جامعة طرابلس ـ كلية تقنية المعلومات

مقدمة في هندسة البرمجيات

Introduction to software Engineering

ITGS-213

المحاضرة الثامنة – تأكيد الجودة واختبار البرمجيات

Software Quality Assurance & Testing

خريف2020

مواضيع المحاضرة

- مقدمة
- مفاهيم أساسية في الجودة وأهميتها
- تأكيد الجودة البرمجيات
- المراجعة والفحص والتصفح
- خصائص الجودة
- معايير البرمجيات
- أنواع الاخطاء البرمجية
- الاختبارTesting

مقدمـــــــــــة

- يعتبر تأكيد أو ضمان الجودة **QA Quality Assurance** من النشاطات المهمة في اي مشروع تجاري .ظهرت هندسة البرمجيات كحل لإعداد منتوج برمجي فائق الجودة ومطابق للمعايير المتفق عليه من قبل مؤسسات معايير ومواصفات الجودة (ACM, IEEE, ISO).

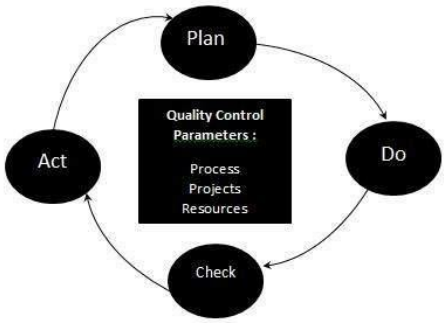
المعيار	الوصف
IEEE 1061	منهجية لتحديد متطلبات الجودة، والتحقق من صحة هذه العملية.
IEEE 1008	هناك معيار لاختبار وحدة.
ISO / IEC 12119	مجموعة من المتطلبات لحزم البرمجيات. تعليمات لاختبار مجموعة من البرامج المقدمة ضد المتطلبات المحددة.

مقدمـــــــــــة

- تأكيد الجودة من مهمة المبرمج ومهندس البرمجيات.
- كان تأكيد جودة البرمجيات في السابق يركز على انتاج منظومات دون المرور على اختبار كامل يؤكد خلوها من الأخطاء والذي سبب في أزمة البرمجيات.
- ولهذا بدأ الخبراء في التركيز على عملية الاختبار والاهتمام بالجودة فظهر مصطلح تأكيد الجودة البرمجيات **Software Quality Assurance.**
- وبالتالي فان التحقق Verificationو التصديق Validationيشيران الى تأكيد الجودة البرمجيات.**SQA**
  - تركز عمليات التحقق على مواصفات التصميم والنظام.Verifications
  - يؤكد التحقق من الصحة على متطلبات المستخدم. Validation

مفاهيم أساسية في الجودة وأهميتها

- **مراقبة الجودة:Quality Control QC**



- **وهي العناية بالمنتوج منذ تصنيعه وقبل وصوله للمستخدم حيث تجرى عليه مجموعة اختبارات الية ويدوية ومختلفة.**
- تتم هذه الاختبارات عبر حلقة تسمى (Feedback loop) حيث يتم مقارنة المنتوج مع المعايير المحددة سلفاً.
- مراقبة الجودة تعطى الإدارة العليا المعلومات الضرورية(في شكل تقرير) للحصول على الثقة في المنتوج من حيث مطابقته للأهداف والمواصفات والمعايير المطلوبة.
- **والهدف منها هو تقليل التكلفة وجعل الزبون في وضع افضل من الامان والرضا.**

مفاهيم أساسية في الجودة وأهميتها

- **تأكيد أو ضمان جودة البرمجيات:Software Quality Assurance SQA**

- هي عملية تأكيد توافق المنتوج البرمجي مع المتطلبات الوظيفية والغير وظيفية وتتطابق الادوات والمنهجيات والطرق والاجراءات مع المعايير المعتمدة في اعداد البرمجيات.
- أو هي مجموع السمات والخصائص المتعلقة بالمنتج البرمجي التي تجعله قادراً على تحقيق الهدف من إنتاجه.
- يفضل ان يقوم فريق مختلف عن فريق اعداد البرمجيات بعملية تأكيد الجودة لكي تتم العملية من وجهة نظر أكثر شمولية واستقلالية.
- يفضل ان يكون الفريق يمثل الزبون لكي يتأكد من أن المنظومة تطابق معايير الجودة المنصوص عليها في وثيقة المتطلبات.

