العدد الثامن والثلاثون
تاريخ الإصدار: 2 – كانون الأول – 2021 م
www.ajsp.net
ISSN: 2663-5798

"تحسين تقنيات اختبارات نظم معلومات التكنولوجية"

إعداد الباحث:

معتصم ماجد محمد حامد

مدخل بيانات - بلدية السلط الكبرى

العدد الثامن والثلاثون
تاريخ الإصدار: 2 – كانون الأول – 2021 م
www.ajsp.net
ISSN: 2663-5798

المقدمة:

يعتمد مجتمعنا واقتصادنا عالي التقنية أكثر فأكثر على القياسات المتطورة والمعايير الفنية وأنشطة الاختبار المرتبطة بها. كان هذا صحيحًا بالنسبة للمجتمع الصناعي في القرن العشرين ولا يزال صحيحًا بالنسبة لمجتمع المعلومات في القرن الحادي والعشرين. تم قبول مساهمات NBS / NIST في تطوير قياسات الخصائص الفيزيائية والكيميائية على نطاق واسع منذ الأيام الأولى لتأسيس المؤسسة. يعتمد عمل قياس NBS / NIST على مبادئ وتقنيات علمية معترف بها ، وعلى مر السنين ، دعمت الاحتياجات المتطورة للصناعات مثل الفولاذ والأدوات الآلية والسيارات والمعالجة الكيميائية للحصول على قياسات دقيقة ودقيقة.

خلال النصف الأخير من القرن العشرين ، أصبحت تقنية المعلومات (IT) عاملاً قوياً للتغيير في كل قطاع من قطاعات الاقتصاد تقريبًا. اليوم ، تواصل تكنولوجيا المعلومات لعب دور مهم في تحسين المنتجات والخدمات ، وجعل الناس أكثر إنتاجية ، ودعم الاقتصاد القوي. قدم التعقيد والطبيعة المتغيرة بسرعة لتكنولوجيا المعلومات تحديات تقنية فريدة للمعهد الوطني للمعايير والتكنولوجيا ولمجتمع القياس العلمي في تطوير البنية التحتية للقياس والاختبار السليم. لا تزال هذه البنية التحتية للقياس والاختبار للخصائص غير الفيزيائية وغير الكيميائية المهمة المرتبطة بأنظمة تكنولوجيا المعلومات المعقدة في مرحلة مبكرة من التطوير.

يمكن وصف أنظمة تكنولوجيا المعلومات ، التي عادة ما تكون مزيجًا من الأجهزة والبرامج ، بأنها أصبحت أكثر رقمية. يمكن وصف الأجهزة بأنها أصبحت أكثر تعقيدًا وصعوبة في التصنيع. يمكن وصف البرنامج بأنه أصبح أكثر تعقيدًا وصعوبة في التطوير مع سهولة النسخ المتماثل. أنظمة تكنولوجيا المعلومات ، بما في ذلك أجهزة الكمبيوتر وشبكات الكمبيوتر والهواتف وشبكات الهاتف وأجهزة التلفزيون وشبكات الكابلات ، موجودة في كل مكان ، وتؤثر على جميع الصناعات (التصنيع والرعاية الصحية والتعليم ، وما إلى ذلك) ، وجميع جوانب الاقتصاد. يتمثل التحدي المستمر في تطوير أساليب قياس واختبار تكنولوجيا المعلومات التي تضمن أن أنظمة تكنولوجيا المعلومات المعقدة هذه قابلة للاستخدام وقابلة للتطوير وآمنة وقابلة للتشغيل البيني. باستثناء قياس الوقت ، لا يعتمد قياس واختبار سمات أنظمة تكنولوجيا المعلومات الرقمية على القياس باستخدام النظام الدولي للوحدات (SI). بدلاً من ذلك ، يتضمن قياس تكنولوجيا المعلومات التأكد من صحة المعلومات أو البيانات عند تخزينها ومعالجتها ونقلها وعرضها وإدارتها وتنظيمها واستردادها ، إلخ.

من هذه التعريفات ، من الواضح أن الأجهزة لها سمات معالجة البيانات المادية وغيرها. في المقابل ، يتكون البرنامج من سمات فكرية أو منطقية ، وباستثناء الوقت ، فإن السمات المادية ليست ضرورية.

من المثير للاهتمام ملاحظة أن المصطلحات والقياس والاختبار لم يتم تعريفها في نفس وثيقة المفردات. تحدد المفردات الدولية للمصطلحات الأساسية والعامة في علم القياس (VIM) القياس ولكن ليس الاختبار أو الاختبار. دليل ISO / IEC 2: 1996 ، التقييس والأنشطة ذات الصلة - تحدد المفردات العامة الاختبار والاختبار ولكن ليس القياس.

