو الاسم الياباني لـ Godzilla، يستخدمه الآن أكثر من مليون عميل في 122 دولة حول العالم.

7. لمحة تاريخية عن جيرا (Jira) وتطوره:

النشأة: ظهرت جيرا (Jira) في بداية الألفية الجديدة كأداة بسيطة لمساعدة فرق تطوير البرمجيات على تتبع الأخطاء والمهام. التوسع: مع مرور الوقت، أصبحت جيرا (Jira) أكثر تعقيداً ومرونة، حيث أضافت ميزات جديدة لتلبية احتياجات مجموعة واسعة من الفرق والمشاريع كما تطورت جيرا (Jira) بمرور الوقت لتشمل ميزات متقدمة لإدارة المشاريع في مختلف الصناعات، مثل الإدارة التقنية، وإدارة الخدمات، والتسويق، والموارد البشرية. الشهرة العالمية: اكتسبت جيرا (Jira) شعبية واسعة في صناعة التكنولوجيا، واعتمدها العديد من الشركات الكبرى والصغيرة على حد سواء، وتعتبر جزءاً أساسياً من بيئات العمل الرقمية الحديثة التي تسهم في تنظيم الفرق وزيادة كفاءتها في إنجاز المهام. التكامل: تم دمج جيرا (Jira) مع العديد من الأدوات الأخرى، مثل أدوات إدارة الإصدارات وأدوات التكامل المستمر، مما جعلها جزءاً لا يتجزأ من سلسلة أدوات تطوير البرمجيات.

8. الاستخدامات الشائعة لجيرا (Jira):

- تطوير البرمجيات: تتبع الأخطاء، وإدارة المهام، والتخطيط لإصدارات جديدة.
- إدارة المشاريع: إدارة المشاريع من البداية إلى النهاية، وتتبع التقدم، وإدارة المخاطر.
- إدارة الخدمات: إدارة طلبات العملاء وحلها.
- إدارة الموارد البشرية: تتبع المهام الموكلة للموظفين وتقييم أدائهم.

9. جيرا (Jira): مركز القيادة لمشاريعك الرقمية:

فكر في جيرا (Jira) كمركز تحكم متطور وشامل، صممتها شركة Atlassian، لمساعدة الفرق، خاصة تلك التي تعمل في مجال تطوير البرمجيات ولكن ليس حصراً عليها، على تنظيم وإدارة مشاريعها بفعالية كبيرة. إنها أكثر من مجرد قائمة مهام؛ إنها بيئة عمل متكاملة تجمع كل ما يتعلق بالمشروع في مكان واحد.

ويأتي هنا السؤال المهم: لماذا تعتمد الفرق على جيرا (Jira)؟

1. تنظيم فائق للعمل: المشاريع الكبيرة يمكن أن تصبح فوضوية بسرعة. جيرا (Jira) يسمح بتقسيم العمل إلى وحدات أصغر وأكثر قابلية للإدارة. قد تكون هذه قصة مستخدم (Story) تصف ميزة من وجهة نظر المستخدم (مثلاً: كمستخدم للموقع، أريد تسجيل الدخول باستخدام حساب جوجل)، أو مهمة (Task) لعمل محدد (مثلاً: إنشاء تصميم واجهة المستخدم لصفحة التسجيل)، أو مهمة فرعية (Sub-task) وهي جزء صغير من مهمة أكبر (مثلاً: تصميم زر تسجيل الدخول). هذا التقسيم يجعل تتبع التقدم وتوزيع العمل أسهل بكثير.
2. أساليب عمل مرئية (سكرم وكنبان): جيرا (Jira) يدعم بقوة منهجيات إدارة المشاريع الرشيقة (Agile). يقدم لوحات سكرم (Scrum Boards) للفرق التي تعمل بنظام السباقات (Sprints) - فترات زمنية قصيرة يتم فيها إنجاز مجموعة محددة من المهام. كما يقدم لوحات كانبان (Kanban Boards) التي تركز على التدفق المستمر للعمل، حيث تنتقل المهام عبر مراحل مختلفة (مثل: قيد الانتظار، قيد التنفيذ، للمراجعة، تم الإنجاز). هذه اللوحات تجعل سير العمل مرئياً للجميع، مما يعزز الشفافية. يمكن لأي عضو في الفريق رؤية ما يتم العمل عليه، ومن يعمل عليه، وما هي المرحلة التي وصل إليها. هذا يساعد في تحديد الاختناقات والمشاكل بسرعة.