6

تأكيد الجودة البرمجيات
Software Quality Assurance SQA

- **تهدف عملية تأكيد جودة البرمجيات إلى:**

1. فحص مكونات وشفرة البرمجية للتأكد من خلوها من الأخطاء ومطابقتها للمواصفات والمعايير الموضوعة من قبل مستخدمي البرمجية وذلك قبل إصدار تلك البرمجية إلى المستخدمين المعنيين بها.
2. تهدف إلى التأكد من أن البرمجيات الموضوعة للإصدار تطابق احتياجات ومتطلبات سوق العمل.
3. تقوم عملية تأكيد جودة البرمجية على اتباع طرق وخطوات وتقنيات منهجية ومدروسة النتائج أثناء عملية الفحص.

تأكيد الجودة البرمجيات
Software Quality Assurance

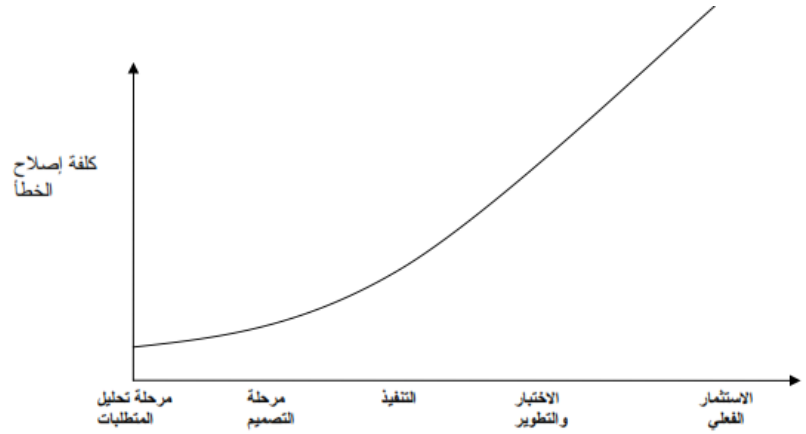
- **من مبدأ الوقاية خير من العلاج**

- يجب أن نطبق ضمان الجودة على أي منتج برمجي. علما بان عملية ضمان الجودة تضاف الى تكلفة المنتج ، الا انها تقلل من تكاليف الاصلاح بعد عملية بيع المنتج.
- مثال: نفرض ان لدينا كود برمجي يبلغ 100000 سطر، وتم انفاق 500 ساعة في اختباره وتنقيحه وكانت تكلفة كل ساعة 30د. وكانت النتيجة انه تم تفادي 200 خطأ محتملا. سيترتب على المؤسسة تكاليف اضافية على المنتج تبلغ 30*500=15000د لتجنب 200 خطأ.
- وعلى افتراض ان عملية الاختبار لم تتم وان أي خطأ سيظهر عند التنفيذ العملي بعد البيع سيكلف 1000د. وبافتراض انه لم يظهر سوى 100 عيب برمجي . وبالتالي فان تكلفة الاصلاح تكون 100000د أي ما يعادل 6 أضعاف التكلفة السابقة.

8

تأكيد الجودة البرمجيات
Software Quality Assurance

- من مبدأ الوقاية خير من العلاج



Quality Management

	QA تأكيد الجودة	QC مراقبة الجودة	TEST الاختبار
الغرض	وضع العملية المناسبة ، وإدخال معايير الجودة لمنع الأخطاء والعيوب في المنتج	التأكد من مطابقة المنتج للمتطلبات والمواصفات قبل طرحه او تسليمه	الكشف عن أخطاء البرامج وعيوبها وحلها
التركيز	عمليات تطوير المنتج البرمجي	المنتج البرمجي ككل	الشفرة المصدرية والتصميم
ماذا	منع او وقاية	تَحَقّق	اكتشاف الاخطاء
من	الفريق يشمل اصحاب المصلحة	الفريق البرمجي	مهندس الاختبار او المطور البرمجي
متى	اثناء عمليات تطوير المنتج البرمجي	قبل طرح المنتج البرمجي في الاسواق قبل تسليمه للزبون	في مرحلة الاختبار او طوال فتره تطوير المنتج البرمجي

QA, QC and Testing in software development process

تأكيد الجودة البرمجيات
Software Quality Assurance

- **نشاطات تخص ضمان الجودة :هي مجموعات نشاطات يقوم بها فريق تأكيد الجودة:**

ا. تجهيز خطة الجودة:**Quality plan**

خطة الاختبار هي مستند ديناميكي يتم تحديثه على أساس الطلب. إنه في الأساس مخطط لكيفية تنفيذ نشاط الاختبار.

