هيئة الاتصالات والفضاء والتقنية
Communications, Space & Technology Commission

الدليل الإرشادي

لمعايير جودة البرمجيات

أكتوبر 2024

cst.gov.sa

هيئة الاتصالات والفضاء والتقنية
Communications, Space & Technology Commission

قائمة المحتويات

1. مقدمة

انطلاقا مـــن اختصاص هيئة الاتصالات والفضاء والتقنية (الهيئة) بالتنظيم والرقابـــة على قطاع التقنية وفقا لتنظيمهـــا المُعدل بقرار مجلس الوزراء رقم (133) وتاريـــخ 1424/5/21هـ، الذي أوكل إليهـــا مهاما تنظيمية لقطاع تقنيـــة المعلومات، والتي أكد عليها البند (ســـابعا) من قرار مجلس الوزراء رقـــم (292) وتاريخ 1441/4/27هـ. وتباشـــر الهيئة تلـــك الصلاحيات مبتغية تحقيـــق أهـــداف تنظيم القطـــاع، والواردة فـــي المادة الثانيـــة من نظام الاتصالات وتقنية المعلومات (النظام)، الصادر بالمرســـوم الملكي الكريم رقم (م/106) وتاريخ 1443/11/2هـ، ومنها: تطوير تقنية المعلومات وتحسين الأســـواق ورفع نضج الخدمات المقدمـــة للمســـتفيدين النهائيين ووضع الإجراءات المناسبة لذلك.

وبناء عليه، قامت الهيئة بتطوير الدليل الارشادي لمعايير جودة البرمجيات (الدليل) لتعزيز ولتحســـين جودة البرمجيات. حيث أصبحت جودة البرمجيات مـــن العناصر التنافســـية المهمة في صناعة البرمجيـــات، وهو الأمر الذي أصبح لزاما على الشركات والجهات التي تسعى إلى الازدهار في الاقتصاد الرقمي كون البرمجيات عالية الجودة والموثوقة تعتبر عنصرا محوريا لنجاح التحول الرقمي. إضافة إلى ذلك، ترســـخ البرمجيات عالية الجودة ســـمعة المملكة فـــي تقديم حلول رقمية رائدة والتي تســـاهم في جعل المملكة مركزا رقميا رائدا إقليميا وعالميا.

يهـــدف هذا الدليـــل إلى وضع مجموعة من الممارســـات التـــي يمكن أن تتبناها الشـــركات أو الجهات التـــي تعمل على تطويـــر برمجيات أو تنفيذ حلول برمجية للحفاظ على جودة عاليـــة للبرمجيات. ويعد هذا الدليل دليلا استرشـــاديا، وليس بـــديلا لأي ضوابط أو إجـــراءات أو معاييـــر أو قواعد أو تعليمـــات أو لوائح صـــادرة بموجب قـــرار تنظيمي عن الهيئـــة أو الجهات الحكوميـــة ذات العلاقة، ولا يشـــكل بأي حال من الأحـــوال مرجعا لأي من الإجراءات والمسؤوليات القانونية المترتبة على الأشخاص والأطراف ذوي العلاقة.

2. التعريفات

فيما يلي قائمة بالتعاريف الرئيسية المستخدمة في هذه الوثيقة:

1.2 البرمجيات أو برمجيات الحاسوب Compute Software:

مجموعة من الأوامر والتعليمات المعبر عنها بأية لغة أو رمز أو إشارة، والتي تتخذ أي شكل من الأشكال، ويمكن استخدامها بطريقة مباشرة أو غير مباشرة في الحاسوب لأداء وظيفة أو تحقيق نتيجة، سواء كانت هذه الأوامر في شكلها الأصلي أو في أي شكل آخر تظهر فيه من خلال الحاسوب. ويمكن أن تكون هذه البرمجيات سحابية بحيث يتم الوصول إليها عبر الشبكة، أو مثبتة على الحاسوب أو الأجهزة اللوحية، أو مدمجة مثل المركبات المتصلة أو الأجهزة الطبية، أو غير ذلك.

2.2 وثائق البرمجيات Software Documents:

الوثائق اللازمة لتطوير أو تحليل أو استخدام أو فهم أو دعم البرمجيات والتي تشمل مجموعة الوثائق والمستندات المرفقة والمتعلقة ببرمجيات الحاسوب ومنها وثائق ومستندات مراحل التصميم والتحليل والتطوير، وثائق الاختبار والصيانة، أو خرائط تدفق العمليات، أو التصاميم، أو المخططات، أو الخوارزميات، ومنها أيضا إرشادات الاستخدام، وأي مستندات مساعدة على فهم أو تشغيل أو إعادة كتابة وصيانة البرمجيات. وتشمل مستندات التوثيق لكافة العمليات والإجراءات.

3.2 جودة البرمجيات Software Quality:

الملاءمة الوظيفية للبرمجيات بالمقارنة بوثائق البرمجيات، وكفاءة الأداء، والتوافق، وسهولة الاستخدام، والأمان والموثوقية، وقابلية الصيانة، وقابلية التكامل للبرمجيات، ويتم تحقيق ذلك من خلال الالتزام بالعمليات المخطط لها بشكل جيد، وإدارة المخاطر، وضمان رضا الأطراف المعنية، وتطبيق آليات التحسين المستمر طوال دورة حياة البرمجيات.