3. تتبع دقيق للأخطاء (Bugs): في تطوير البرمجيات، الأخطاء أمر لا مفر منه. جيرا (Jira) يتفوق في تتبع الأخطاء. يمكنك بسهولة تسجيل خطأ جديد، وتحديد أولويته، وتعيينه للشخص المسؤول عن إصلاحه. الأهم من ذلك، يمكنك تتبع دورة حياة الخطأ بالكامل من لحظة اكتشافه (جديد)، إلى مرحلة العمل على إصلاحه (قيد التنفيذ)، ثم اختباره (جاهز للاختبار)، وأخيراً إغلاقه (مغلق). يمكن ربط هذه الأخطاء مباشرة بالكود المصدري أو بمتطلبات معينة، مما يساعد المطورين والمختبرين على فهم السياق وتسريع عملية الإصلاح وتحسين جودة المنتج النهائي.

4. محور للتعاون: جيرا (Jira) ليس مجرد أداة لإدارة المهام، بل هو منصة للتعاون. يمكن لأعضاء الفريق ترك تعليقات على المهام، الإشارة لزملائهم (@mention)، مشاركة الملفات والمرفقات، ومناقشة التفاصيل المتعلقة بكل مهمة. هذا يقلل الحاجة إلى رسائل بريد إلكتروني لا نهاية لها أو اجتماعات غير ضرورية، ويبقي جميع المحادثات والمعلومات الهامة مرتبطة مباشرة بالعمل نفسه.

5. مرونة وقابلية تخصيص عالية: لا يوجد مشروعان متماثلان تمامًا. جيرا (Jira) يدرك ذلك ويتيح درجة عالية من التخصيص. يمكنك إنشاء سير عمل (Workflow) مخصص يعكس بالضبط الخطوات التي تمر بها المهام في فريقك (قد يكون سير عمل بسيط مثل للقيام به - قيد التنفيذ - تم أو معقد يتضمن خطوات موافقة ومراجعة متعددة). يمكنك أيضًا إضافة حقول مخصصة (Custom Fields) لجمع معلومات محددة تحتاجها (مثل: نوع الجهاز المتأثر، إصدار البرنامج، القسم المسؤول). هذه المرونة تجعل جيرا (Jira) قابلاً للتكيف مع احتياجات أي فريق أو مشروع تقريباً.

6. قوة التكامل: تزداد قوة جيرا (Jira) عندما يتم دمجها (Integration) مع الأدوات الأخرى التي يستخدمها فريقك. يمكن ربطه بأدوات إدارة الشيفرة المصدريّة (مثل Git، Bitbucket)، أدوات التواصل (مثل Slack، Microsoft Teams)، أدوات الاختبار، منصات الدعم، وغيرها الكثير. هذا التكامل يخلق تجربة عمل سلسلة حيث تتدفق المعلومات تلقائياً بين الأنظمة المختلفة، مما يوفر الوقت ويقلل الأخطاء.

7. رؤية تحليلية وتقارير: جيرا (Jira) لا يقتصر على تتبع العمل؛ بل يساعدك على فهمه وتحسينه. يوفر مجموعة من التقارير التحليلية. أشهرها مخطط الاحتراق (Burndown Chart) الذي يوضح بصرياً مقدار العمل المتبقي في سياق (Sprint) مقارنة بالوقت المتاح، مما يساعد الفريق على معرفة ما إذا كانوا على المسار الصحيح. هناك تقارير أخرى مثل مخطط السرعة (Velocity Chart) الذي يقيس كمية العمل التي إنجازها الفريق عادةً في كل سباق، مما يساعد في التخطيط المستقبلي. هذه التقارير تدعم اتخاذ قرارات مبنية على البيانات.

10. إصدارات جيرا (Jira):

- **Jira Software**: مخصص للفريق التقنية.
- **Jira Service Management (JSM)**: لإدارة الدعم الفني وفرق IT.
- **Jira Work Management (JWM)**: موجه للفريق غير التقنية كالموارد البشرية والتسويق.

11. مكونات جيرا (Jira) الأساسية:

1. **المشاريع (Projects)**: المشروع في جيرا (Jira) هو بمثابة الحاوية الرئيسية أو المجلد الذي يجمع كل ما يتعلق بهدف معين أو منتج تعمل عليه. كل مشروع يمكن أن يكون له إعداداته الخاصة، مثل أعضاء الفريق، أنواع التذاكر المتاحة، والمخططات التي تتبعها المهام. هو الوحدة التنظيمية العليا التي تضم كل العمل المتعلق بمبادرة معينة.