يتم تجهيز الخطة في مرحلة التخطيط ويتم مراجعتها من قبل الزبون وتشمل هذه الخطة الآتي:

- تحديد البنود المراد تقييمها.
- تحديد المراجعات المراد أداؤها.
- تحديد المعايير المراد التعاطي معها.
- مراجعة الأدوات والطرق المستخدمة لأعداد المنظومة البرمجية.

المراجعة والفحص والتصفح
Reviewing, inspection & Walkthrough

- **تحديد أهم المراجعات المراد أداؤها ضمن خطة الجودة Quality plan بالتالي:**
- التصفح.**Walkthrough**
- فحص الشفرة.**Code Inspection**
- مراجعة البرمجيات.**Software Review**

1) التصفح:Walkthrough

- هي طريقة لمراجعة المنتوج البرمجي مراجعة غير رسمية و ذلك بمشاركة المخطط الانسيابي لجزء برمجي مع فريق العمل أو المنظومة فيما بينهم ولا ضرورة من معرفة مدير المشروع ولايتم توثيقها.

12

المراجعة والفحص والتصفح
Reviewing, inspection & Walkthrough

2) فحص الشفرة Code Inspection

- هي عملية كشف عن أخطاء في جزء برمجي وليس تصحيحه مثل (الاخطاء المنطقية في الشفرة) الذي يقوم بتصحيح هذه الاخطاء المبرمج.
- تتم عملية الفحص في جلسة تعقد بحضور معد الجزء البرمجي وفريق الفحص لفحص الشفرة يتم توفير الاتي:

1. الشفرة المراد تفحصها.
2. المعايير المستخدمة في اعداد المنظومة.
3. قائمة بالأخطاء المتوقعة ويتم استخلاصها من منظومات سابقة بواسطة اخصائيين.

مثل هل تم اعطاء قيم ابتدائية للمتغيرات المستخدمة في (الشفرة)؟ -هل كل حلقة لها نهاية؟

المراجعة والفحص والتصفح
Reviewing, inspection & Walkthrough

- **الجدول التالي يعرض الفرق بين الفحص Inspection والتصفح Walkthrough.**

التصفح Walkthrough	الفحص Inspection
جلسات غير رسمية	تستخدم جلسات رسمية
جلسات غير منتظمة	اسلوب منتظم للبنود
اداة تعليمية ممتازة لأعضاء الفريق الجدد	تتم بواسطة عناصر مؤهلة
تتم بواسطة زملاء فريق إعداد المنظومة	معد الجزء البرمجي وفريق الفحص

14

المراجعة والفحص والتصفح
Reviewing, inspection & Walkthrough

3) مراجعة البرمجيات Software Review

- مراجعة البرمجيات عملية مهمة حيث يتم فيها مراجعة التوثيق في نهاية كل مرحلة (التحليل – التصميم – البرمجة)
- يتم تجهيز تقرير للإدارة العليا في نهاية كل مرحلة بعد المراجعة بحضور الإدارة ونتيجة لذلك يتم اتخاذ احد القرارات التالية:

1. الاستمرار في المرحلة التي تليها.
2. التوقف عن المشروع بالكامل.
3. اجراء التغييرات على التوثيق ثم الاستمرار.

خصائص الجودة
Quality Attributes

- **نورد بعض الخصائص التالي يجب أن تتصف بها المنظومة البرمجية الناجحة:**

- **الاعتمادية: Reliability** تتصل المصداقية أو الموثوقية بالدرجة التي يمكن للبرنامج أن يؤدي فيها وظائفه دون أخطاء , وعموماً كلما زادت مرات المراجعة وفترة الاختبار للبرنامج كلما قلت أخطاؤه.
- **قابلية الصيانة : Maintainability** تكون مصممة بطريقة تسهل عملية تحديد واصلاح الاخطاء.
- **الامن والحماية : Security** يجب أن يتمتع البرنامج بأمن البيانات , ومن ذلك التحكم في الدخول للبرنامج على شبكات محلية أو عامة , والتأكد من أن من يستخدم البرنامج مسموح له باستخدامه بالفعل.
- **التوسعية : Expandability** وهي إمكانية ترقية البرنامج وزيادة قدراته ووظائفه لرفع كفاءته وزيادة محتواه أو تحديثه.

