

الوحدة التدريبية الثانية
الحفاظ على وثائق العمل في جميع مراحل تطوير التطبيق

رمز الوحدة
ADS-U2-CAVT

لا يوجد

نتائج التعلم:

عند الانتهاء من دراسة هذه الوحدة واكتساب مهاراتها الادائية والاتجاهات السلوكية الصحيحة والعلوم المهنية المرافقة من خلال التفاعل مع أنشطتها وخبراتها المختلفة يصبح المتدرب قادراً على أداء نتائج التعلم الآتية:

1. المساهمة في إنشاء وصيانة السجلات الادارية الورقية والرقمية (المراسلات والتقارير والوثائق الإدارية الأخرى).
2. المساهمة في توثيق متطلبات العملاء بما فيها مدخلات التطبيق والمخرجات والعمليات والمستخدمين وصلاحيات الوصول الخاصة بهم.
3. المساهمة في توثيق المتطلبات الفنية لبيئة تطوير التطبيقات والمتطلبات الفنية من جانب العميل لنشر التطبيقات.
4. المساهمة في توثيق مواصفات التطبيق حالات الاستخدام (User Cases) وقصص المستخدم (User Stories).
5. المساهمة في توثيق التعليمات البرمجية، ومراجعتها عند الحاجة وعلى التعليمات البرمجية الأصلية.
6. المساهمة في أرشفة الوثائق والبيانات بشكل ورقي و/أو رقمي وفي جميع المراحل.

مصادر التعلم	الأنشطة التعليمية والتدريبية
الوحدة التدريبية	قراءة المعلومات النظرية
الشبكة العنكبوتية	البحث في المواقع الالكترونية التعليمية
منصة الكلية التعليمية	روابط التعليم الالكتروني
المشغل / المختبر	تنفيذ التمارين العملية
سوق العمل	التدريب الميداني

روابط التعلم الإلكتروني

سيتم تزويد المتدربين بروابط التعلم الإلكتروني من خلال المدرب

بين ما أهمية أرشفة وتوثيق الوثائق خلال عملية تطوير التطبيقات وأثرها على جودة العمل ؟

1. أرشفة الوثائق وأهميتها لمشاريع تطوير البرمجيات .

في عالم تطوير البرمجيات يعد التوثيق الفعال أمراً ضرورياً لضمان نجاح المشروع. بدءاً من جمع متطلبات العملاء وحتى توثيق المواصفات الفنية وتعليقات التعليمات البرمجية، يعد الحفاظ على التوثيق الشامل على مدار دورة حياة تطوير التطبيق أمراً بالغ الأهمية لتسهيل التعاون وضمان الجودة ودعم جهود الصيانة المستقبلية. في هذا الوحدة ، سوف نستكشف أهمية حفظ وثائق العمل والحفاظ عليها في جميع مراحل تطوير التطبيق، وسنعمل على تقديم إرشادات عملية حول كيفية توثيق الجوانب المختلفة لمشروع البرنامج بشكل فعال.

• مفهوم التوثيق وأهميته :

يشير التوثيق في تطوير البرمجيات إلى عملية جمع وتسجيل المعلومات ذات الصلة بتصميم أنظمة البرمجيات وتطويرها واختبارها ونشرها وصيانتها. يمكن أن تتخذ هذه الوثائق أشكالاً مختلفة، بما في ذلك المستندات المكتوبة والرسوم البيانية وتعليقات التعليمات البرمجية وأدلة المستخدم والمواصفات الفنية وخطط المشروع. يخدم التوثيق العديد من الأغراض الحاسمة في مشاريع تطوير البرمجيات:

• **التواصل والتعاون:** يسهل التوثيق التواصل الفعال بين أعضاء الفريق وأصحاب المصلحة والمشاركين الآخرين في المشروع. فهو يضمن حصول جميع المشاركين في المشروع على نفس المعلومات وفهم أهداف المشروع ومتطلباته وتفاصيل التنفيذ. يساعد التوثيق الواضح المطورين على التعاون بكفاءة وحل أوجه الغموض ومواءمة جهودهم نحو الأهداف المشتركة.

• **نقل المعرفة والحفاظ عليها:** غالباً ما تشتمل المشاريع البرمجية على أعضاء فريق متعددين يعملون على جوانب مختلفة من المشروع في مراحل مختلفة. يعد التوثيق بمثابة مستودع للمعرفة، مما يسمح لأعضاء الفريق بمشاركة الأفكار وأفضل الممارسات والدروس المستفادة. فهو يمكّن أعضاء الفريق الجدد من الانضمام بسرعة، وفهم التعليمات البرمجية، والمساهمة بفعالية في المشروع. يحافظ التوثيق أيضاً على المعرفة المؤسسية، مما يضمن عدم فقدان المعلومات المهمة عندما يغادر أعضاء الفريق أو ينتقلون إلى أدوار جديدة.