يشير اختبار البرنامج إلى عملية تقييم البرنامج بهدف اكتشاف الخطأ فيه. اختبار البرامج هو أسلوب يهدف إلى تقييم سمة أو قدرة لبرنامج أو منتج وتحديد مدى توافقه مع جودته. يستخدم اختبار البرامج أيضًا لاختبار البرنامج لعوامل جودة البرامج الأخرى مثل الموثوقية ، وقابلية الاستخدام ، والسلامة ، والأمان ، والقدرة ، والكفاءة ، وقابلية النقل ، وقابلية الصيانة ، والتوافق ، وما إلى ذلك.

لسنوات عديدة حتى الآن ما زلنا نستخدم نفس تقنيات الاختبار ، وبعضها مصنوع بطريقة متقنة بدلاً من أساليب هندسية جيدة. قد يكون الاختبار مكلفًا ولكن عدم اختبار البرامج قد يكون أكثر تكلفة. يهدف اختبار البرمجيات إلى تحقيق أهداف ومبادئ معينة يجب اتباعها.

الحاجة لاختبار البرمجيات

يتضمن تطوير البرامج تطوير البرامج وفقًا لمجموعة من المتطلبات. يلزم اختبار البرامج للتحقق والتحقق من أن البرنامج الذي تم إنشاؤه لتلبية هذه المواصفات. إذا لم يكن الأمر كذلك ، فقد نفقد عميلنا على الأرجح. لذلك من أجل التأكد من أننا نقدم لعملائنا حلاً برمجيًا مناسبًا ، نذهب للاختبار. يضمن الاختبار أن ما تحصل عليه في النهاية هو ما تريد بناءه. نتحقق مما إذا كانت هناك أي مشكلة أو أي خطأ في النظام ، مما قد يجعل البرنامج غير قابل للاستخدام من قبل العميل. هذا يساعد في منع الأخطاء في النظام.

العدد الثامن والثلاثون
تاريخ الإصدار: 2 – كانون الأول – 2021 م
www.ajsp.net
ISSN: 2663-5798

أهداف اختبار البرمجيات

الأهداف هي نتاج عملية البرمجيات. اختبار البرمجيات له الأهداف التالية:

- **التحقق والمصادقة**

يمكن أيضًا استخدام الاختبار للتحقق من أن المنتج أو البرنامج يعملان بالشكل المطلوب والتحقق مما إذا كان البرنامج يفي بالشرط المنصوص عليه.

- **تغطية الأولوية**

يجب إجراء الاختبار بطريقة كفؤة وفعالة ضمن حدود الميزانية والجدول الزمني.

- **متوازن**

يجب أن توازن عملية الاختبار بين المتطلبات والقيود التقنية وتوقعات المستخدم.

- **تعقبها**

يجب إعداد المستندات الخاصة بكل من نجاح وفشل عملية الاختبار. لذلك لا حاجة لاختبار نفس الشيء مرة أخرى.

- **حتمية**

يجب أن نعرف ما نقوم به ، وما الذي نستهدفه ، وماذا ستكون النتيجة المحتملة.

مبادئ الاختبار

المبدأ هو القاعدة أو الطريقة في العمل التي يجب اتباعها. مبادئ الاختبار المختلفة هي كما يلي:

- **اختبر برنامجًا لمحاولة جعله يفشل**

الاختبار هو عملية تنفيذ برنامج بقصد اكتشاف الأخطاء. يجب أن نكشف حالات الفشل لجعل عملية الاختبار أكثر فعالية.

- **ابدأ الاختبار مبكرًا**

هذا يساعد في إصلاح الأخطاء الجسيمة في المراحل الأولى من التطوير ، ويقلل من إعادة العمل للعثور على الأخطاء في المراحل الأولية.

- **الاختبار يعتمد على السياق**

يجب أن يكون الاختبار مناسبًا ومختلفًا حسب نقاط زمنية مختلفة.

- **تحديد خطة الاختبار**

تصف خطة الاختبار عادةً نطاق الاختبار، وأهداف الاختبار، واستراتيجية الاختبار، وبيئة الاختبار، وتسليمات الاختبار، والمخاطر والتخفيف، والجدول الزمني، ومستويات الاختبار التي سيتم تطبيقها، والأساليب، والتقنيات، والأدوات المستخدمة. يجب أن تلبي خطط الاختبار بكفاءة احتياجات المؤسسة والعملاء أيضًا.