خصائص الجودة
Quality Attributes

- **الدقة : Accuracy** الدقة في نتائج المنظومة خاصة الحسابات. وتقاس الدقة بعدد الأخطاء في الألف ويعد مستوى ثلاثة أخطاء في الألف مستوى مقبول في الدقة.
- **الاداء : Performance** سريعة الاستجابة والمعالجة.
- **سهولة الاستخدام :Usability** وتعد من أهم المعايير المؤثرة على نجاح البرنامج , فلو صعب استخدام البرنامج من قبل المستخدم النهائي فإن ذلك يجعل البرنامج عديم الفائدة حتى لو تمتع بالكفاءة والمصداقية , وهذا يتطلب واجهة رسومية جذابة للمستخدم.
- **قابلة لإعادة الاستخدام : Reusability** يمكن استخدام اجزاء المنظومة في برامج مستقبلية. ويقصد تحويل محتويات البرنامج لكي تستخدم في برنامج آخر أو الانتفاع بمحتويات البرنامج في تطبيقات أخرى تحت نفس نظام التشغيل , ويجب أن يسمح بذلك من قبل المنتج الأصلي للبرنامج.

معايير البرمجيات
Software Standards

أهداف المعايير :نضع بعض المعايير للبرمجيات للاهداف الآتية:

1. للتفاهم والوصول بين متخصص الحاسوب والزبون لعمل مثالي.
2. لتجنب تكرار أخطاء الماضي.
3. لاستمرارية العمل في المنظومة حتى في غياب بعض أعضاء فريق اعداد المنظومة.
4. لتوحيد التقنيات والأدوات المستخدمة من قبل المبرمجين في المؤسسة الواحدة.
5. للحصول على منتوج ذي جودة عالية.

أنواع المعايير
Types of standards

1) معايير خاصة بعملية اعداد المنظومة:

- هي معايير تتعلق بالطرق والادوات الفنية في اعداد البرمجيات .مثل تحديد المتطلبات يجب أن يكون كاملا قبل التنفيذ.

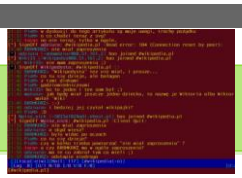
2) معايير خاصة بالمنتوج البرمجي:

- هي معايير خاصة بشكل المنتوج البرمجي وليس كيفية اعداده .**مثل**
  - معايير في تسمية الوثائق المستخدمة.
  - معايير في شكل وهيئة شفرة البرنامج.

أنواع أخطاء البرمجيات

- **هناك ثلاث أخطاء تحدث في المنظومات:**

1. أخطاء في المتطلبات (الوظائف غير كاملة)
2. أخطاء في التصميم (الاخفاق في ترجمة المتطلبات الى مواصفات التصميم)
3. أخطاء في التنفيذ (مثل خطأ في اجراء العمليات الحسابية في منظومة المرتبات).

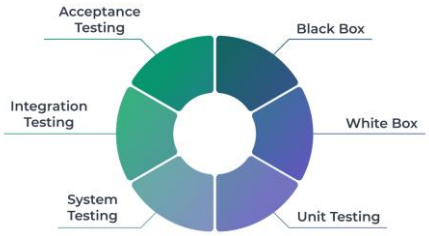


الاختبار
Testing

- الاختبار هو مجموعة من الانشطة activities التي اذا تم تخطيطها وتنفيذها بصورة صحيحة اثناء اعداد البرمجيات فأنها ستؤدي الى منظومة موثوقة Reliable خالية من الاخطاء.
- أو هو عملية تقييم نظام أو عنصرها لإيجاد ما إذا كانت تلبي متطلبات محددة أم لا .
- أو الاختبار يتم تنفيذه من أجل تحديد أي ثغرات أو أخطاء أو متطلبات المفقودة من الاحتياجات الفعلية.
- وفقا ل IEEE 1059 يمكن تعريف الاختبار بأنه عملية تحليل البرنامج للكشف عن الاختلافات بين الاحتياجات القائمة والمطلوبة وتقييم هذه البرنامج.