4.2 ضمان جودة البرمجيات Software Quality Assurance:

مجموعة من الأنشطة المستمرة لمراقبة جميع عمليات صناعة البرمجيات، لضمان الامتثال لمعايير الجودة المحددة بما يضمن أن جميع المشاركين في العمليات الأساسية لدورة حياة البرمجيات قد نفذوا جميع الإجراءات والعمليات بشكل صحيح وفعال.

5.2 المستفيد Beneficiary or User:

الشخص أو الأشخاص الطبيعيين أو الاعتباريين الذي لديهم مصلحة أو تأثير في استخدام النظام البرمجي، ويشمل ذلك المستخدمين والعملاء الذين يستخدمون البرمجيات لتحقيق أهداف معينة، بالإضافة إلى الجهات ذات العلاقة مثل الشركاء والموردين، وأصحاب المصلحة مثل المساهمين والمديرين والملاك، حيث يساهم فهم احتياجات وتوقعات هؤلاء المستفيدين في توجيه عملية التصميم والتطوير لضمان تحقيق نجاح المشروع البرمجي.

6.2 مطوري البرمجيات Software Developers:

الشخص أو الأشخاص الطبيعيين أو الاعتباريين المتخصصون في تصميم، تطوير، أو صيانة البرمجيات المخصصة لتلبية احتياجات محددة للأعمال ويشمل ذلك الجهات التي تطور البرمجيات داخليا.

7.2 المتطلبات الوظيفية Functional Requirements:

قدرة البرمجيات على تلبية المتطلبات والمواصفات المحددة في وثائق البرمجيات من أجل تلبية احتياجات المستفيد أو توقعاته (مثل: أن يكون هناك صفحة إلكترونية لتسجيل الزوار).

8.2 المتطلبات غير الوظيفية Non-Functional Requirements:

المواصفات التي تحدد الخصائص والقيود التي يجب أن يمتلكها النظام البرمجي، والتي لا تتعلق بالوظائف المحددة أو السلوك المباشر للنظام. هذه المتطلبات تركز على الجودة والأداء، والتي تغطي كذلك الاحتياجات المتبقية التي لا تغطيها المتطلبات الوظيفية (مثل: قدرة النظام على معالجة 1000 طلب في الثانية).

9.2 الأداء Performance:

كفاءة وسرعة استجابة البرمجيات لأوامر المستفيد.

10.2 الموثوقية Reliability:

قدرة البرمجيات على العمل بشكل مستقر ومتوقع دون وقوع أخطاء أو أعطال غير متوقعة.

11.2 التوافقية والتكامل Interoperability and Compatibility:

قدرة البرمجيات على التفاعل مع مختلف المكونات والأنظمة الأخرى بشكل سلس وبدون صعوبات.

12.2 الأمان والموثوقية Security and Reliability:

تمتع البرمجيات بالحماية بشكل جيد بما يضمن سلامة وسرية البيانات وضمان منع الوصول غير المصرح به.

13.2 الصيانة Maintenance:

القدرة على التعديل وتحديث البرمجيات بسهولة وفعالية دون التأثير على الوظائف الأساسية للنظام.

14.2 قابلية النقل Portability:

القدرة على نقل البرمجيات من بيئة أو نظام إلى آخر.

15.2 قابلية الاستخدام Usability:

سهولة استخدام البرمجيات وتعلمها.

16.2 التوافر Availability:

توفر البرمجيات واستمرارية تشغيلها لضمان الوصول للخدمة.

17.2 جمع المتطلبات Requirements Gathering:

عمليــة ممارســة البحث واستكشــاف وتعريــف متطلبــات النظام الوظيفية وغير الوظيفية من المســتفيد والعملاء وأصحاب المصلحة الآخرين.

18.2 تصميم البرمجيات Software Design:

عملية إنشــاء وتحديد الهيكل المعماري والسلوك والمكونات لنظام برمجي. يتضمن ترجمــة المتطلبات واحتياجات المســتفيد إلى خطة توجيهية توجه عملية التطوير.

19.2 تطوير البرمجيات Software Development:

يقصــد بــه العمليــة الفعلية لإنشــاء نظــام برمجي مــن مواصفات التصميم. يتضمن كتابة الأوامر (كــود)، ودمج المكونات، وإجراء العديد من أنشــطة الاختبــار والتصحيح للتأكد مــن أن نظــام البرنامج يعمل ويتوافق مع المتطلبات المقصودة.

20.2 اختبار البرمجيات Software Testing:

عمليـــة التقييم والتحقق من أن وظائـــف البرمجيات تقوم بما يفترض القيام به بناء على وثائق البرمجيات ذات العلاقة، وعملية التحقق من الأخطاء والفجـــوات وما إذا كانت نتيجة البرمجيـــات تطابق التوقعات المرغوبة قبل تثبيتها وتشغيلها.

21.2 نشـــر وتشـــغيل وصيانة البرمجيات Deployment, Operation and Maintenance :

مجموعة الأنشـــطة والعمليـــات المتضمنة فـــي إطلاق وإدارة أنظمة البرامج. بالإضافة إلى الأنشـــطة والعمليات لصيانة ودعم المنتج بعد الاطلاق.