- **حالات الاختبار الفعال للتصميم**

يجب تحديد حالة الاختبار بطريقة يمكن قياسها بحيث تكون نتائج الاختبار واضحة.

- **اختبار الشروط الصالحة وغير الصالحة**

بالإضافة إلى المدخلات الصالحة، يجب علينا أيضًا اختبار النظام للمدخلات / الشروط غير الصالحة وغير المتوقعة.

- **يجب أن يتم الاختبار من قبل أشخاص مختلفين على مستويات مختلفة**

يتم تناول أغراض مختلفة على مستويات مختلفة من الاختبار لذلك يجب على الأشخاص المختلفين إجراء الاختبار بشكل مختلف باستخدام تقنيات اختبار مختلفة على مستويات مختلفة.

- **نهاية الاختبار**

يجب إيقاف الاختبار في مكان ما. يمكن إيقاف الاختبار عندما تكون المخاطر أقل من بعض الحدود أو إذا كان هناك قيود.

- **تقنيات اختبار البرامج**

في هذا القسم ، ينصب التركيز بشكل أساسي على تقنيات اختبار البرامج المختلفة.

يمكن تقسيم تقنيات اختبار البرمجيات إلى نوعين:

أولاً: الاختبار اليدوي

إنها عملية بطيئة وشاقة حيث يتم إجراء الاختبار بشكل ثابت. يتم ذلك في المرحلة المبكرة من دورة الحياة. ويسمى أيضًا الاختبار الثابت. يتم إجراؤه بواسطة المحلل والمطور وفريق الاختبار.

تقنيات الاختبار اليدوي المختلفة هي كما يلي: -

1. تجول
2. مراجعة غير رسمية
3. مراجعه فنية
4. تفتيش

ثانياً: الاختبار الآلي

يقوم هذا المختبر بتشغيل البرنامج النصي على أداة الاختبار ويتم الاختبار. يسمى الاختبار الآلي أيضًا الاختبار الديناميكي.

يتم تصنيف الاختبار الآلي إلى أربعة أنواع:

1. اختبار الصواب
2. اختبار أداء
3. اختبار الموثوقية
4. اختبار الأمان

اختبار الصواب

الدقة هي الحد الأدنى من متطلبات البرنامج. سيحتاج اختبار الصواب إلى نوع من الوحي ، لإخبار السلوك الصحيح عن السلوك الخاطئ. قد يعرف المختبر أو لا يعرف التفاصيل الداخلية لوحدة البرنامج قيد الاختبار. لذلك يمكن استخدام إما اختبار الصندوق الأبيض أو اختبار الصندوق الأسود.

AJSP
المجلة العربية للنشر العلمي
العدد الثامن والثلاثون
تاريخ الإصدار: 2 – كانون الأول – 2021 م
www.ajsp.net
ISSN: 2663-5798

يتكون اختبار الصحة من ثلاثة أشكال: -

- اختبار الصندوق الأبيض
- اختبار الصندوق الأسود
- اختبار الصندوق الرمادي

أولاً: اختبار الصندوق الأبيض

يعتبر اختبار الصندوق الأبيض فعالًا للغاية في اكتشاف المشكلات وحلها لأنه يمكن غالبًا العثور على الأخطاء قبل أن تسبب المشاكل. اختبار الصندوق الأبيض هو عملية إعطاء المدخلات للنظام والتحقق من كيفية معالجة النظام لتلك المدخلات لتوليد المخرجات المطلوبة. يُطلق على اختبار المربع الأبيض أيضًا تحليل المربع الأبيض أو اختبار المربع الشفاف أو تحليل المربع الأبيض. يمكن تطبيق اختبار الصندوق الأبيض على مستويات التكامل والوحدة والنظام لعملية اختبار البرنامج. يعتبر اختبار المربع الأبيض طريقة اختبار أمان يمكن استخدامها للتحقق مما إذا كان تنفيذ الكود يتبع التصميم المقصود، للتحقق من صحة وظائف الأمان المطبقة، والكشف عن الثغرات الأمنية القابلة للاستغلال.

بعض الأنواع المختلفة لتقنيات اختبار الصندوق الأبيض هي كما يلي: -

1. اختبار المسار الأساسي
2. اختبار الحلقة
3. اختبار هيكل التحكم

مزايا اختبار الصندوق الأبيض

- سيتم ممارسة جميع المسارات المستقلة في الوحدة مرة واحدة على الأقل.
- سيتم ممارسة جميع القرارات المنطقية.
- سيتم تنفيذ جميع الحلقات في حدودها.
- ستتم ممارسة هياكل البيانات الداخلية للحفاظ على صحتها.
- تم الكشف عن الأخطاء في الرموز المخفية.
- تقريب التقسيم الذي تم بواسطة معادلة التنفيذ.
- يقدم المطور بعناية أسباب التنفيذ.