مثال: يمكن أن يكون لديك مشروع باسم تطوير تطبيق الجوال ومشروع آخر باسم حملة التسويق للربع الثالث.

2. التذاكر (Issues): التذكرة هي الوحدة الأساسية للعمل داخل جيرا (Jira). لا تقتصر على المشاكل أو الأخطاء فقط، بل يمكن أن تمثل أي شيء يحتاج إلى تتبع وإنجاز. يمكن أن تكون مهمة (Task)، قصة مستخدم (Story)، خطأ برمجي (Bug)، طلب ميزة جديدة (Feature Request)، أو حتى مهمة فرعية (Sub-task) تابعة لتذكرة أكبر. كل تذكرة تحتوي على تفاصيل مثل الوصف، المسؤول عنها، الأولوية، الحالة الحالية (مثل: مفتوح، قيد التنفيذ، تم الإنجاز)، وغيرها.

مثال: تذكرة بعنوان إصلاح خطأ تسجيل الدخول أو تصميم واجهة المستخدم لصفحة الملف الشخصي.

3. اللوحات (Boards): اللوحات هي الواجهة المرئية التي تعرض التذاكر الخاصة بمشروع (أو أكثر) بطريقة تسهل تتبع تقدم العمل. أشهر أنواع اللوحات هي لوحة سكرم (Scrum Board) التي تُستخدم عادةً للفرق التي تعمل بنظام السبرنت (فترات عمل قصيرة ومحددة)، ولوحة كانبان (Kanban Board) التي تركز على التدفق المستمر للعمل وتحديد مراحل العمل المختلفة (مثل: قائمة المهام، قيد التنفيذ، للمراجعة، تم). تساعد هذه اللوحات الفرق على فهم ما يتم العمل عليه حاليًا، ومن يعمل عليه، وما هي الخطوة التالية.

مثال: لوحة كانبان بأعمدة: To Do, In Progress, Code Review, Done.

4. المخططات (Workflows) / سير العمل: المخطط أو سير العمل هو العملية المحددة مسبقًا التي تتبعها التذكرة من بدايتها حتى نهايتها. يحدد المخطط الحالات (Statuses) المختلفة التي يمكن أن تكون عليها التذكرة (مثل: Open, In Progress, Resolved, Closed) والانتقالات (Transitions) المسموح بها بين هذه الحالات (أي ما هي الخطوات التي يمكن اتخاذها لنقل التذكرة من حالة لأخرى). هذا المخطط هو الذي يتحكم فعليًا في كيفية تحرك التذكرة (مثل قصة المستخدم أو المهمة) عبر الأعمدة المختلفة على اللوحة (Board)، حيث يمثل كل عمود عادةً حالة معينة في المخطط. يمكن تخصيص المخططات لتناسب العمليات الفريدة لكل فريق.

مخطط بسيط قد يكون: To Do -> In Progress -> Done. مخطط أكثر تعقيدًا قد يشمل: Backlog -> Selected for Development -> In Progress -> Code Review -> Ready for Testing -> Testing -> Done.

5. التقارير (Reports): توفر جيرا (Jira) مجموعة متنوعة من التقارير المضمنة التي تساعد الفرق والمديرين على تحليل الأداء وفهم التقدم المحرز وتحديد المشاكل المحتملة. هذه التقارير تحول البيانات الموجودة في التذاكر وحركتها عبر المخططات إلى رؤى قابلة للتنفيذ.

أمثلة شائعة: مخطط الاحتراق (Burndown Chart) يُظهر معدل إنجاز العمل المتبقي، مخطط السرعة (Velocity Chart) يقيس كمية العمل التي إنجازها الفريق في كل سبرنت، والمخطط التراكمي للتدفق (Cumulative Flow Diagram) يوضح كيفية تدفق العمل عبر المراحل المختلفة مع مرور الوقت.

12. القوالب في جيرا (Jira):

هو نموذج معد مسبقاً يستخدم لإنشاء مشروع جديد أو تذكرة (issue) بطريقة منظمة حسب نوع العمل المطلوب.

يتيح جيرا (Jira) إمكانية استخدام قوالب تتناسب مع احتياجات الشركة أو الفريق بناءً على المتطلبات الفردية. هذه القوالب تسهل البدء بمشاريع جديدة وتوحيد العمليات داخل المؤسسة بشكل فعال ومنظم نذكر منها: (تطوير البرمجيات، إدارة الخدمات، التسويق، التعليم، الموارد البشرية).