3. نطاق الدليل

يمكن أن يســتفيد من هذه الوثيقة أي شــخص طبيعي أو اعتباري يمارس صناعة البرمجيات في المملكة العربية الســعودية. ويندرج تحت ممارســة صناعة البرمجيات: الاســتخدام، التطويــر، الاختبار، وغيرهــا من العمليات الأساسية في البرمجيات.

4. أهداف الدليل

1.4 **مساعدة مطوري البرمجيات والمستفيدين في تنفيذ ضمان جودة** واختبار البرمجيات الفعال بحيث يتوافق مع الضوابط والمعايير المحلية والدولية.

2.4 **الترويج لاعتماد أفضل الممارسات في جودة البرمجيات لتحسين** وظائفها وأمانها وموثوقيتها.

3.4 **تحسين رضا العملاء** من خلال التأكد من أن البرمجيات تلبي متطلبات المستفيد وتوقعاته.

4.4 **التخفيف من المخاطر المالية والقانونية أو المخاطر المتعلقة بالسمعة** التي قد تنتج عن الأخطاء ونقاط الضعف بالبرمجيات.

5.4 **تعزيز نمو وتطوير صناعة البرمجيات في المملكة العربية السعودية**، لتكون قادرة على تقديم منتجات برمجية عالية الجودة إلى الأسواق المحلية والدولية.

5. متطلبات الجودة خلال دورة حياة البرمجيات

تم تطويـــر هذه الإرشـــادات، التي يمكـــن اتباعهـــا خلال دورة حياة تطوير البرمجيات، بما يتسق مع المعايير والأطر الدولية الرائدة بما في ذلك " ISO 25000" و "ITIL 4" و "CMMI" و "COBIT 2019" وغيرهـــا، وتـــم تعريـــف الارشـــادات لـــكل مرحلة فـــي دورة حياة البرمجيـــات عبر المراحـــل التالية والموضحة في الشكل رقم 1:

شكل رقم 1: دورة حياة البرمجيات

مع التأكيد على أهمية التقيد بالتنظيمـــات المتعلقة بالأمان والموثوقية وإدارة وحوكمـــة البيانات الصـــادرة من الهيئة الوطنية للأمن الســـيبراني ومكتب إدارة البيانات الوطنية خلال دورة حياة تطوير البرمجيات.

1.5 مرحلة جمع المتطلبات

خلال مرحلة جمع المتطلبات، يمكن لمطوري البرمجيات والمستفيدين تبني الممارسات التالية حال انطباقها:

1.1.5 تعريف جميع الأطراف المعنية بالمنتج البرمجي والتي قد تشمل أصحاب المصلحـــة والمســـتفيدين النهائيين ومديري الأقســـام وأي أنظمة خارجية سيتفاعل معها البرنامج.

2.1.5 تنفيذ مقابلات واستطلاعات لاستخلاص معلومات مفصلة حول الغرض المتوقـــع والمواصفات للمنتج البرمجي بحيث يتم اســـتخدام منهجيات أو اساليب مثل الاجتماعات، والاستطلاعات، أو الاستبيانات التي تشمل أصحاب المصلحة المحددين.

3.1.5 في حال أن المنتج البرمجي المقترح مصمم لاستبدال أو التعامل مع أنظمة أو تطبيقات قائمة؛ يجب تحليل تلك الأنظمة أو التطبيقات بعمق وفهم قوتها التشغيلية وعيوبها والوظائف المطلوبة.

4.1.5 توثيق المتطلبات التي تم جمعها بطريقة واضحة وبشكل شامل ومستمر.

5.1.5 إنشاء نماذج أولية لمحاكاة التصميم وتحسينه حيث يمكن أن تساعد النمذجة في الكشف عن أخطاء التصميم وقابلية الاستخدام وتزويد أصحاب المصلحة بتمثيل مرئي للمنتج النهائي.

6.1.5 تحديد أولويات المتطلبات من خلال العمل المشترك مع أصحاب المصلحة لتحديد الأهمية النسبية لكل متطلب، مع أخذ عوامل مثل القيمة التجارية أو المخاطر المرتبطة بعين الاعتبار.

7.1.5 المراجعة والتحقق من المتطلبات بشكل دقيق مع جميع أصحاب المصلحة لضمان دقة وقابلية التطبيق قبل الانتقال إلى المرحلة التالية.

8.1.5 قد تتغير المتطلبات مع مرور الوقت وتطبيق نهج التطوير المرن "Agile"، من خلال أدوات (مثل سكرم "Scrum" أو كانبان "Kanban")، لتمكين التطوير التكراري والمرن يمكن استيعاب هذه التغييرات ويضمن أن البرمجيات الناتجة تبقى متوافقة مع احتياجات المستفيد، وينطبق النهج المرن على جميع مراحل دورة حياة التطوير وليس فقط مرحلة جمع المتطلبات

2.5 مرحلة تصميم ومعمارية البرمجيات

خلال مرحلة تصميم ومعمارية البرمجيات، يمكن لمطوري البرمجيات والمستفيدين تبني الممارسات التالية حال انطباقها:

1.2.5 التصميم المعماري والذي يتضمن تحديد البنية العامة للنظام البرمجي، بما في ذلك المكونات على المستوى العالي وتفاعلاتها وتوزيع الوظائف.