عيوب اختبار الصندوق الأبيض: -

- فقدان عن الحالات المحذوفة في الكود.
- نظرًا لأن معرفة الكود والهيكل الداخلي شرط أساسي ، يلزم وجود مُختبِر ماهر لإجراء هذا النوع من الاختبارات ، مما يزيد من التكلفة.
- ويكاد يكون من المستحيل النظر في كل جزء من التعليمات البرمجية لاكتشاف الأخطاء المخفية، والتي قد تخلق مشاكل ، مما يؤدي إلى فشل التطبيق.

ثانياً: اختبار الصندوق الأسود

- اختبار الصندوق الأسود هو اختبار البرنامج بناءً على متطلبات الإخراج ودون أي معرفة بالهيكل الداخلي أو الترميز في البرنامج.
- في الأساس، يعد اختبار الصندوق الأسود جزءًا لا يتجزأ من "اختبار الصواب" ولكن أفكاره لا تقتصر على اختبار الصواب فقط. الهدف هو اختبار مدى توافق المكون مع المتطلبات المنشورة للمكون. لا يهتم اختبار الصندوق الأسود بالهيكل المنطقي الداخلي للنظام أو لا يهتم به، فهو يفحص فقط الجانب الأساسي للنظام. يتأكد من أن الإدخال مقبول بشكل صحيح وأن الإخراج يتم إنتاجه بشكل صحيح.

فيما يلي بعض الأنواع المختلفة لتقنيات اختبار الصندوق الأسود:

1. التقسيم المعادل
2. تحليل قيمة الحدود
3. تقنيات الرسم البياني السبب والنتيجة
4. اختبار المقارنة
5. اختبار الزغب
6. الاختبار القائم على النموذج

مزايا اختبار الصندوق الأسود: -

- يتم تقليل عدد حالات الاختبار لتحقيق اختبار معقول.
- يمكن أن تظهر حالات الاختبار وجود أو عدم وجود فئات من الأخطاء.
- لا يحتوي جهاز اختبار الصندوق الأسود على "ارتباط" بالرمز.
- المبرمج والاختبار كلاهما مستقلان عن بعضهما البعض.
- أكثر فاعلية في وحدات الكود الأكبر من اختبار الصندوق الشفاف.

عيوب اختبار الصندوق الأسود: -

- يصعب تصميم حالات الاختبار بدون مواصفات واضحة.
- يمكن اختبار عدد قليل فقط من المدخلات الممكنة.
- لم يتم اختبار بعض أجزاء النهاية الخلفية على الإطلاق.
- فرص وجود مسارات مجهولة خلال هذا الاختبار.
- فرص تكرار الاختبارات التي قام بها المبرمج بالفعل.

ثالثاً: اختبار الصندوق الرمادي

منهجية اختبار الصندوق الرمادي هي طريقة اختبار برمجية تُستخدم لاختبار تطبيقات البرامج. المنهجية مستقلة عن النظام الأساسي واللغة. يعتمد التنفيذ الحالي لمنهجية الصندوق الرمادي بشكل كبير على استخدام مصحح أخطاء النظام الأساسي المضيف لتنفيذ والتحقق من صحة البرنامج قيد الاختبار. أكدت الدراسات الحديثة أنه يمكن تطبيق طريقة الصندوق الرمادي في الوقت الفعلي باستخدام برنامج يتم تنفيذه على النظام الأساسي المستهدف. جمعت تقنيات اختبار الصندوق الرمادي بين منهجية اختبار الصندوق الأبيض والصندوق الأسود. تُستخدم تقنية اختبار الصندوق الرمادي لاختبار برنامج مقابل مواصفاته ولكن باستخدام بعض المعرفة بعمله الداخلي أيضًا. إن فهم الأجزاء الداخلية للبرنامج في اختبار الصندوق الرمادي هو أكثر من اختبار الصندوق الأسود ولكنه أقل من اختبار الصندوق الشفاف.

منهجية الصندوق الرمادي هي عملية من عشر خطوات لاختبار برامج الكمبيوتر.