كما يمكن تخصيص القوالب بحسب طبيعة المشروع وحجم الفريق، ولاختيار القالب المناسب، عليك مراعاة العوامل التالية:

طبيعة المشروع: ما هو نوع المشروع الذي تقوم به؟

حجم الفريق: كم عدد الأشخاص الذين يعملون في المشروع؟

المنهجية المستخدمة: هل تستخدم منهجية Agile أم Waterfall؟

الاحتياجات المحددة: ما هي الميزات والوظائف التي تحتاجها في المشروع؟

13. فوائد استخدام جيرا (Jira) في إدارة المشاريع:

يوفر جيرا (Jira) لعدد من المزايا التي تجعله أحد أفضل أدوات إدارة المشاريع، ومنها:

- مرونة عالية: يمكن تخصيصه وفق احتياجات الفرق المختلفة.
- دعم قوي لـ Agile: يساعد الفرق على تطبيق Scrum و Kanban بكفاءة.
- تتبع أداء الفريق: من خلال Dashboards و Reports التي تعرض التقدم بالمشروع
- تكاملات قوية: مع أدوات مثل Confluence و Slack و GitHub.
- تحكم دقيق في الأذونات: يسمح بتحديد أدوار المستخدمين وضبط الصلاحيات.
- الأتمتة والتقارير: يتيح إنشاء Automation Rules لتوفير الوقت وتحسين العمليات.

14. عيوب جيرا (Jira):

- صعب الاستخدام لغير التقنيين.
- أدوات التعاون محدودة بدون تكامل خارجي.
- عدم وجود مخطط زمني مدمج (Timeline view) بدون أدوات إضافية.

15. تكاملات جيرا (Jira):

- GitHub, Bitbucket, GitLab: لإدارة الكود.
- Jenkins, Bamboo, CircleCI: لـ CI/CD.
- Xray, Zephyr: لاختبار البرمجيات.
- Slack, Teams: للتواصل.
- Zapier, Power Automate: لأتمتة المهام.

• **Confluence**: للتوثيق، يُستخدم مع جيرا (Jira) لتوليد تقارير وPRDs.

16. من يمكنه استخدام برنامج جيرا (Jira):

يمكن استخدام جيرا (Jira) من قبل أنواع مختلفة من الفرق في مختلف الصناعات وذلك بسبب تعدد وظائفه. وأهم هذه الفرق التي تستخدم برنامج جيرا (Jira) هي:

- فرق تطوير البرمجيات.
- فرق تكنولوجيا المعلومات.
- فرق التسويق.
- فرق إدارة المشاريع.
- فرق إدارة المنتجات.
- فرق دعم العملاء.
- فرق الموارد البشرية.
- فرق التصنيع والهندسة.
- المؤسسات التعليمية.
- فرق التمويل والعمليات.

17. مقارنة جيرا (Jira) مع أدوات إدارة المشاريع الأخرى:

على الرغم من أن جيرا (Jira) هو أداة قوية، إلا أن هناك العديد من الأدوات المنافسة في مجال إدارة المشاريع. يوضح الجدول مقارنة بين جيرا (Jira) وبعض الأدوات الأخرى الشهيرة:

الجدول (1): مقارنة بين أدوات إدارة المشاريع.

الميزة	Jira	Trello	Asana	Azure DevOps
دعم Agile	نعم	جزئيًا	محدود	نعم
إدارة المشاريع الكبيرة	نعم	لا	نعم	نعم
إدارة تتبع الأخطاء (Bug Tracking)	نعم	لا	لا	نعم
تكاملات DevOps	نعم	لا	لا	نعم
سهولة الاستخدام	متوسط	سهل جدًا	سهل	متوسط

• Trello: مناسب للمشاريع الصغيرة والبسيطة التي تحتاج إلى لوحات Kanban مرنة .

• Asana: مناسب للفرق غير التقنية التي تحتاج إلى إدارة المهام والتعاون السريع .