الاختبار
Testing

- تعد عملية الاختبار ذات أهمية بالغة حيث تستغرق حوالي 40% من الزمن الكلي لأعداد المنظومة في المنظومات الكبيرة والمعقدة.
- يجب أن يعمل كل من معد المنظومة وفريق الاختبار المستقل جنبا الى جنب للتأكد من خلو المنظومة من الاخطاء وذلك بتنفيذ الاختبارات الكافية.

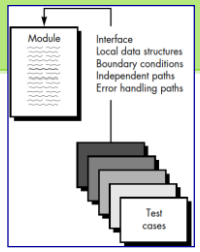


و يتم تصنيف 4 مستويات الاختبار:

1. اختبار الوحدة Unit Test.
2. اختبار التكامل. Integration Test
3. اختبار النظام System Test.
4. اختبار القبول

يجب التحقق من كل وحدة على حدة وضمن بقية النظام. إذا أظهر البرنامج نتائج مرضية في جميع المراحل ، فسيكون آخرها اختبار قبول يشير إلى ما إذا كان المنتج جاهزًا للسوق.

انواع الاختبارات
Type of Testing

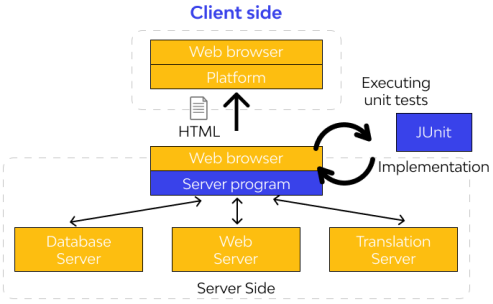


1) **اختبار الجزء البرمجي:Module Test**

- يطلق عليه اختبار الوحدةUnit Test
- يركز على اختبار أصغر وحدة في النظام Module من قبل المبرمج.
- تبدأ عملية اختبار الجزء البرمجي أثناء عملية البرمجة أي في المرحلة السابقة لمرحلة الاختبار .
- يجب تسليم الجزء البرمجي دون اخطاء

وصالحا للاستعمال لمدير البرمجة لأجراء

الاختبار التالي (الاختبار المتكامل)



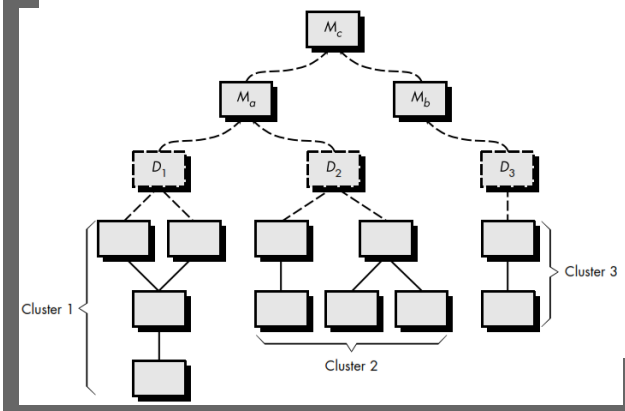
Type of Testing انواع الاختبارات

2) **اختبار التكامل:Integration Test**

- يتم تركيب واختبار كل الوحدات المجمعة من الاختبار السابق كوحدة واحدة.
- يتم ايضا اختبار Interfaces في هذا النوع من الاختبار.
- وكذلك يتم اختبار الزبون / الخادم وكذلك الانظمة الموزعة.

a. هذا النوع من الاختبار يسمى

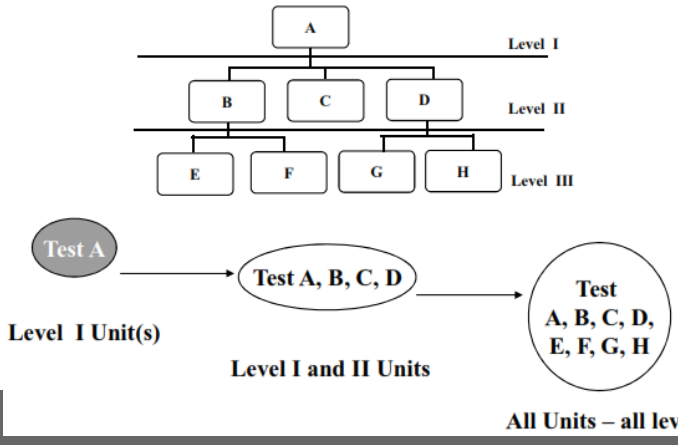
الاختبار من اسفل الى أعلىBottom-up



Type of Testing انواع الاختبارات

2) **اختبار التكامل:Integration Test**

b. هذا النوع من الاختبار يسمى **الاختبار من اعلى الى اسفلTop-down Test**



Type of Testing انواع الاختبارات

4) **اختبار النظامSystem Test**

- هواختبار لنظام مع الحاسوب بالكامل مثل أن النظام يعمل في بيئات مختلفة مثل أنظمة التشغيل.