منهجية عشر خطوات للمربع الرمادي

- تحديد المدخلات
- تحديد النواتج
- تحديد المسارات الرئيسية
- تحديد الوظيفة الفرعية X (SF)
- تطوير المدخلات لـ SF X
- تطوير مخرجات SF X
- نفذ حالة الاختبار لـ SF X
- تحقق من النتيجة الصحيحة لـ SF X
- كرر الخطوات 4: 8 مع سادس أخرى
- كرر الخطوتين 7 و 8 للانحدار

العدد الثامن والثلاثون
تاريخ الإصدار: 2 – كانون الأول – 2021 م
www.ajsp.net
ISSN: 2663-5798

تستخدم منهجية الصندوق الرمادي أدوات اختبار البرامج المؤتمتة لتسهيل إنشاء برنامج اختبار فريد. يتم إنشاء برامج تشغيل الوحدات النمطية والأوتاد بواسطة مجموعة الأدوات لإعفاء مهندس اختبار البرنامج من الاضطرار إلى إنشاء هذا الرمز يدويًا. تتحقق مجموعة الأدوات أيضًا من تغطية الكود من خلال ضبط كود الاختبار. "تساعد أدوات الأجهزة في إدخال رمز الأجهزة دون تكبد الأخطاء التي قد تحدث من الأجهزة اليدوية". من خلال التشغيل في مصحح أخطاء أو محاكي مستهدف ، تتحكم مجموعة أدوات Gray box في تشغيل برنامج الاختبار. لقد انتقلت منهجية المربع الرمادي من مصحح الأخطاء إلى العالم الحقيقي إلى الوقت الفعلي. يمكن تطبيق المنهجية في الوقت الفعلي عن طريق تعديل الفرضية الأساسية التي مفادها أنه يمكن إرسال المدخلات إلى برنامج الاختبار عبر رسائل النظام العادية ، ثم يتم التحقق من المخرجات باستخدام رسائل إخراج النظام.

اختبار أداء

يتضمن اختبار الأداء جميع المراحل مثل دورة حياة الاختبار السائدة كنظام مستقل يتضمن إستراتيجية مثل التخطيط والتصميم والتنفيذ والتحليل وإعداد التقارير.

ليست كل البرامج لها مواصفات خاصة بالأداء بشكل صريح. لكن سيكون لكل نظام متطلبات أداء ضمنية.

لطالما كان الأداء مصدر قلق كبير وقوة دافعة لتطور الكمبيوتر. يمكن أن تكون أهداف اختبار الأداء هي تحديد عنق الزجاجة ومقارنة الأداء والتقييم.

من خلال اختبار الأداء ، يمكننا قياس خصائص أداء أي تطبيق. أحد أهم أهداف اختبار الأداء هو الحفاظ على زمن انتقال منخفض لموقع ويب وإنتاجية عالية واستخدام منخفض.

اختبار الأداء له شكلين: -

أولاً: اختبار الحمل

اختبار الحمل هو عملية إخضاع جهاز كمبيوتر أو جهاز طرفي أو خادم أو شبكة أو تطبيق لمستوى عمل يقترب من حدود مواصفاته. يمكن إجراء اختبار الحمل في ظل ظروف معملية خاضعة للرقابة لمقارنة إمكانيات الأنظمة المختلفة أو لقياس قدرات نظام واحد بدقة. في هذا ، يمكننا التحقق مما إذا كان البرنامج يمكنه التعامل مع عبء العديد من المستخدمين أم لا.

ثانياً: اختبار الإجهاد

اختبار الإجهاد هو اختبار يتم إجراؤه لتقييم نظام أو مكون عند أو خارج حدود متطلباته المحددة لتحديد الحمل الذي يفشل تحته وكيف.

اختبار الموثوقية

الغرض من اختبار الموثوقية هو اكتشاف المشكلات المحتملة في التصميم في أقرب وقت ممكن، وفي النهاية ، توفير الثقة في أن النظام يلبي متطلبات الموثوقية الخاصة به. يرتبط اختبار الموثوقية بالعديد من جوانب البرنامج الذي يتم تضمين عملية الاختبار فيه ؛ عملية الاختبار هذه هي طريقة أخذ عينات فعالة لقياس موثوقية البرامج. في النظام بعد تطوير البرنامج ، يمكن إجراء تقنيات اختبار الموثوقية مثل تقنيات التحليل أو الإصلاح للتحقق من استخدام البرنامج.

اختبار الأمان

ترتبط جودة البرامج والموثوقية والأمان ارتباطًا وثيقًا. يمكن للمتطفلين استغلال الثغرات الموجودة في البرامج لفتح ثغرات أمنية.