• Azure DevOps: خيار قوي للفرق التي تعمل على مشاريع Microsoft. [3]

18. بعض تجارب المستخدمين:

1. **Nasa**: الاستخدام: تتبع وإدارة أخطاء البرمجيات في المشاريع البحثية والفضائية.
أثر جيرا (**Jira**): سهّل على الفرق التقنية إدارة التذاكر والمشاكل البرمجية بشكل دقيق، خاصة في مشاريع تتطلب أعلى درجات الدقة.
2. **Spotify**: الاستخدام: إدارة فرق تطوير التطبيقات وتنظيم سير العمل.
أثر جيرا (**Jira**): ساعد في تنسيق عمل الفرق المتعددة حول العالم مع الحفاظ على إيقاع تطوير سريع ومتكرر.
3. **Airbnb**: الاستخدام: تتبع الميزات الجديدة والمشاكل في نظام الحجز.
أثر جيرا (**Jira**): زاد من قدرة فرق المنتج على الاستجابة السريعة لمتطلبات المستخدمين والمشكلات الطارئة.
4. **eBay**: الاستخدام: إدارة مشاريع برمجية ضخمة متعددة الفرق.
أثر جيرا (**Jira**): وفر رؤية مركزية لسير العمل ومكن الفرق من العمل بتكامل رغم تنوع المهام.
5. **Tesla**: الاستخدام: فرق تطوير الأنظمة البرمجية للسيارات.
أثر جيرا (**Jira**): مكّن الفرق من إدارة التحديثات البرمجية والتكامل بين البرمجيات والهاردوير بشكل منظم.
6. **Cisco Systems**: الاستخدام: مشاريع البنية التحتية والبرمجيات الشبكية.
أثر جيرا (**Jira**): دعم توثيق ومتابعة الأنظمة التقنية المعقدة.
7. **LinkedIn**: الاستخدام: تتبع تطوير المزايا وتحسين الأداء.
أثر جيرا (**Jira**): وفر هيكلية واضحة لإدارة مراحل التطوير من التصميم حتى النشر. [4]

19. دراسة حالة شركة (JPL) Nasa's Jet Propulsion Laboratory :

مختبر الدفع النفاث التابع لناسا (JPL) هو مركز أبحاث وتطوير يركز على استكشاف الفضاء. واجهت فرق العمل في JPL تحديات في إدارة المشاريع المعقدة والتنسيق بين الفرق المختلفة. لتجاوز هذه التحديات، اعتمدت JPL على جيرا (Jira) لتحسين عملياتها.

التحديات:

- إدارة مشاريع معقدة تتطلب تنسيقاً دقيقاً بين فرق متعددة التخصصات.
- الحاجة إلى نظام يتسم بالمرونة وقابلية التوسع لتلبية احتياجات المشاريع المختلفة.

الحلول المقدمه:

- تخصيص سير العمل: تم تعديل سير العمل في جيرا (Jira) ليتناسب مع العمليات الفريدة لـ JPL ، مما سمح بتتبع المهام والمشاريع بكفاءة.
- التكامل مع الأدوات الأخرى: تم دمج جيرا (Jira) مع أدوات أخرى مستخدمة في JPL ، مما أدى إلى تحسين تدفق المعلومات وتقليل الازدواجية.

- التقارير ولوحات المعلومات: استخدمت JPL ميزات التقارير ولوحات المعلومات في جيرا (Jira) لمراقبة تقدم المشاريع واتخاذ قرارات مستنيرة.

النتائج:

- تحسين الكفاءة: أدى استخدام جيرا (Jira) إلى تحسين التنسيق بين الفرق وتقليل الوقت المستغرق في إدارة المهام.
- زيادة الشفافية: أصبح لدى جميع أعضاء الفريق رؤية واضحة لحالة المشاريع والمهام، مما ساعد في تحديد المشكلات المحتملة مبكرًا.
- تعزيز التعاون: ساهم جيرا (Jira) في تعزيز التعاون بين الفرق المختلفة من خلال توفير منصة مشتركة لتبادل المعلومات.

20. مستقبل جيرا (Jira):

استجابةً للتطورات التقنية واحتياجات السوق المتغيرة، تسعى جيرا (Jira) إلى تعزيز تكاملها مع تقنيات الذكاء الاصطناعي لتحسين قدرات إدارة المشاريع. من المتوقع أن يشمل ذلك ميزات مثل التنبؤ بالمخاطر، وتخصيص الموارد بذكاء، وتحليل البيانات لتوفير رؤى أعمق حول سير العمل. تهدف هذه الخطوات إلى تعزيز كفاءة الفرق وتمكينها من اتخاذ قرارات مستنيرة ونذكر بعضها:

- التحول إلى السحابة.
- دعم DevOps بشكل أعمق.
- أتمتة أكثر.
- تقارير تحليلية متقدمة.
- دعم أفضل لتجربة المستخدم ومنهجيات Agile متقدمة.
- التوسع في التكامل مع الأدوات الأخرى.