2.2.5 تصميم المكونات والذي يتضمن تقسيم النظام إلى مكونات أصغر وأكثر قابلية للإدارة وتحديد واجهات وتبادل البيانات بينها.

3.2.5 تصميم واجهة المستخدم والتي تتضمن تصميم واجهة المستخدم الرسومية (UI) أو تجربة المستخدم (UX) للنظام البرمجي.

4.2.5 تصميم قاعدة البيانات وتشـمل تصميم هيـكل وتنظيم قاعدة البيانات التي يستخدمها النظام البرمجي.

5.2.5 تصميـم الخوارزميـات والتي تنطوي على تصميـم خوارزميات وهياكل بيانات لحل مشكلات محددة أو تحقيق وظائف محددة بكفاءة وذلك يشـمل تحديد الخوارزميات والهياكل البيانات المناسـبة وفقا للمتطلبات واعتبارات مثل تعقيد الوقت وتعقيد المساحة.

6.2.5 تصميم آليات وتدابير الأمان والموثوقية لحماية النظام البرمجي مـن الوصول غير المصرح بـه وانتهاكات البيانـات وتهديدات الأمان والموثوقية الأخرى.

7.2.5 تصميم اسـتراتيجيات وآليات للتعامل مع الأخطاء والاستثناءات والحالات غير المتوقعة بشكل لائق وذلك يشمل تحديد آليات التعامل مع الاستثناءات وتسجيل الأخطاء وإبلاغها.

8.2.5 تصميم التكامل والذي يشـمل تصميم واجهات ونقاط التكامل مـع أنظمة أو مكونـات أخرى وذلك يشـمل تحديد تنسـيقات تبادل البيانات وبروتوكولات الاتصال وواجهات البرمجة الفائقة.

9.2.5 إشـراك أصحـاب المصلحة، تحديـدا أصحاب الأعمـال والعملاء والخبـراء، طوال مرحلـة التصميم وذلك لضمـان معالجة الملاحظات والتوقعـات، والتقليل من احتمالية حـدوث تغييرات كبرى في وقت متأخر من دورة التطوير.

10.2.5 إجراء مسـح شـامل للسـوق المسـتهدف لفهمه، وشـريحة العملاء، والتوجهات، والمشـهد التنافسـي بهـدف تحليل منتجات المنافسين لتحديد الثغرات وفرص المنافسة.

11.2.5 تحديـد نمـوذج الأعمال للمنتـج وكيفية خلـق القيمة المضافة والإيرادات واكتساب العملاء والحفاظ على الربحية مع اعتبار العوامل الأخرى مثل تكاليف التسويق للعملاء، واستراتيجية التسعير، وقنوات التوزيع، والشراكات.

12.2.5 تحديد نموذج للأرباح والإيرادات بحيث يتوافق مع أهداف العمل والسوق المستهدف مع اعتبار عدد من الخيارات مثل النماذج القائمة على الاشتراكات، أو الشراء لمرة واحدة، أو النماذج المجانية، أو النماذج القائمة على الإعلانات، وذلك مع تحليل إيجابيات وسلبيات كل نموذج واختر الأسلوب الذي يناسب المنتج والفئة المستهدفة.

13.2.5 إجراء دراســة جدوى لتقييم الجدوى الفنية والاقتصادية للمنتج المقتــرح، والتأكد من أن تصميم المنتج ســهل الاســتخدام مع اعتبار العوامل الاخرى مثل ســهولة التنقل والتصميم والاستجابة والتوافق مع الأجهزة المختلفة والتقنيات المســاعدة. ينبغــي أيضا إجراء اختبار قابلية الاســتخدام مع مســتخدمين حقيقيين للإفادة حــول التصميم وتحديد مواضع التحســين، والتحقق من أن البرمجيات تلبي احتياجات المستفيدين.

14.2.5 تحديد نموذج تملك المنتج البرمجي، والذي يشــمل عدة خيارات كالمصدر المفتوح أو الملكية (Source Code) أو كلاهما. وتقييم المزايا والسلبيات والتأثير على الإيرادات واعتماد السوق المرتبط بكل نموذج ترخيص.

15.2.5 الامتثال بالأنظمة ولوائح حماية البيانات ذات العلاقة والمعايير الدولية وتنفيذ تدابير أمان وموثوقية عالية المســتوى لحماية بيانات المستفيد ومعالجة نقاط الضعف المحتملة.

16.2.5 مراعــاة التكامــل الســلس مع المنصــات أو الأنظمــة وتحديد واجهــات برمجة التطبيقــات أو البروتوكولات القياســية التي تمكن عمليات التكامل مع مكونــات البرمجيات الخارجية أو خدمات الجهات الخارجية والذي يعزز القيمة المعروضة للمستخدم النهائي.

17.2.5 تصميم منتج البرمجيات بحيث يحظى بقابلية التوسع لاستيعاب الزيــادة المحتملة في قاعدة المســتفيدين وحجم البيانات وتحســين الأداء من خلال تنفيــذ خوارزميات فعالة، واســتخدام هياكل البيانات المناســبة، ومراعاة قيود النظام مثل أوقات الاســتجابة واســتغلال الموارد.