يتأكد اختبار الأمان من أن الأشخاص المصرح لهم فقط هم من يمكنهم الوصول إلى البرنامج ويمكن للموظفين المعتمدين فقط الوصول إلى الوظائف المتاحة لمستوى الأمان الخاص بهم. يتم إجراء اختبار الأمان للتحقق مما إذا كان هناك أي تسرب للمعلومات بالمعنى المقصود من خلال تشفير التطبيق أو استخدام مجموعة واسعة من البرامج والأجهزة وجدار الحماية وما إلى ذلك.

العدد الثامن والثلاثون
تاريخ الإصدار: 2 – كانون الأول – 2021 م
www.ajsp.net
ISSN: 2663-5798

استراتيجيات اختبار البرامج

تدمج إستراتيجية اختبار البرامج طرق تصميم حالة اختبار البرامج في سلسلة من الخطوات جيدة التخطيط والتي تؤدي إلى بناء ناجح للبرنامج. تعطي استراتيجيات اختبار البرامج خارطة طريق للاختبار. يجب أن تكون استراتيجية اختبار البرامج مرنة بما يكفي للترويج لنهج اختبار مخصص وفي نفس الوقت يجب أن تكون صحيحة بدرجة كافية. يتم تطوير الإستراتيجية بشكل عام من قبل مديري المشاريع ومهندسي البرمجيات والمتخصصين في الاختبار.

هناك أربع استراتيجيات مختلفة لاختبار البرامج:

1. وحدة التجارب
2. اختبار التكامل
3. اختبار القبول / التحقق من الصحة
4. اختبار النظام هناك أربع استراتيجيات مختلفة لاختبار البرامج

أولاً: وحدة التجارب

الوحدة هي أصغر وحدة نمطية ، أي أصغر مجموعة من سطور التعليمات البرمجية التي يمكن اختبارها. يعد اختبار الوحدة أحد مستويات الاختبار التي تتكامل معًا لتكوين الصورة الكبيرة لاختبار النظام. يكمل تكنولوجيا المعلومات اختبار التكامل والاختبار على مستوى النظام. يجب أن تكمل أيضًا مراجعات التعليمات البرمجية والإرشادات التفصيلية.

يُنظر إلى اختبار الوحدة عمومًا على أنه فئة اختبار المربع الأبيض. وهذا يعني أنه من المتحيز النظر إلى الكود وتقييمه كما تم تنفيذه. بدلاً من تقييم التوافق مع مجموعة من المتطلبات.

فوائد اختبار الوحدة: -

1. يعتبر الاختبار على مستوى الوحدة فعالاً للغاية من حيث التكلفة.
2. يوفر تحسين موثوقية أكبر بكثير للموارد الموسعة من الاختبار على مستوى النظام. على وجه الخصوص ، تميل إلى الكشف عن الأخطاء الخبيثة وغالبًا ما تكون كارثية مثل تعطل النظام الغريب الذي يحدث في الميدان عندما يحدث شيء غير عادي.
3. كن قادرًا على اختبار أجزاء من المشروع دون انتظار توفر الأجزاء الأخرى.
4. تحقيق التوازي في الاختبار من خلال القدرة على اختبار المشكلات وإصلاحها في وقت واحد من قبل العديد من المهندسين.
5. أن تكون قادرًا على اكتشاف العيوب وإزالتها بتكلفة أقل بكثير مقارنة بالمراحل اللاحقة الأخرى من الاختبار.
6. أن تكون قادرًا على الاستفادة من عدد من تقنيات الاختبار الرسمية المتاحة لاختبار الوحدة.
7. تبسيط تصحيح الأخطاء من خلال قصر مناطق الكود الممكنة على وحدة صغيرة للبحث عن الأخطاء.
8. أن تكون قادرًا على اختبار الظروف الداخلية التي لا يمكن الوصول إليها بسهولة عن طريق المدخلات الخارجية في الأنظمة المتكاملة الأكبر.
9. أن تكون قادرًا على تحقيق مستوى عالٍ من التغطية الهيكلية للكود.
10. تجنب دورات تصحيح أخطاء إنشاء التجميع المطولة عند تصحيح المشكلات الصعبة.

تقنيات اختبار الوحدة

يمكن استخدام عدد من تقنيات الاختبار الفعالة في مرحلة اختبار الوحدة. يمكن تقسيم تقنيات الاختبار على نطاق واسع إلى ثلاثة أنواع:

- الاختبار الوظيفي
- الاختبار الإنشائي
- اختبار إرشادي أو بديهي

العدد الثامن والثلاثون
تاريخ الإصدار: 2 – كانون الأول – 2021 م
www.ajsp.net
ISSN: 2663-5798

ثانياً: اختبار التكامل

اختبار التكامل هو أسلوب منهجي لبناء هيكل البرنامج مع إجراء اختبارات في نفس الوقت للكشف عن الأخطاء المرتبطة بالتفاعل. الهدف هو أخذ المكونات المختبرة بالوحدة وبناء هيكل البرنامج الذي تمليه التصميم.