• **تقليل الأخطاء:** يساعد التوثيق المطورين على فهم متطلبات المشروع ومواصفات التصميم ومعايير الترميز، مما يقلل من احتمالية الأخطاء وسوء الفهم أثناء عملية التطوير. من خلال توفير إرشادات واضحة وأفضل الممارسات، تساعد التوثيق المطورين على الالتزام بالمعايير المعمول بها، وتجنب المخاطر الشائعة، وإنتاج تعليمات برمجية عالية الجودة. بالإضافة إلى ذلك، فإن التوثيق الشامل لإجراءات الاختبار ونتائجه يمكن

القائمين على الاختبار من تحديد العيوب ومعالجتها في وقت مبكر من دورة حياة التطوير، مما يقلل العدد الإجمالي للأخطاء ويعزز موثوقية البرامج.

- **إدارة المتطلبات:** يلعب التوثيق دورًا حيويًا في جمع وإدارة متطلبات المشروع. ومن خلال توثيق احتياجات العملاء والمواصفات الوظيفية ومعايير القبول، فإنه يضمن أن المطورين لديهم فهم واضح لما يجب بناؤه وكيف يجب التصرف. تعمل المتطلبات الموثقة جيدًا كنقطة مرجعية للتحقق من صحة وظائف البرنامج وتتبع تقدم المشروع وإدارة التغييرات على مدار دورة حياة التطوير.

- **ضمان الجودة والاختبار:** يدعم التوثيق أنشطة ضمان الجودة والاختبار من خلال توفير خطط الاختبار وحالات الاختبار والنتائج المتوقعة. فهو يمكّن المختبرين من التحقق من أن البرنامج يلي المتطلبات المحددة ويتصرف على النحو المقصود في ظل سيناريوهات مختلفة. تساعد الوثائق التفصيلية على تحديد العيوب وتتبع إصلاحات الأخطاء والتأكد من استيفاء معايير جودة البرامج قبل الإصدار.

- **الصيانة واستكشاف الأخطاء وإصلاحها:** بعد النشر، تتطلب أنظمة البرامج صيانة ودعمًا مستمرين لمعالجة المشكلات وتنفيذ التحديثات واستيعاب التغييرات في احتياجات المستخدم أو متطلبات العمل. يسهل التوثيق أنشطة الصيانة من خلال توفير رؤى حول بنية النظام وقرارات التصميم وبنية التعليمات البرمجية. فهو يساعد المطورين على فهم الأساس المنطقي وراء القرارات السابقة، واستكشاف المشكلات وإصلاحها بكفاءة، وإجراء تعديلات مستنيرة دون حدوث آثار جانبية غير مقصودة.

استعرض أنواع مختلفة من الوثائق المستخدمة في تطوير البرمجيات، بما في ذلك وثائق المتطلبات ووثائق التصميم وأدلة المستخدم



2. أنواع التوثيق :

في تطوير البرمجيات، يتم استخدام أنواع مختلفة من الوثائق لجمع جوانب مختلفة من دورة حياة المشروع، وتسهيل الاتصال، وضمان نجاح المشروع. تتضمن بعض الأنواع الرئيسية للوثائق ما يلي:

- **وثائق المتطلبات:** توضح وثائق المتطلبات الاحتياجات والتوقعات والمواصفات الخاصة بنظام البرنامج من وجهة نظر أصحاب المصلحة، بما في ذلك العملاء والمستخدمين والمشاركين الآخرين في المشروع، حيث تتضمن هذه المستندات عادةً ما يلي:
- **المتطلبات الوظيفية:** أوصاف ووظائف النظام وميزاته وإمكانياته، والتي يتم التعبير عنها غالبًا في شكل قصص مستخدم أو حالات استخدام أو مواصفات وظيفية.

- المتطلبات غير الوظيفية: المواصفات المتعلقة بسمات النظام مثل الأداء وقابلية التوسع والموثوقية والأمان وسهولة الاستخدام.
- معايير القبول: المعايير التي يجب استيفاؤها حتى يعتبر البرنامج مقبولاً من قبل أصحاب المصلحة، وغالباً ما تستخدم لتحديد نطاق الاختبار واختبار القبول.
- وثائق التصميم: تصف وثائق التصميم بنية نظام البرنامج ومكوناته، مما يوفر إرشادات للمطورين أثناء مرحلة التنفيذ. قد تشمل هذه المستندات ما يلي:
 - هندسة النظام: أوصاف عالية المستوى للبنية العامة للنظام، بما في ذلك المكونات والوحدات والطبقات وتفاعلاتها.
 - تصميم قاعدة البيانات: مواصفات تنظيم وهيكل وعلاقات تخزين بيانات النظام، بما في ذلك مخططات العلاقة بين الكيانات (ERDs) ومخططات قاعدة البيانات.
 - تصميم الواجهة: تصميمات لواجهات المستخدم وواجهات برمجة التطبيقات ونقاط التكامل، بما في ذلك الإطارات السلوكية والنماذج بالحجم الطبيعي ومواصفات الواجهة.
- أدلة المستخدم: توفر أدلة المستخدم إرشادات وتعليمات للمستخدمين النهائيين حول كيفية استخدام نظام البرنامج بفعالية. تتضمن هذه المستندات عادةً ما يلي:
 - تعليمات التثبيت: التعليمات تبين لك خطوة بخطوة تثبيت البرنامج وتكوينه على منصات وبيئات مختلفة.
 - أدلة المستخدم: تبين لك أوصاف لميزات البرنامج ووظائفه وسير العمل، مصحوبة بقطاعات شاشة وأمثلة ونصائح حول استكشاف الأخطاء وإصلاحها.
 - المواد المرجعية: تبين لك توثيق أوامر النظام، والخيارات، و (Parameters) ، والاختصارات، والتي يتم تنظيمها غالباً في شكل جداول أو قوائم أو فهرس قابلة للبحث.
- خطط الاختبار وحالات الاختبار: وهي توثيق استراتيجيات الاختبار وأهدافه وإجراءاته، بالإضافة إلى حالات الاختبار التفصيلية والسيناريوهات والنتائج المتوقعة لتحقيق من صحة وظائف البرنامج وجودته.
- توثيق التعليمات البرمجية: ان التعليقات والشروح والوثائق المضمنة في التعليمات البرمجية المصدرية (Source Code) توضح الغرض منها ومنطقها واستخدامها، كما يتضمن ذلك التعليقات المضمنة وتعليقات التوثيق (مثل JavaDoc وملفات README).