a. **اختبار الحمايةSystem Test - Security Test**

- يتم التأكد من آلية حماية المنظومة من دخول غير مسموح الى المنظومة لأي سبب أو غرض.و كذلك قاعدة البيانات محمية من الهجمات الخارجية.

b. **اختبار الاستردادSystem Test - Recovery Test**

(4 يتم في هذا الاختبار التأكيد على قدرة المنظومة على استرداد قدرتها على العمل في حالة حدوث مشاكل بدون تلف للبيانات المخزنة، ومن أمثلة الاختبار:

- إعادة تشغيل أو تهيئة المنظومة.
- التخزين التلقائي في حالة حصول عطل.

Type of Testing انواع الاختبارات

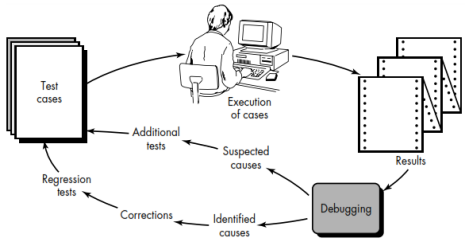
c. **اختبار القبول System Test - Acceptance Test**

- يطلق عليه أيضاً User Test بحيث يقوم المستخدم أو الزبون بإجراء هذا الاختبار ليتم اتخاذ القرار بخصوص أن المنتج يلبي المتطلبات المتوقعة ويتم قبول المنظومة.
- كما يقوم فريق ضمان الجودة باختبار مجموعة من السيناريوهات مكتوبة مسبقا وحالات الاختبار التي يستخدمها لاختبار التطبيق.

Methods of Testing طرق الاختبار

- **اختبار ألفا Validation Test – Alpha Test**

- هو اختبار للجزء البرمجي في مكان معد المنظومة في نهاية تطوير البرنامج ويتم تنفيذه بين فرق (المطور وضمان الجودة) حيث يقوم بتسجيل الاخطاء وتعديلها ثم يتم اعادة الاختبار مرة اخرى الى أن يتم الوصول الى منظومة تلبي احتياجات المستخدم.



Type of Testing انواع الاختبارات

c. **اختبار بيتا Validation Test – Beta Test**

- يتم إجراء هذا الاختبار بعد أن تم إجراء اختبار ألفا بنجاح ويبدأ هذا الاختبار في مقر الزبون حيث يختبر الزبون او المستخدم النهائي المنظومة الناتجة من الاختبار السابق بمفرده ويسجل الاخطاء قبل البدء في اصدار النسخة التجارية بحيث يتم اختبار التالي:
  - اختبار الوظائف : التأكد من صحة عمل كل وظائف المنظومة مثل اضافة صنف.
  - اختبار الكفاءة : اختبار زمن الاستجابة Response Time Test والتأكد من قدرة المنظومة على استيعاب كم هائل من السجلات .ويقوم معد المنظومة بالرجوع للزبون عدة مرات بشكل منتظم لمعرفة الاخطاء وتصحيحها وهكذا حتى تصبح المنظومة في وضع أفضل لقبولها.

Type of Testing انواع الاختبارات

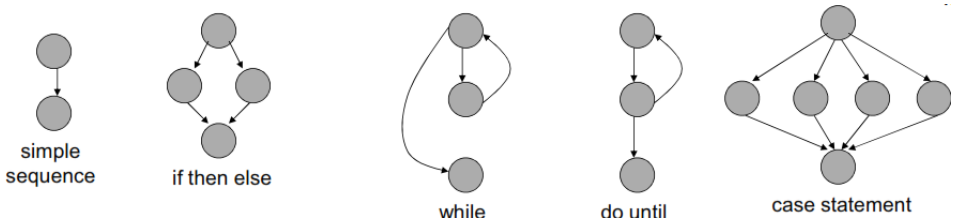
c. **اختبار بيتا – Beta Test**

- اختبار بيتا، أيضا هو **اختبار ما قبل النشر** .يتم توزيع الإصدارات اختبار بيتا على جمهور واسع على شبكة الإنترنت، وذلك لإعطاء البرنامج اختبار حقيقيا، حيث يقوم الجمهور بتثبيت وتشغيل التطبيق وإرسال ملاحظاتهم إلى فريق المشروع للحصول على ردود التي تمكن فريق المشروع من إصلاح المشاكل قبل إطلاق البرنامج للمستخدمين الفعلين.