تتم مناقشة استراتيجيات اختبار التكامل المختلفة أدناه: -

1. اختبار التكامل من أعلى إلى أسفل
2. اختبار التكامل السفلي

- **التكامل من أعلى إلى أسفل**

اختبار التكامل من أعلى إلى أسفل هو نهج تدريجي لبناء هيكل البرنامج. يتم دمج الوحدات عن طريق التحرك لأسفل خلال الهيكل ، بدءًا من وحدة التحكم الرئيسية. يتم دمج الوحدات التابعة لوحدة التحكم الرئيسية في الهيكل إما بطريقة العمق أولاً أو العرض أولاً.

يتم تنفيذ عملية الدمج في سلسلة من خمس خطوات:

1. يتم استخدام وحدة التحكم الرئيسية كسائق اختبار ويتم استبدال الأذرع لجميع المكونات التابعة مباشرة لوحدة التحكم الرئيسية.
2. اعتمادًا على نهج التكامل ، يتم استبدال الأجزاء الجذرية الثانوية المحددة واحدة تلو الأخرى بمكونات فعلية.
3. يتم إجراء الاختبارات عند دمج كل مكون.
4. عند الانتهاء من كل مجموعة من الاختبارات ، يتم استبدال كعب آخر بالمكون الحقيقي.
5. يمكن إجراء اختبار الانحدار للتأكد من عدم إدخال أخطاء جديدة.

إنها ليست بسيطة نسبيًا كما تبدو. في هذه المشكلة اللوجستية يمكن أن تنشأ. تظهر المشكلة عند اختبار الوحدات ذات المستوى المنخفض والتي تتطلب اختبار المستوى الأعلى. استبدل Stub وحدة المستوى المنخفض في بداية الاختبار من أعلى إلى أسفل. لذلك لا يمكن أن تتدفق أي بيانات في اتجاه تصاعدي.

- **التكامل من الأسفل إلى الأعلى**

يبدأ اختبار التكامل التصاعدي ، كما يوحي اسمه ، في البناء والاختبار باستخدام الوحدات الذرية. نظرًا لأن المكونات تتكامل من الأسفل إلى الأعلى ، فإن المعالجة المطلوبة للمكونات التابعة لمستوى معين متاحة دائمًا ويتم التخلص من الحاجة إلى بذرة. يمكن تنفيذ استراتيجية تكامل من أسفل إلى أعلى بالخطوات التالية:

- يتم دمج المكونات منخفضة المستوى في مجموعات تؤدي وظيفة برمجية فرعية معينة.
- تتم كتابة برنامج تشغيل لتنسيق إدخال وإخراج حالة الاختبار.
- يتم اختبار الكتلة.
- تتم إزالة برامج التشغيل ويتم دمج المجموعات التي تتحرك صعودًا في هيكل البرنامج

اختبار القبول

اختبار القبول (المعروف أيضًا باسم اختبار قبول المستخدم) هو نوع من الاختبار يتم إجراؤه للتحقق مما إذا كان المنتج قد تم تطويره وفقًا للمعايير والمعايير المحددة ويلبي جميع المتطلبات المحددة من قبل العميل. يتم إجراء هذا النوع من الاختبار بشكل عام بواسطة المستخدم / العميل حيث يتم تطوير المنتج خارجيًا بواسطة طرف آخر. يندرج اختبار القبول في إطار منهجية اختبار الصندوق الأسود حيث لا يهتم المستخدم كثيرًا بالعمل الداخلي / الترميز الداخلي للنظام ، ولكنه يقيم الأداء العام للنظام ويقارنه بالمتطلبات المحددة من قبله. يعتبر اختبار قبول المستخدم أحد أهم الاختبارات التي يقوم بها المستخدمون قبل تسليم النظام أو تسليمه إلى المستخدم النهائي.

يُعرف اختبار القبول أيضًا باسم اختبار التحقق من الصحة ، والاختبار النهائي ، واختبار ضمان الجودة ، واختبار قبول المصنع واختبار التطبيق ، وما إلى ذلك. وفي هندسة البرمجيات ، يمكن إجراء اختبار القبول على مستويين مختلفين ؛ واحد على مستوى مزود النظام والآخر على مستوى المستخدم النهائي.