21. أبرز التحديات التي تواجه جيرا (Jira):

1. التكيف مع احتياجات المستخدمين المتنوعة: على الرغم من أن جيرا (Jira) بدأت كأداة موجهة لفرق تطوير البرمجيات، إلا أنها توسعت لتشمل صناعات مختلفة. هذا التوسع يتطلب تلبية احتياجات مستخدمي ذوي خلفيات تقنية متفاوتة، مما يشكل تحديًا في توفير تجربة مستخدم سلسة للجميع.
2. منافسة الأدوات البديلة: مع وجود أدوات أخرى في السوق تقدم ميزات مشابهة أو محسنة، مثل Rally وAha، تواجه جيرا (Jira) تحديًا في الحفاظ على حصتها السوقية وضمان تقديم قيمة مضافة لعملائها.
3. التغيرات في استراتيجيات التسعير والدعم: إعلان شركة Atlassian عن إنهاء دعم خوادم جيرا (Jira) بحلول عام 2024 وتغيير استراتيجيات التسعير قد يدفع بعض العملاء للبحث عن بدائل أخرى، خاصةً الشركات التي تعتمد على حلول الخوادم المحلية مثل Easy Redmine.

22. الأفاق المستقبلية لجيرا (Jira):

بالرغم من التحديات المذكورة، فإن جيرا (Jira) تمتلك فرصًا كبيرة لتعزيز مكانتها في سوق إدارة المشاريع. من خلال الاستثمار في تقنيات الذكاء الاصطناعي وتعلم الآلة، يمكنها تحسين تجربة المستخدم وتقديم حلول أكثر ذكاءً وفعالية. بالإضافة إلى ذلك، قد تسهم الشركات مع منصات وأدوات أخرى في توسيع نطاق خدماتها وتلبية احتياجات قاعدة أوسع من العملاء. [5]

الخلاصة:

تواجه جيرا (Jira) تحديات متعددة في ظل سوق ديناميكي ومتغير، إلا أن التزامها بالابتكار والتكيف مع احتياجات المستخدمين قد يضمن لها مستقبلًا مشرقًا. من خلال التركيز على تحسين ميزاتها وتقديم حلول متكاملة، يمكن لجيرا (Jira) الاستمرار في كونها أداة رائدة في مجال إدارة المشاريع.

23. الخاتمة:

يُعد جيرا (Jira) أداة رائدة في إدارة المشاريع البرمجية وتتبع الأخطاء، حيث استطاع منذ إنطلاقه عام 2002 أن يتطور من أداة بسيطة لتتبع الأخطاء البرمجية إلى منصة متكاملة تدعم منهجيات Agile مثل Scrum و Kanban بفضل قدراته الواسعة في تسجيل الأخطاء وتحليلها وإعداد التقارير التحليلية، ساهم جيرا (Jira) في مساعدة الفرق على تحسين الإنتاجية وتعزيز التعاون الفعال.

تعتمد العديد من الشركات العالمية على جيرا (Jira) مثل مختبر الدفع النفاث التابع لناسا (JPL) لإدارة المشاريع الفضائية المعقدة، بالإضافة إلى شركات أخرى التي دمجت جيرا (Jira) لتوحيد أنظمة إدارة المشاريع وتحسين سير العمل. كما يوفر جيرا (Jira) تكاملًا قويًا مع أدوات متنوعة مثل GitHub ، Slack ، Bitbucket ، و Confluence ، مما يساهم في دعم التعاون وتحسين الأداء.

إن التزام جيرا (Jira) المستمر بالابتكار والاستجابة لاحتياجات المستخدمين المختلفة يجعله خيارًا مثاليًا لإدارة المشاريع في بيئات العمل الحديثة. ورغم التحديات المتغيرة، يظل جيرا (Jira) أداة فعالة لتحقيق الأهداف وتعزيز التنسيق والإنتاجية في الفرق المتنوعة.

المراجع:

- [1] Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.
- [2] Project Management Institute (PMI), A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 7th Edition.
- [3] Atlassian Documentation - <https://www.atlassian.com/software/jira>
- [4] Atlassian Case Studies - <https://www.atlassian.com/customers>
- [5] مرحبا بكم في العصر الجديد لجيرا - <https://www.atlassian.com/blog/announcements/the-next-era-of-jira>