Methods of Testingطرق الاختبار

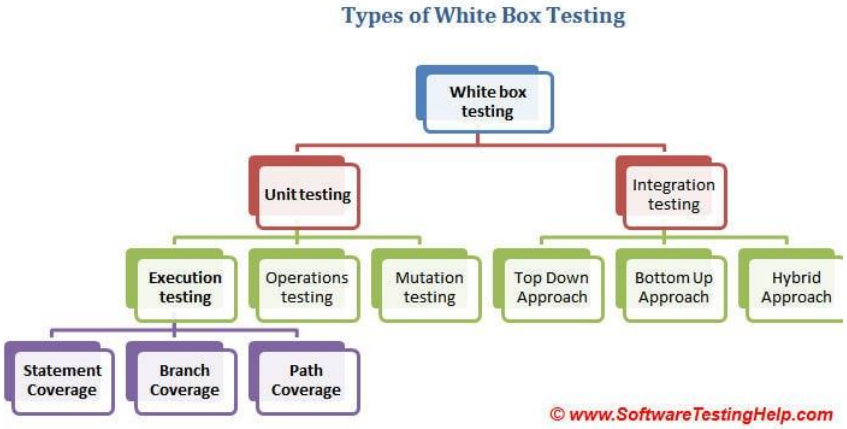
اختبار الصندوق الأبيض White-Box Test – Unit Test

- يطلق عليه ايضاً اختبار الصندوق الزجاجي.Glass Box
- يتم في هذا الاختبار إدخال بيانات متنوعة ومختلفة للتأكد من سلامة أداء كل مسارات الخوارزمية للجزء البرمجي الواحد.
- يحتاج اختبار لإلقاء نظرة داخل شفرة المصدر ومعرفة أي وحدة تتصرف بشكل غير صحيح.



Types of WBT

نركز بشكل أساسي على أنواع اختبار التنفيذ "لتقنية الصندوق الأبيض لاختبار الوحدة".

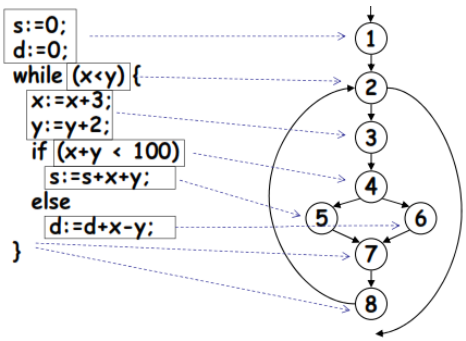


WBT: execution test

1. Statement coverage سنحتاج فقط إلى حالة اختبار واحدة للتحقق من جميع سطور الكود.
2. Branch coverage نظرًا لأن Statement coverage ليست كافية لاختبار الشفرة بالكامل ، فإننا سنطلب تغطية الفرع لضمان أقصى تغطية.
3. Path coverage
باستخدم تغطية المسار لاختبار مقتطفات التعليمات البرمجية المعقدة ، والتي تتضمن أساسًا عبارات الحلقة أو مجموعة من الحلقات وبيانات القرار

طرق الاختبار Methods of Testing

White-Box Test – Unit Test اختبار الصندوق الأبيض



المثال التالي نفرض أن البرنامج سيمر بعدة خطوات للتنفيذ:

وهذا الاختبار يوضح بأن node6 لن يتم تنفيذه.

To calculate the McCabe metric:

- Identify primitive conditions and operations
- Number them
- Drown the flow graph
- Compute the cyclomatic complexity

Test case #1: follows path 1-2-exit

Test case #2: 1-2-3-4-5-7-8-2-3-4-5-7-8-2-exit

Cyclomatic Complexity:التعقيد السيكلوماتي

- التعقيد السيكلوماتي: إنه مقياس للتعقيد المنطقي للبرنامج ويستخدم لتحديد عدد المسارات المستقلة.