أنواع اختبار القبول

- **اختبار قبول المستخدم**

يعتبر اختبار قبول المستخدم في هندسة البرمجيات خطوة أساسية قبل قبول المستخدم النهائي للنظام. بشكل عام ، اختبار قبول المستخدم هو عملية اختبار النظام قبل قبوله نهائيًا من قبل المستخدم.

- **اختبار ألفا واختبار بيتا**

اختبار ألفا هو نوع من اختبار القبول يتم إجراؤه في موقع المطور من قبل المستخدمين. في هذا النوع من الاختبار ، يواصل المستخدم اختبار النظام ، ويلاحظ المطور النتيجة ويلاحظها في وقت واحد.

الاختبار التجريبي هو نوع من الاختبار يتم إجراؤه في موقع المستخدم. يقدم المستخدمون ملاحظاتهم للمطور لنتائج الاختبار. يُعرف هذا النوع من الاختبارات أيضًا باسم الاختبار الميداني. تُستخدم التعليقات الواردة من المستخدمين لتحسين النظام / المنتج قبل إصداره للمستخدمين / العملاء الآخرين.

- **اختبار القبول التشغيلي**

يُعرف هذا النوع من الاختبارات أيضًا باسم اختبار الاستعداد / الاستعداد التشغيلي. إنها عملية ضمان أن جميع المكونات المطلوبة (العمليات والإجراءات) للنظام موجودة من أجل السماح للمستخدم / المختبِر باستخدامها.

- **اختبار قبول الاتصال واللوائح**

في اختبار قبول العقد واللوائح ، يتم اختبار النظام وفقًا للمعايير المحددة كما هو مذكور في مستند العقد وأيضًا اختباره للتحقق مما إذا كان يفي / يلتزم بجميع لوائح وقوانين الحكومة والسلطات المحلية وكذلك جميع المعايير الأساسية.

الخاتمة:

يصف هذا البحث الخاصة باختبار البرامج بالتفصيل اختبار البرامج ، والحاجة إلى اختبار البرامج ، وأهداف اختبار البرامج ، والمبادئ. غالبًا ما يكون اختبار البرامج أقل رسمية وصرامة مما ينبغي ، والسبب الرئيسي لذلك هو أننا كافحنا لتحديد أفضل الممارسات والمنهجيات والمبادئ والمعايير لاختبار البرنامج الأمثل. لإجراء الاختبار بفعالية وكفاءة ، يجب أن يكون كل شخص مشارك في الاختبار على دراية بأهداف اختبار البرامج الأساسية والمبادئ والقيود والمفاهيم.

نوضح كذلك تقنيات اختبار البرامج المختلفة مثل اختبار الصحة واختبار الأداء واختبار الموثوقية واختبار الأمان. علاوة على ذلك ، ناقشنا المبادئ الأساسية لاختبار الصندوق الأسود واختبار الصندوق الأبيض واختبار الصندوق الرمادي. لقد قمنا بمسح بعض الاستراتيجيات التي تدعم هذه النماذج ، وناقشنا مزاياها وعيوبها. نصف أيضًا استراتيجيات اختبار البرامج المختلفة مثل اختبار الوحدة واختبار التكامل واختبار القبول واختبار النظام.

المصادر والمراجع:

Sawant, A. A., Bari, P. H., & Chawan, P. M. (2012). Software testing techniques and strategies. International Journal of Engineering Research and Applications (IJERA), 2(3), 980-986.

Jan, S. R., Shah, S. T. U., Johar, Z. U., Shah, Y., & Khan, F. (2016). An innovative approach to investigate various software testing techniques and strategies. International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Print ISSN, 2395-1990.

Anwar, N., & Kar, S. (2019). Review paper on various software testing techniques & strategies. Global Journal of Computer Science and Technology.

Basili, V. R., & Selby, R. W. (1987). Comparing the effectiveness of software testing strategies. IEEE transactions on software engineering, (12), 1278-1296.

Briand, L., & Labiche, Y. (2004). Empirical studies of software testing techniques: Challenges, practical strategies, and future research. ACM SIGSOFT Software Engineering Notes, 29(5), 1-3.

Subhiyakto, E. R., & Utomo, D. W. (2016). Software Testing Techniques and Strategies Use in Novice Software Teams. SISFO Vol 5 No 5, 5.

Fraser, G., & Arcuri, A. (2012, April). The seed is strong: Seeding strategies in search-based software testing. In 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation (pp. 121-130). IEEE.

Batra, S. (2011). Improving quality using testing strategies. Journal of Global Research in Computer Science, 2(6), 113-117.