18.2.5 مراعاة متطلبات "توطيــن" منتج البرمجيات في وقت مبكر من عمليــة التصميم في حال اســتهداف الأســواق العالمية مــع اعتبار التفضيلات اللغوية والثقافيــة والإقليمية ودعم العملات، والمناطق الزمنية، واللوائح المحلية المختلفة في حال انطباقها.

3.5 مرحلة تطوير البرمجيات

خلال مرحلة تطوير البرمجيات، يمكن لمطوري البرمجيات والمستفيدين تبني الممارسات التالية حال انطباقها:

1.3.5 يجب كتابة التعليمات البرمجية بطريقة يسهل فهمها دون الحاجة إلى تعليقات خارجية موسعة. يمكن أن يساعد في ذلك التسمية الجيدة للمتغيرات وتقسيم الوظائف الكبيرة إلى وظائف أصغر. كلما كانت التعليمات البرمجية أبسط وأكثر وضوحا، كان من الأسهل فهمها وتصحيح أخطائها وصيانتها.

2.3.5 الالتزام بأفضل الممارسات للغة البرمجة المستخدمة يجعل التعليمات البرمجية أكثر كفاءة وقابلية للصيانة بحيث أن كل لغة برمجة لديها دليل لأفضل الممارسات.

3.3.5 استخدام نظام التحكم في الإصدارات بشكل فعال، مثل "Git"، لإدارة كود المصدر وتتبع التغييرات حيث يتيح ذلك التعاون بين المطورين، ويسهل مراجعة التعليمات البرمجية، ويساعد على منع تعارض التغييرات على التعليمات البرمجية.

4.3.5 يجب الحفاظ على تحديث أطر العمل واستخدام النسخ الحديثة منها حيث تصدر أطر العمل تحديثات بانتظام، والتي تتضمن عادة تحسينات في الأداء ومميزات جديدة وتصحيحات أمان وموثوقية مهمة.

5.3.5 القيام بإجراء مراجعات منتظمة لكود المصدر لتحديد المشاكل المحتملة ومعالجتها، وضمان الامتثال بمعايير البرمجة، وتحسين الجودة بشكل عام ويمكن أن تساعد أدوات مراجعة الكود مثل "Gerrit" أو "Crucible" في تسهيل هذه العملية.

6.3.5 اتباع أفضل الممارسات في استخدام الأكواد مفتوحة المصدر والتي تشمل -على سبيل المثال لا الحصر- مراجعة تراخيص الاكواد مفتوحة المصدر، إجراء مراجعة أو تقييم الأمان والموثوقية للأكواد مفتوحة المصدر، الإشارة إلى الكتاب الأصليين والتراخيص في المنتج البرمجي.

7.3.5 تطبيق ممارسات الأمان والموثوقية في جميع عمليات التطوير، والتي تتضمن مراجعات التعليمات البرمجية لثغرات الأمان والموثوقية، واستخدام ممارسات البرمجة الآمنة، وتنفيذ أطر العمل ومعايير الأمان والموثوقية بما يتوافق مع متطلبات الأمان والموثوقية في تطوير البرمجيات الصادرة عن الهيئة الوطنية للأمن السيبراني.

8.3.5 تطبيق تقنيات وأساليب معالجة الأخطاء والاستثناءات من خلال استخدام آليات التسجيل "Logging" المناسبة لالتقاط الأخطاء والاستثناءات أثناء تشغيل البرمجيات لاستكشاف وتصحيح الأخطاء.

9.3.5 توثيق عملية تطوير البرمجيات، بما في ذلك قاعدة التعليمات البرمجية والوظائف وواجهات برمجة التطبيقات والتكوينات "Config-uration" حيث يساعد التوثيق في فهم النظام وصيانته واستكشاف الأخطاء وإصلاحها في المستقبل والتعاون مع أعضاء الفريق الآخرين.

10.3.5 كتابة ملفات البناء "Build files" حيث تعمل على تبسيط عملية الإنشاء وتوحيدها، مما يوفر الوقت ويقلل الأخطاء ويضمن إمكانية إنشاء التعليمات البرمجية واختبارها وتوزيعها بشكل متناسق عبر البيئات.

4.5 مرحلة اختبار البرمجيات

خلال مرحلة اختبار البرمجيات، يمكن لمطوري البرمجيات والمستفيدين تبني الممارسات التالية حال انطباقها:

1.4.5 إنشاء فريق مسؤول عن ضمان الجودة ومهام الاختبار حيث ينبغي أن يكون هذا الفريق مستقلا عن فريق التطوير لتجنب تضارب المصالح المحتمل وضمان إجراء اختبار محايد.

2.4.5 في حال الاستعانة بطرف خارجي لتطوير المشاريع عالية التكلفة أو الخطورة؛ ينبغي على هذه الجهات تحديد طرف ثالث مستقل عن مقدم خدمة التطوير للقيام بمهام ضمان الجودة والاختبار.