- خطط المشروع وتقارير الحالة: تعني توثيق الجداول الزمنية للمشروع، والمعالم الرئيسية، والتسليمات، وتحديثات التقدم، المستخدمة لتتبع أنشطة المشروع، وتخصيص الموارد، وإبلاغ حالة المشروع إلى أصحاب المصلحة.

اطلع
ونسأل

اشرح كيف يساعد التوثيق في جمع وتوضيح متطلبات المشروع، بما في ذلك قصص المستخدم وحالات الاستخدام والمواصفات الوظيفية

3. التوثيق على مدار دورة حياة تطوير البرمجيات :

التوثيق هو عنصر أساسي في كل مرحلة من مراحل دورة حياة تطوير البرمجيات (Software Development Life Cycle - SDLC). تبدأ هذه العملية بتوثيق المتطلبات، حيث يتم تحديد أهداف المشروع ومواصفاته بدقة. ثم تأتي وثائق التصميم التي تسلط الضوء على هيكل النظام ومكوناته بالتفصيل. خلال مرحلة التطوير، تساعد وثائق التعليمات البرمجية والتعليقات المضمنة في تعزيز الوضوح وسهولة الصيانة. أما أثناء الاختبار، فتوثق حالات الاختبار والإجراءات والنتائج بشكل دقيق. وفي المرحلة الأخيرة، توفر وثائق النشر والصيانة إرشادات شاملة حول التثبيت والاستخدام والتحديثات المستقبلية. يساهم التوثيق المتكامل في ضمان سير عملية التطوير بسلاسة وكفاءة، مما يسهل إدارة المشروع ويعزز استدامة البرامج على المدى البعيد.

أولاً : مرحلة جمع المتطلبات :

يلعب التوثيق دوراً حاسماً في جمع وتوضيح متطلبات المشروع من خلال توفير إطار منظم لتحديد وتوصيل الاحتياجات والتوقعات والمواصفات الخاصة بنظام البرمجيات. على وجه التحديد، يساعد التوثيق في جمع وتوضيح متطلبات المشروع، بما في ذلك قصص المستخدم وحالات الاستخدام والمواصفات الوظيفية، بالطرق التالية:

- تحديد قصص المستخدم:
- جمع احتياجات المستخدم: قصص المستخدم هي عبارة عن وصف موجز للوظائف المطلوبة من وجهة نظر المستخدمين النهائيين. يسهل التوثيق لك وللمطورين عملية جمع احتياجات المستخدم ومتطلباته من خلال توفير القوالب والإرشادات والأمثلة لكتابة قصص مستخدم واضحة وقابلة للتنفيذ.
- توضيح المتطلبات: يتيح التوثيق لك ولأصحاب المصلحة توضيح قصص المستخدمين وتوفير سياق إضافي وتفاصيل ومعايير قبول لتوضيح النتائج والوظائف المطلوبة. ويساعد ذلك على ضمان حصول المطورين على فهم واضح لما يجب تنفيذه وكيف يجب أن يتصرفوا.

• تطوير حالات الاستخدام :

- وصف تفاعلات المستخدم: تصف لك حالات الاستخدام التفاعلات بين المستخدمين (الجهات الفاعلة) والنظام لتحقيق أهداف أو مهام محددة، كما يسهل التوثيق تطوير حالات الاستخدام من خلال توفير

إرشادات لتحديد الجهات الفاعلة وتحديد السيناريوهات وتحديد سلوك النظام استجابةً لإجراءات المستخدم.