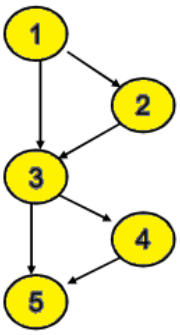
- For a graph G, V(G) is its cyclomatic complexity.
  Calculating V(G):
- V(G) = P + 1, P **عدد العقد المسند في الرسم البياني الانسيابي**
- **(العقدة التي تحتوي على الشرط)**

- V(G) = E – N + 2, E هو عدد الحواف و N/ هو العدد الإجمالي للعقد
- V(G) = Number of non-overlapping regions in the graph

Methods of Testing طرق الاختبار

White-Box Test – Unit Test اختبار الصندوق الأبيض

```
  int example1 (int value, boolean cond1, boolean cond2)
  {
1    if ( cond1 )
2       value ++;
3    if ( cond2 )
4       value --;
5    return value;
  }
```



Basis Paths	Test Data
1 3 5	false false
1 2 3 5	true false
1 2 3 4 5	true true

Methods of Testing طرق الاختبار

a. **اختبار الصندوق الأسود Black-Box Test – Validation Test**

- يتم اختبار وظيفة جزء البرمجي بحيث يتم التركيز فقط على مدخلات ومخرجات النظام .أي اختبار غير مدرك لبنية النظام وليس لديها إمكانية الوصول إلى شفرة المصدر ويسمى الاختبار الوظيفي functional أو السلوكي.behavioural

- فمثلا اختبار جزء برمجي يقوم بترتيب اسماء بحيث يتم إدخال مجموعة أسماء واختبار حالة الأسماء الناتجة ان كانت مرتبة أم لا.

Methods of Testing طرق الاختبار

a. **اختبار الصندوق الأسود Black-Box Test – Validation Test**

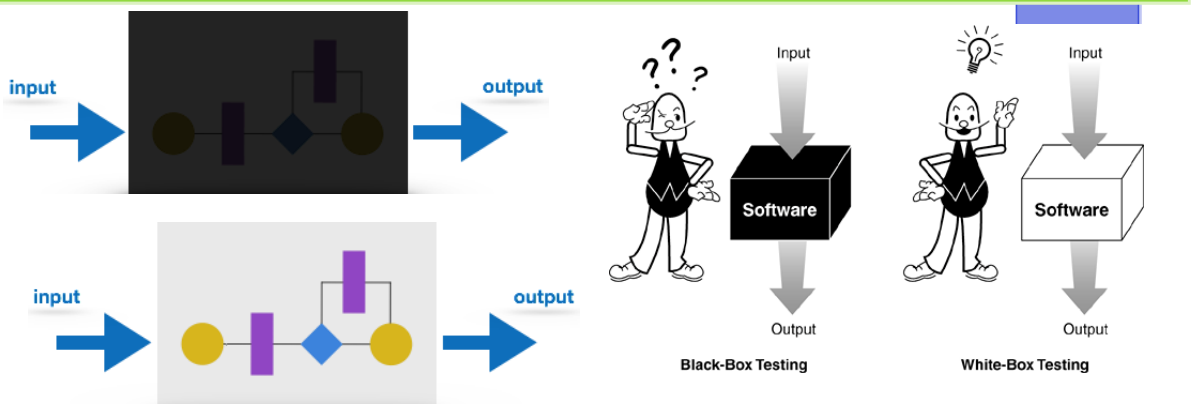
- مثال اختبار البحث عن موقع index لعنصر معين داخل المصفوفة ، على ان يتم ترجيع 1- في حالة عدم وجود القيمة في المصفوفة أو المصفوفة فارغة.

Example: Searching

Array	Value	Output
empty	5	-1
[7]	7	0
[7]	2	-1
[1,6,4,7,2]	1	0
[1,6,4,7,2]	4	2
[1,6,4,7,2]	2	4
[1,6,4,7,2]	3	-1

38

Black-box vs. White-box



Black box: test based on interface, through interface

White box: test based on code, through code (Control-flow testing, Data-flow testing

End

- **White Box Testing Tools**

VERACODE
Securing the Software That Runs the World

EMMA

googletest
Google C++ Testing Framework

JUnit 5

JSUnit.net
Java Script Testing Tool