3.4.5 تحديد منهجية/دليل اختبار واضحة وموثقة تحدد نهج الاختبار والأهداف والمنهجيات التي ينبغي اتباعها، وينبغي ايضا أن يغطي هذا الدليل أنواعا مختلفة من الاختبارات، مثل الاختبار الوظيفي، واختبار الأداء، واختبار والأمان والموثوقية، واختبار وقابلية الاستخدام.

4.4.5 إنشاء اجراء للاختبارات محدد وموحد يتضمن أنشطة محددة مثل تخطيط الاختبار وتصميم الاختبار وتنفيذ الاختبار وإعداد تقارير الاختبار بحيث يضمن التناسق والتكرار في عمليات الاختبار.

5.4.5 الاستفادة من أدوات وأطر الأتمتة بحيث يتم أتمته حالات الاختبار المتكررة والتي تتطلب وقتا كبير لإتمامها وبذلك يتم تنفيذ الاختبارات بشكل أسرع وأكثر موثوقية، عبر التقليل من الأخطاء البشرية، ورفع نطاق تغطية الاختبار.

6.4.5 إعداد بيئة اختبارية مستقلة تكون مطابقة في الخصائص الفنية تماما لبيئة الرئيسية (البيئة المستضيفة للأنظمة المطورة) بحيث يسمح هذا للمختبرين بإجراء اختبار واقعي وتحديد أي مشاكل أو تعارض قد ينشأ في البيئة الرئيسية.

7.4.5 تنفيذ الاختبارات في وقت مبكر من دورة حياة التطوير بحيث يفضل إجراء الاختبارات بالتوازي مع التطوير للمساعدة في تحديد المشاكل في مرحلة مبكرة، بحيث يهدف هذا لتقليل التكلفة والجهد اللازمين لإصلاحها.

8.4.5 تبني أنظمة تتبع الأخطاء لكشف وتتبع جميع المشاكل خلال عملية الاختبار وذلك ليساعد في تنظيم وتحديد أولويات حل الأخطاء ويضمن عدم تخطي أي منها.

9.4.5 تضمين البيانات في البيئة الاختبارية لمحاكاة سيناريوهات الاستخدام الفعلي بحيث يساعد هذا في الكشف عن المشاكل المحتملة المتعلقة بسلامة البيانات والأداء وقابلية التوسع.

10.4.5 إشراك المستفيدين النهائيين أو العملاء المحتملين في عملية الاختبار حيث يمكنهم تقديم ملاحظات ذات قيمة عالية وتحديد مشاكل قابلية الاستخدام والمساعدة في التحقق مما إذا كان البرمجيات تلبي متطلباتهم وتوقعاتهم. ويمكن إعداد بيئة اختبارية مستقلة للمستفيدين النهائيين أو العملاء المحتملين في عملية الاختبار مطابقا للبيئة الاختبارية المعدة سابقا وذلك لاختبارات قابلية المستفيد على هذه البيئة.

11.4.5 ينبغي القيام بإجراء اختبار تراجع شامل (Roll Back) لضمان بقاء الوظائف غير متأثرة عند إجراء تحديثات أو تغييرات على البرمجيات، ويساعد اختبار التراجع في اكتشاف أي أخطاء جانبية غير مقصودة أو أخطاء قد تكون حدثت أثناء التطوير.

12.4.5 تحديد أولويات تنفيذ الاختبارات بناء على المخاطر المحددة وأهمية مكونات النظام حيث يساعد في تخصيص الموارد بشكل فعال والتركيز على المكونات التي لها تأثير عالي على النظام.

13.4.5 استخدام ممارسات التكامل المستمر لدمج تغييرات التعليمات البرمجية بانتظام وتشغيل الاختبارات الآلية تلقائيا للتعرف على مشاكل التكامل مبكرا ويضمن الاختبار المنتظم طوال عملية التطوير.

14.4.5 تشجيع المشاركة المبكرة لفريق الاختبار في مرحلة جمع وتحليل المتطلبات وذلك لتحديد المشاكل المحتملة في وقت مبكر وضمان إمكانية اختبار متطلبات البرمجيات.

15.4.5 تتبع وقياس نطاق الاختبار للتأكد من أن جميع الوظائف وحالات الاستخدام تم اختبارها بدقة، وذلك من خلال شمولية الاختبار على كل من التالي الكود، والمتطلبات، وتحليل المخاطر.

16.4.5 إنشاء إجراء لإدارة بيانات الاختبار بشكل فعال بتحديد وتوفير بيانات الاختبار المناسبة، وضمان خصوصية البيانات وأمانها وموثوقيتها، وتحديث بيانات الاختبار بانتظام للحفاظ على إمكانية تكرار الاختبار وفعاليته.

17.4.5 تنفيذ اختبار الأداء للتحقق من قدرة البرمجيات على التعامل مع زيادة الحمل والضغط المتوقع من قبل المستفيدين ليساعد في تحديد حدود القدرة ومعرفة نطاق الاختناقات الممكنة في الأداء ومشاكل قابلية التوسع والقيود على الموارد.

18.4.5 تضمين اختبار الأمان والموثوقية كمرحلة اختبار مخصصة لتحديد نقاط الضعف والمخاطر المتعلقة بأمن وموثوقية البيانات، والمصادقة، والتراخيص، والتشفير.