- التحقق من صحة المتطلبات: يعمل التوثيق كأساس للتحقق من صحة المتطلبات من خلال مخططات حالة الاستخدام، والمخططات الانسيابية، والأوصاف النصية. من خلال تصور تفاعلات المستخدم واستجابات النظام، يمكن لأصحاب المصلحة التأكد من أن الوظيفة المقترحة تتوافق مع احتياجات المستخدم وتوقعاته.

- إنشاء المواصفات الوظيفية :

- تفصيل وظائف النظام: تحدد المواصفات الوظيفية الميزات والوظائف والسلوكيات المحددة التي يجب أن يظهرها نظام البرنامج. يساعد التوثيق في إنشاء المواصفات الوظيفية من خلال تنظيم المتطلبات وهيكلتها في قائمة شاملة لقدرات النظام.

- حل أوجه الغموض : يتيح التوثيق لأصحاب المصلحة توضيح أوجه الغموض والتناقضات والصراعات في المتطلبات الوظيفية من خلال المناقشات والمراجعات والتعليقات. ومن خلال توثيق القرارات ، يضمن التوثيق فهم المتطلبات والاتفاق عليها من قبل جميع الأطراف المعنية.

- تسهيل التعاون والتواصل:

- الفهم المشترك: يعد التوثيق بمثابة نقطة مرجعية مشتركة لأصحاب المصلحة، مما يوفر فهمًا مشتركًا لمتطلبات المشروع وأهدافه. و يتيح التواصل والتعاون الفعالين بين المشاركين في المشروع.
- التغذية الراجعة والتكرار: يسهل التوثيق مراجعة وتحسين متطلبات المشروع من خلال حلقات التغذية الراجعة. فمن خلال توثيق سجل المراجعات والتحديثات والتغييرات في المتطلبات، يمكن لأصحاب المصلحة تتبع تطور متطلبات المشروع والتأكد من أنها تبقى متوافقة مع احتياجات وأولويات العمل المتطورة.

ناقش دور التوثيق في وصف بنية النظام ومخططات قاعدة البيانات وتصميمات الواجهة

ثانيا : مرحلة التصميم:

يلعب التوثيق دورًا حاسمًا في وصف الجوانب المختلفة لنظام البرمجيات، بما في ذلك بنية النظام ومخططات قاعدة البيانات وتصميمات الواجهة. وإليك كيف يخدم التوثيق هذا الغرض لكل جانب:

• وصف بنية النظام:

- نظرة عامة على المكونات: توفر الوثائق نظرة عامة على بنية النظام، مع تحديد هيكل وتنظيم مكوناته ووحداته وطبقاته، كما تصف كيفية تفاعل هذه المكونات مع بعضها البعض لتحقيق أهداف النظام.
- التواصل بشأن قرارات التصميم: يوضح لك التوثيق الأساس المنطقي وراء قرارات التصميم، بما في ذلك اختيار الأنماط والأطر والتقنيات. وهو يشرح المقايضات التي تمت أثناء عملية التصميم ويبرر سبب اختيار أنماط تصميم معينة على غيرها.
- تصور العلاقات: قد يتضمن التوثيق مخططات مثل المخططات المعمارية، ومخططات المكونات، ومخططات النشر لتمثيل العلاقات بين مكونات النظام، وتبعياتها، وتوزيع المكونات عبر الأجهزة أو البيئات المتصلة بالشبكة بشكل مرئي.

• توثيق مخططات قاعدة البيانات:

- نمذجة العلاقة بين الكيان: تصف الوثائق بنية قاعدة البيانات باستخدام مخططات العلاقة بين الكيان (ERDs) أو تقنيات النمذجة المشابهة. فهو يحدد الكيانات (الجدول)، والسّمات (الأعمدة)، والعلاقات بين الكيانات، مما يوفر نظرة عامة مفاهيمية لمخطط قاعدة البيانات.
- تعريفات الجدول: توفر الوثائق وصفًا تفصيليًا لكل جدول في قاعدة البيانات، بما في ذلك الأسماء وأنواع البيانات والقيود والعلاقات مع الجداول الأخرى. وهو يوثق الغرض واستخدام كل جدول والبيانات الوصفية المرتبطة به.
- قواعد سلامة البيانات: توضح الوثائق قواعد سلامة البيانات والقيود وقواعد التحقق من الصحة التي يفرضها مخطط قاعدة البيانات لضمان اتساق البيانات ودقتها وسلامتها. وهو يحدد الشروط التي بموجبها يُسمح بتعديلات البيانات أو يتم تقييدها.

• شرح تصاميم الواجهة:

- تصميم واجهة المستخدم: تصف الوثائق تصميم واجهة المستخدم (UI)، بما في ذلك الإطارات السلوكية والنماذج بالحجم الطبيعي ومواصفات التصميم. فهو يوضح التخطيط والعناصر المرئية وتدفق التنقل وأنماط التفاعل المستخدمة لإنشاء واجهة بديهية وسهلة الاستخدام.

- مواصفات واجهة برمجة التطبيقات (API): تحدد الوثائق الواجهات التي يعرضها النظام، مثل واجهات برمجة التطبيقات (APIs) للتفاعل مع الأنظمة أو الخدمات الخارجية. وهو يوثق نقاط النهاية وتنسيقات الطلب/الاستجابة وآليات المصادقة وإرشادات الاستخدام للمطورين الذين يتكاملون مع النظام.
- نقاط التكامل: تحدد الوثائق نقاط التكامل بين مكونات النظام أو الأنظمة الفرعية أو الأنظمة الخارجية، مع تحديد البروتوكولات وتنسيقات البيانات وآليات الاتصال المستخدمة لقابلية التشغيل البيئي. ويوضح كيفية تفاعل أجزاء مختلفة من النظام وتبادل البيانات لتحقيق الوظيفة المطلوبة.



وضح كيف تساهم تعليقات التعليمات البرمجية والوثائق المضمنة ورسائل الالتزام الخاصة بالتحكم في الإصدار في الحفاظ على وثائق العمل أثناء عملية الترميز:

ثالثاً: مرحلة التنفيذ:

مرحلة التنفيذ في تطوير البرمجيات هي المرحلة التي يتم فيها الترميز والتطوير الفعلي للبرنامج. خلال هذه المرحلة، يقوم المطورون بترجمة مواصفات التصميم إلى تعليمات برمجية قابلة للتنفيذ، ودمج مكونات النظام المختلفة، والتأكد من أن البرنامج يعمل وفقاً للمتطلبات المحددة. تتضمن هذه المرحلة الكتابة والاختبار وتصحيح الأخطاء ودمج التعليمات البرمجية لإنشاء منتج برمجي فعال. يعد التنفيذ الفعال أمراً بالغ الأهمية لتقديم حل برمجي موثوق وعالي الجودة يلبي احتياجات المستخدم وأهداف المشروع.

- تعليقات التعليمات البرمجية :

- شرح منطق التعليمات البرمجية:

مثال: في دالة تحسب مضروب رقم، قد يشرح التعليق النهج التكراري المستخدم وحالة الإنهاء.

```
python
# Calculate the factorial of a number using an iterative approach
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result
```

- توثيق النوايا:

مثال: قد يوضح تعليق السبب وراء اختيار تنفيذ محدد، مثل استخدام خوارزمية معينة لتحقيق الكفاءة.

```
javascript
// Using binary search for faster retrieval of elements in a sorted array
```

- أمثلة الاستخدام:

مثال: يمكن أن توفر التعليقات أمثلة حول كيفية استخدام الوظائف أو الأساليب داخل قاعدة التعليمات البرمجية.

```
java
// Example usage: calculate the sum of two numbers
int result = add(3, 5);
```

- الوثائق المضمنة:

- توثيق الوظيفة والطريقة:

مثال: قد تصف سلسلة المستندات الخاصة بالوظيفة في بايثون غرضها، ومعلماتها، وقيمتها المرجعة.

```
python
def calculate_area(length, width):
    """
    Calculate the area of a rectangle.

    Parameters:
        length (float): The length of the rectangle.
        width (float): The width of the rectangle.

    Returns:
        float: The area of the rectangle.
    """
    return length * width
```

- وثائق واجهة برمجة التطبيقات:

مثال: قد تصف الوثائق المضمنة في واجهة برمجة التطبيقات نقاط النهاية ومعلمات الطلب والاستجابات المتوقعة.

```
yaml
/api/v1/users:
  get:
    summary: Retrieve a list of users
    parameters:
      name: limit
      in: query
      description: Number of users to retrieve
      required: false
      type: integer
```

- المعلمة وأوصاف المتغير:

مثال: يمكن للتعليقات أو التعليقات التوضيحية أن تشرح غرض واستخدام معلمات الدالة أو المتغيرات.

```
typescript
function greet(name: string): void {
    // Print a greeting message with the provided name
    console.log(`Hello, ${name}!`);
}
```

- رسائل الالتزام بالتحكم في الإصدار:

- تغيير الأوصاف:

مثال: قد تصف رسالة الالتزام إضافة ميزة جديدة أو حل خطأ.

```
sql
Add user authentication feature
```

- إصلاحات الأخطاء والتحسينات:

مثال: يمكن أن تتضمن رسائل الالتزام تفاصيل الإصلاحات المطبقة على مشكلات معينة أو التحسينات التي تم إجراؤها على الوظائف الحالية.

```
csharp
```

```
Fix null pointer exception in login form validation
```

- التعاون والتواصل:

مثال: تعمل رسائل الالتزام كوسيلة للاتصال بين أعضاء الفريق، حيث توفر تحديثات حول تغييرات التعليمات البرمجية والتقدم.

```
sql
```

```
Update README with installation instructions
```

استكشف استخدام خطط الاختبار وحالات الاختبار وتقارير الأخطاء باعتبارها عناصر توثيقية أساسية لضمان الجودة.