5.5 مرحلة نشر وتشغيل وصيانة البرمجيات

خلال مرحلة نشر وتشغيل وصيانة البرمجيات، يمكن لمطوري البرمجيات والمستفيدين تبني الممارسات التالية حال انطباقها :

1.5.5 تنفيذ عمليات نشر مؤتمتة لتبسيط نشر حزم البرمجيات وذلك باستخدام أدوات مثل "Docker" أو "Kubernetes" أو "Ansible" لإنشاء عمليات النشر وتسهيل أتمتتها.

2.5.5 استخدام أدوات إدارة التهيئة مثل "Puppet" أو "Chef" أو "Ansible" لإدارة وأتمتة إعدادات التهيئة لبيئات مختلفة.

3.5.5 تطبيق نظام إدارة الإصدارات والإطلاقات لإدارة إصدارات وإطلاقات المنتج البرمجيات بكفاءة وذلك باستخدام أدوات (مثل "GitHub" أو "GitLab" أو "Bitbucket") لإدارة التحكم في الإصدار والإطلاق للنسخ البرمجيات.

4.5.5 تنفيذ التكامل المستمر من مرحلة التطوير لأتمتة عملية نشر حزم البرمجيات لبيئة الرئيسية بحيث يضمن عملية نشر سلسة ومتوافقة.

5.5.5 تضمين خطط التعافي من الكوارث، والنسخ الاحتياطي للبرمجيات والبيانات وذلك لحمايتها من حالات الفشل أو الكوارث المحتملة وذلك باستخدام أدوات النسخ الاحتياطي وأنظمة التخزين الموزعة لضمان تكامل البرمجيات البيانات وتوافرها.

6.5.5 استخدام أدوات المراقبة، لمراقبة أداء البرمجيات ومدى واستغلالها للموارد وتجربة المستفيد لتحسين الأداء باستمرار بناء على ملاحظات المراقبة.

7.5.5 ضمان الإطلاق والنشر الآمن من خلال تنفيذ ممارسات النشر الآمنة، مثل بروتوكولات النقل الآمن (مثل: HTTPS, SFTP) وتشفير البيانات الحساسة عبر استخدام أدوات الفحص الأمان والموثوقية (مثل: "SonarQube" أو "OWASP ZAP") لتحديد ومعالجة نقاط الضعف.

8.5.5 استخدام آليات التقاط وتسجيل عالية المستوى في البيئة الرئيسية، لضمان إمكانية لتحليل وتصحيح الأخطاء والمشاكل واستكشاف المشاكل وإصلاحها عبر تطبيق أساليب معالجة الأخطاء والاستثناءات المناسبة.

9.5.5 إجراء اختبار قبول المستفيد قبل نشر البرمجيات بمشاركة المستفيدين النهائيين في اختبار البرمجيات وتقديم الملاحظات لضمان تجربة مستخدم سلسة.

10.5.5 توفير وثائق شاملة لإجراءات نشر حزم البرمجيات، بما في ذلك إعدادات التهيئة وأدلة استكشاف الأخطاء وإصلاحها. وقنوات دعم للمستخدم، مثل البريد الإلكتروني أو الدردشة، لمساعدة المستفيدين النهائيين في حالة حدوث مشاكل.

11.5.5 تطوير وثائق شاملة ومواد تدريبية لمساعدة المستفيدين النهائيين والمسؤولين والمطورين ويتضمن ذلك أدلة المستفيد، ووثائق الواجهات البرمجية للمنتج، وأدلة استكشاف الأخطاء وإصلاحها، والبرامج التعليمية لضمان حصول المستفيدين على الموارد اللازمة لاستخدام المنتج وصيانته بشكل فعال.

12.5.5 ضمان التزام نظام البرنامج بالمعايير التنظيمية والصناعية ولوائح خصوصية البيانات والسياسات التنظيمية وقد يشمل ذلك إجراء عمليات تدقيق منتظمة وتنفيذ تدابير الأمان والموثوقية واتباع أفضل الممارسات.

13.5.5 إنشاء فريق خدمة عملاء مخصص يمكن الوصول إليه من خلال قنوات متعددة مثل الهاتف والبريد الإلكتروني والدردشة ووسائل التواصل الاجتماعي، يتم من خلالها تقديم ردود سريعة وفعالة، ينبغي أيضا التأكد من أن ممثلي الدعم مدربون جيدا على تقديم الدعم بشكل اجتماعي ودقيق.

14.5.5 إنشـــاء قواعد للمعرفة الشاملة أو للأســـئلة الشائعة (FAQs) أو مقاطع الفيديو التعليمية أو أدلة المســـتفيد لمســـاعدة العملاء في العثور على إجابات للمشاكل الشائعة بحيث يسهل الوصول إلى مواد المســـاعدة الذاتيـــة هذه من موقـــع الويب الخاص بالشـــركة أو بوابة الدعم.

15.5.5 إنشـــاء نظـــام التذاكر للشـــكاوى بحيـــث يتيح للعملاء إرســـال مشاكلهم أو استفساراتهم وتتبع تقدمها ليساعد الشركة على تنظيم طلبات الدعم وتحديد أولوياتها، مما يضمن معالجة جميع استفسارات العملاء بكفاءة وفي الوقت المناسب.