رابعاً : مرحلة الاختبار:

تعد خطط الاختبار وحالات الاختبار وتقارير الأخطاء أدوات توثيقية أساسية في مجال ضمان الجودة (QA). دعونا نستكشف أهميتها في ضمان جودة المنتجات البرمجية :

• خطط الاختبار:

- **التعريف:** تحدد خطة الاختبار النطاق والنهج والموارد والجدول الزمني لأنشطة الاختبار. وهي بمثابة مستند شامل يوجه عملية الاختبار من البداية إلى النهاية.
- **الأهمية:** توفر خطط الاختبار خريطة طريق لفرق ضمان الجودة، وتحدد الأهداف والاستراتيجيات لاختبار منتج البرنامج. فهي تساعد على ضمان أن تكون جهود الاختبار جيدة التنظيم وشاملة ومتوافقة مع متطلبات المشروع والجدول الزمنية.
- **المكونات الرئيسية:** تتضمن خطط الاختبار عادةً أقساماً مثل:

أ. مقدمة ونظرة عامة

ب. أهداف الاختبار ونطاقه

ج. اختبار الاستراتيجية والنهج

د. بيئة الاختبار وأدواته

هـ. جدول الاختبار ومعالجه

و. المخاطر والطوارئ

ز. الأدوار والمسؤوليات

• حالات الاختبار:

- **التعريف:** حالات الاختبار هي تعليمات تفصيلية لتنفيذ اختبارات معينة للتحقق من وظائف منتج البرنامج وسلوكه. وهي تصف بيانات الإدخال والنتائج المتوقعة وظروف الاختبار لكل سيناريو اختبار.
- **الأهمية:** تعمل حالات الاختبار بمثابة وثائق قابلة للتنفيذ، مما يمكن فرق ضمان الجودة من التحقق بشكل منهجي من صحة البرنامج مقابل المتطلبات والمواصفات المحددة.
- **المكونات الرئيسية:** تتضمن حالات الاختبار عادةً عناصر مثل:

أ. معرف حالة الاختبار والعنوان

ب. وصف الاختبار وشروطه

ج. خطوات وإجراءات الاختبار

د. النتائج المتوقعة ومعايير القبول

هـ. اختبار البيانات والمدخلات

و. تكوين بيئة الاختبار

• تقارير الأخطاء:

- التعريف: تقوم تقارير الأخطاء، المعروفة أيضًا باسم تقارير العيوب أو تقارير الإصدار، بتوثيق أية حالات شاذة أو أخطاء أو تناقضات تم تحديدها أثناء الاختبار. وهي توفر معلومات تفصيلية عن طبيعة الخلل وتأثيره وخطوات إعادة إنتاجه.
- الأهمية: تعمل تقارير الأخطاء كأداة اتصال بين فرق ضمان الجودة وفرق التطوير، مما يسهل تحديد عيوب البرامج وفرزها وحلها. فهي تساعد في تتبع حالة المشكلات التي تم الإبلاغ عنها والتأكد من معالجة الأخطاء الحرجة على الفور.
- المكونات الرئيسية: تتضمن تقارير الأخطاء عادةً المعلومات التالية:
 - أ. معرف العيب والعنوان .
 - ب. وصف المشكلة
 - ج. سيناريو المشكلة (Bug scenario) .
 - د. النتائج الفعلية والمتوقعة
 - هـ. مستويات الخطورة والأولوية
 - و. تفاصيل البيئة
 - ز. المرفقات (على سبيل المثال، لقطات الشاشة والسجلات)



استعرض أدوات التوثيق الشائعة مثل Microsoft Word و Markdown و Confluence، مع تسليط الضوء على ميزاتها وحالات استخدامها:

4. أدوات وتقنيات توثيق المشاريع البرمجية

في تطوير البرمجيات، يتم استخدام أدوات وتقنيات مختلفة لإنشاء وإدارة وثائق المشروع. تتضمن بعض الأدوات الرئيسية أنظمة التحكم في الإصدار، ومولدات الوثائق، ومواقع الويكي ومنصات التعاون، وبيئات التطوير المتكاملة (IDEs). بالإضافة إلى ذلك، تتضمن التقنيات لغات الترميز، ومعايير التعليق على التعليمات البرمجية، وأدلة المستخدم، والأدلة الفنية. تضمن هذه الأدوات والتقنيات أن تكون الوثائق دقيقة، ويمكن الوصول إليها، وقابلة للصيانة طوال عملية تطوير البرمجيات.

• أدوات التوثيق:

• مايكروسوفت وورد (Microsoft Word):

- الميزات: Microsoft Word عبارة عن أداة متعددة الاستخدامات لمعالجة النصوص توفر نطاقًا واسعًا من الميزات لإنشاء المستندات وتنسيقها. كما يوفر مايكروسوفت وورد واجهة سهلة الاستخدام مع أدوات لتحرير النص وتنسيقه وتصميمه، بالإضافة إلى دعم الصور والجداول والرسوم البيانية. كما يقدم مايكروسوفت وورد أيضًا ميزات التعاون مثل تتبع التغييرات والتعليقات، مما يجعله مناسبًا لتحرير المستندات على أساس الفريق.
- حالات الاستخدام: يُستخدم برنامج Microsoft Word بشكل شائع لإنشاء أنواع مختلفة من المستندات، بما في ذلك التقارير والمذكرات والأدلة ووثائق المشروع. يحظى بشعبية خاصة في بيئات الأعمال لسهولة استخدامه وتوافقه مع تطبيقات Microsoft Office الأخرى.