16.5.5 تقديم الدعم الاســـتباقي للمشـــاكل المحتملـــة والتواصل مع العـــملاء قبل أن يضطروا إلى الاتصـــال بالدعم حيث يتم ذلك من خلال المراقبة الاســـتباقية لحسابات العملاء أو التحليل أو التواصل المنتظم لتقديم التحديثات أو المساعدة اللازمة.

17.5.5 جمـــع الملاحظات بانتظام من العملاء لتحديد مواضع التحســـين في عمليات الدعم وذلك عبر إجراء استطلاعات الرأي أو مراقبة وسائل التواصـــل الاجتماعي أو تحليل تفاعلات العـــملاء لفهم نقاط الضعف وإجراء التعديلات اللازمة لتحسين تجربة الدعم.

18.5.5 تحديد إجراءات تصعيـــد واضحة للحالات التي لا يمكن حلها على الفـــور بحيث يضمن توجيـــه المشـــاكل المعقدة فورا إلـــى فرق دعم متخصصة أو ممثلي دعم ذو مستوى أعلى لتقديم دعم أفضل.

19.5.5 إنشاء قنوات تواصل داخلية ومنصات لمشاركة المعرفة للسماح لممثلـــي الدعـــم بالتعاون والتعلـــم من تجارب بعضهـــم البعض وحل مشـــاكل العملاء الصعبة بشكل تعاوني، ويمكن القيام بذلك من خلال اجتماعـــات للفريق أو منتديات المناقشـــة أو توثيـــق الوثائق الداخلية للعم الفني.

20.5.5 توفير التدريب المســـتمر وفرص تطوير لدعـــم الممثلين لتعزيز معرفتهم بالمنتجات البرمجية، ومهـــارات خدمة العملاء، وقدرات حل المشاكل بحيث يمكنهم ذلك من تقديم دعم أكثر فعالية والبقاء على اطلاع دائم بأي تغييرات في المنتج أو السياسة.

21.5.5 تقديم دعم متعدد اللغات لتعزيز تجربة الدعم ورضا العملاء.

6. حوكمة البرمجيات

تشـير حوكمة البرمجيات إلى العمليات والسياسـات والضوابط التي يتم تبنيها لضمان تناسـق الأنشـطة المتعلقة بتطوير البرمجيات وإدارتها مع أهداف المسـتفيد والمعايير المحلية والدولية والمتطلبـات التنظيمية. تشـمل حوكمـة البرمجيـات مختلف جوانـب تطوير البرمجيات ونشـرها وصيانتها، ومنها إدارة المشـاريع واتخاذ القرارات وإدارة المخاطر، والأمان والموثوقيـة، والامتثـال للقوانيـن. ويشـمل الإطـار الفعـال لحوكمة البرمجيات ما يلي:

1.6 السياسات والإجراءات:

وضع الإرشـادات والمعايير لتطوير البرمجيات وتنفيذها وصيانتها، ومنها اختيـار البرمجيات والهندسـة المعمارية والتصميـم، والبرمجة، والاختبار، والوثائق.

2.6 الأدوار والمسؤوليات:

تحديد الأدوار والمسـؤوليات لضمان المراقبة السـليمة والمسائلة على مرحلة تطويـر البرمجيات بأكملهـا، مثل: مهندسـي البرمجيات ومديري المشاريع والمطورين والمختبرين ومسؤولي نظام التشغيل.

3.6 عمليات اتخاذ القرار:

تحديـد كيفية اتخاذ القرارات المتعلقة بدورة حياة البرمجيات، ومنها عملية الموافقة على تغييرات البرمجيات وتحديد أولويات المشـاريع وتخصيص الموارد.

4.6 إدارة المخاطر:

تحديد وتقييم المخاطر المحتملة المرتبطة بتطوير ونشر البرمجيات، وتنفيذ اسـتراتيجيات للحد من تلـك المخاطر بما يشـمل معالجة ثغـرات الأمان والموثوقيـة، والحفـاظ علـى خصوصيـة البيانـات وضمـان الامتثـال للمتطلبات التنظيمية.

5.6 ضمان جودة البرمجيات:

تنفيـذ عمليـات وأدوات لرصد وتقييم جـودة البرمجيات علـى مدار دورة حياتها ويشـمل ذلك الاختبـار المسـتمر ومراجعة كود المصـدر والالتزام بمعايير وأفضل الممارسات في البرمجة وذلك بهدف ضمان الجودة.

6.6 إدارة التغيير:

وضع إجـــراءات لإدارة تغييـــرات البرمجيـــات لضمان نجاحهـــا، ومنها جمع متطلبات التغييـــر وتحليل أثر التغيير والاختبار والنشـــر وذلك لتضمن إدارة التغيير أن التغييرات في البرمجيات تتم بطريقة مخططة ومجربة وفعالة.

7.6 مراقبة الأداء وإعداد التقارير:

مراقبة أداء البرمجيات بانتظام وقياسها مقابل المقاييس المحددة مسبقا ليساعد على تحديد مجالات التحسين ويدعم عمليات اتخاذ القرار.


CST
هيئة الاتصالات والفضاء والتقنية
Communications, Space &
Technology Commission