• لغة الترميز (Markdown):

- الميزات: Markdown هي لغة ترميزية خفيفة الوزن تتيح التنسيق السهل للمستندات النصية العادية باستخدام بناء جملة بسيط. لقد تم تصميمها لتكون قابلة للقراءة من قبل الإنسان ويمكن تحويلها بسهولة إلى HTML أو تنسيقات أخرى. تدعم Markdown عناصر التنسيق الأساسية مثل العناوين والقوائم والروابط والتأكيد، مما يجعلها مثالية لإنشاء مستندات منظمة دون الحاجة إلى أدوات تنسيق معقدة.
- حالات الاستخدام: يتم استخدام Markdown على نطاق واسع لإنشاء الوثائق وملفات README و منشورات المدونات والملاحظات الفنية. تحظى بشعبية خاصة بين مطوري البرامج، لأنها تتيح كتابة الوثائق بسرعة وكفاءة مباشرة داخل مستودعات التعليمات البرمجية. يمكن التحكم في إصدار مستندات Markdown بسهولة ومشاركتها باستخدام منصات مثل GitHub أو GitLab.

Task	Markdown syntax
Heading 1	#
Heading 2	##
Heading 3	###
Italics	<i>"Italics"</i>
Bold	"Bold"
Strike	"insert text"
Block quote	>
Links	[link name] (link.com)
Unordered list	* List item * List item
Code Block	`insert code here`

شكل رقم. MARKDOWN 1-ERROR! NO TEXT OF SPECIFIED STYLE IN DOCUMENT. مثال على

- منصة كونفلونس (Confluence):
الميزات: Confluence عبارة عن منصة للتعاون والتوثيق تم تطويرها بواسطة Atlassian. وهي تقدم مجموعة واسعة من الميزات لإنشاء المحتوى وتنظيمه ومشاركته، بما في ذلك تحرير النص المنسق والقوالب ووحدات الماكرو وسجل الإصدارات. يتكامل Confluence بسلاسة مع منتجات Atlassian الأخرى مثل Jira وBitbucket، مما يمكن الفرق من تبسيط سير عمل التوثيق وعمليات إدارة المشاريع.
- حالات الاستخدام: يتم استخدام Confluence بشكل شائع لإنشاء وصيانة الوثائق الداخلية وقواعد المعرفة ووثائق المشروع ومواقع wiki الخاصة بالفرق. تعتبر Confluence منصة مركزية للفرق للتعاون في التوثيق ومشاركة المعلومات وتتبع التغييرات. تحظى منصة Confluence بشعبية خاصة بين فرق تطوير البرمجيات وأقسام تكنولوجيا المعلومات بسبب ميزاته القوية وقدرات التكامل.



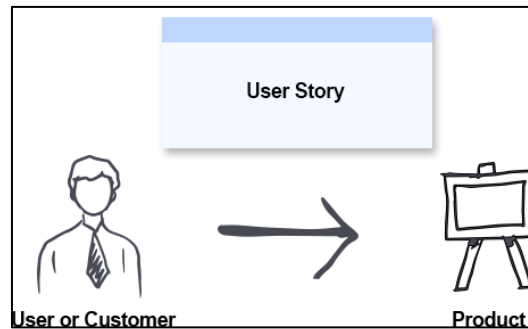
صف عملية توثيق متطلبات المستخدم من خلال حالات الاستخدام وقصص المستخدم، مع التركيز على التقاط احتياجات المستخدم وسلوكه

5. حالات الاستخدام وقصص المستخدم:

يعد توثيق متطلبات المستخدم من خلال حالات الاستخدام وقصص المستخدم خطوة حاسمة في تطوير البرمجيات، لأنه يساعد على التقاط احتياجات المستخدم وسلوكياته وتوقعاته بشكل فعال.

• ما هي قصة المستخدم؟

في تطوير البرمجيات وإدارة المنتجات البرمجية، قصة المستخدم هي وصف غير رسمي باللغة الطبيعية (عربية، انجليزية،) لاحتياجات المستخدمين. قصة المستخدم هي أداة تستخدم في تطوير البرمجيات للحصول وصف لميزة برمجية من وجهة نظر المستخدم النهائي. تصف قصة المستخدم نوع المستخدم وما يريده ولماذا. تساعد قصة المستخدم في إنشاء وصف مبسط لمتطلب النظام كما هو موضح بالشكل التالي.

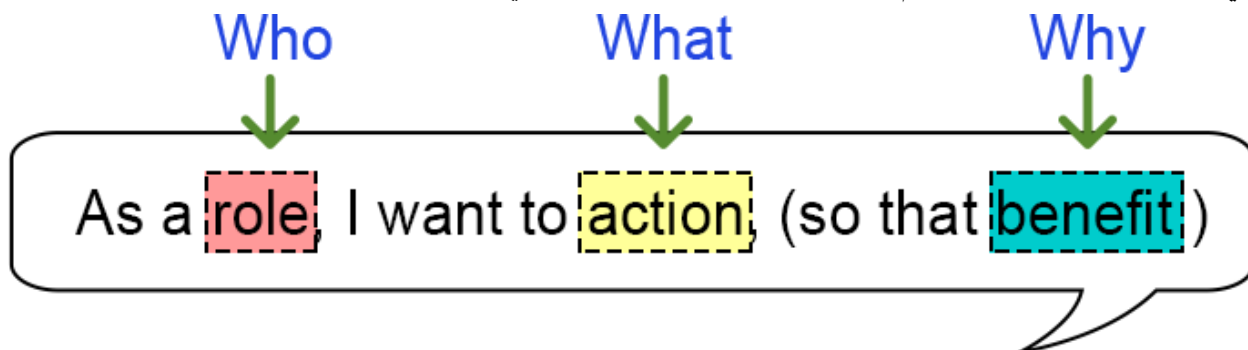


• قالب قصة المستخدم:

تكون قصص المستخدم العناصر الأساسية للمتطلب فقط:

- من هو المستفيد منها؟
- ما الذي تتوقعه من النظام؟
- لماذا هي مهمة (اختيارية؟)؟

فيما يلي تنسيق بسيط لقصة المستخدم يستخدمه 70% من المبرمجين والمطورين:



- فيما يلي عملية تفصيلية لتوثيق متطلبات المستخدم باستخدام هذه التقنيات:
- تحديد أصحاب المصلحة وجمع المتطلبات:
 - حدد جميع أصحاب المصلحة المشاركين في المشروع، بما في ذلك المستخدمين النهائيين والعملاء وأصحاب المنتجات والأطراف الأخرى ذات الصلة.
 - اجر المقابلات وورش العمل والاستطلاعات والملاحظات لجمع متطلبات المستخدمين، مع التركيز على فهم أهدافهم ومهامهم ونقاط الضعف والتفضيلات.
- تحديد حالات الاستخدام:
 - صف حالات الاستخدام والتفاعلات بين المستخدمين والنظام لتحقيق أهداف أو مهام محددة.
 - حدد الجهات الفاعلة الأساسية (المستخدمون أو الأنظمة التي تتفاعل مع التطبيق) وحدد أهدافهم.
 - وثق حالات الاستخدام باستخدام تنسيق موحد، بما في ذلك العنوان والوصف الموجز والشروط المسبقة والتدفق الرئيسي للأحداث والتدفقات البديلة والشروط اللاحقة.
 - حدد الخطوات المتبعة في كل حالة استخدام، بما في ذلك إجراءات المستخدم واستجابات النظام وأية مسارات شرطية أو استثنائية.
- تحديد الأولويات والتحقق من صحة حالات الاستخدام:
 - حدد أولويات حالات الاستخدام بناءً على أهميتها للمستخدمين وأهداف العمل وقيود المشروع.
 - تحقق من صحة حالات الاستخدام مع أصحاب المصلحة لضمان الدقة والاكتمال والمواءمة مع متطلبات المستخدم.
 - كرر تعريفات حالة الاستخدام بناءً على التعليقات وقم بدمج أية مراجعات أو تحسينات ضرورية.
- تجميع وتوثيق قصص المستخدم:
 - قصص المستخدم عبارة عن أوصاف موجزة تتمحور حول المستخدم لوظائف أو ميزات محددة للنظام، وعادةً ما تكون مكتوبة من منظور المستخدم النهائي.
 - قسم حالات الاستخدام إلى قصص مستخدم أصغر وأكثر قابلية للإدارة والتي تمثل وحدات منفصلة من الوظائف.
 - اتبع معايير INVEST (مستقلة، قابلة للتفاوض، ذات قيمة، قابلة للتقدير، صغيرة، قابلة للاختبار) للتأكد من أن قصص المستخدمين محددة جيدًا وقابلة للتنفيذ.

- اكتب قصص المستخدمين باستخدام تنسيق منظم، بما في ذلك دور المستخدم والإجراء والنتيجة (على سبيل المثال، "باعتباري [دور مستخدم]، أريد [الإجراء] حتى يتم الحصول على [النتيجة]")

- تفاصيل قصص المستخدم مع معايير القبول:

- حدد معايير القبول لكل قصة مستخدم لتحديد الشروط التي بموجبها يمكن اعتبار القصة كاملة ومرضية.
- حدد معايير القبول السلوك المتوقع والوظائف والنتائج التي يجب استيفاؤها حتى يتم قبول قصة المستخدم من قبل أصحاب المصلحة.
- ضمن معايير القبول الإيجابية (مثل سيناريوهات النجاح) والسلبية (مثل معالجة الأخطاء وحالات الحافة) لضمان التغطية الشاملة.