**الاسم : أيوب قاسم الأيوبي**

**رقم القيد : 216010366**

**What is Test Driven Development (TDD)?**

**هو يستخدم في عملية تطوير البرامج والذي يركز على تطوير حالات الاختبار قبل تنفيذ الكود الفعلي ثم يقوم إعادة هيكلة الكود لجعله أكثر بساطة وسهولة استخدام وصيانة من خلال اتباع هذه العملية يمكنك التأكد من أن الكود الخاص بك يلبي المتطلبات ولا يحتوي على أخطاء ويكون عالي الجودة و يتم تحديد الاختبارات بناءً على المتطلبات الوظيفية للتطبيق والتي : تتمثل في هذه الخطوات**

1. **كتابة اختبار الفشل:  ابدأ بكتابة اختبار والذي يمثل في الوظيفة المطلوبة، وتأكد من فشل هذا الاختبار قبل تنفيذ اي كود**
2. إعادة هيكلة الكود : **بمجرد اجتياز الاختبار الفشل يقوم المطور بإعادة هيكلة الكود لتحسين القراءة والهيكلة والكفاءة وما إلى ذلك دون تغيير شكله الخارجي**
3. تنفيذ الاختبار : يتم تشغيل الاختبار والتحقق مما إذا كان يفشل أو ينجح

* **: مزايا TDD**
* **على اكتشاف الأخطاء مبكرا وتمنع تراجع الميزات لاحقا TDD كود أكثر موثوقية:  تساعد**
* **على التفكير في واجهات برمجة التطبيقات جيدة التصميم قبل كتابة الكود مما يؤدي TDD تصميم أفضل: يشجع**

**الي تصميم أفضل بشكل عام**

* **كود قابل للصيانة:  يؤدي التركيز على اختبارات الوحدة السهلة القراءة إلى كود أكثر قابلية للصيانة وتحديثا في المستقبل**
* **توثيق أفضل:  تعتبر اختبارات الوحدة نفسها بمثابة توثيق للكود، مما يسهل على المطورين الآخرين فهمه**

**What are the important principles of TDD?**

1. **يتم كتابة حالات الاختبار قبل كتابة اي كود الفعلي TDD كتابة الاختبار أولاً: وهو المبدأ الأساسي لـ**
2. **كتابة الكود اللازم لاجتياز الاختبار: يجب أن يقتصر كتابة الكود اللازم لتلبية الاختبار**
3. **عدم إضافة ميزات جديدة دون اختباراها : يجب تأكيد أية وظائف أو ميزات جديدة من خلال اختبارها قبل التنفيذ**
4. **الاختبارات مستقلة ومعزولة: لا يجب أن تعتمد حالات الاختبار على بعضها تقدم الاختبارات المعزولة مردوداً دقيقاً**
5. **إعادة صياغة الكود باستمرار: تعمل المطورون باستمرار على إعادة صياغة وتحسين كود الإنتاج دون تغيير شكله الخارجي**
6. **جعل الاختبارات قابلة للقراءة و قابل الصيانة: يجب أن تكون حالات الاختبار واضحة و ساهلة لفهم مقاصدها وقدرة أي شخص على صيانتها**
7. **التأكد من تغطية الكود بالاختبارات: مقدار الكود الذي تغطيه حالات الاختبار يجب أن يكون أكبر ما أمكن للتحقق من جميع التوقعات**
8. **مبدأ الفشل المبكر: الفشل المبكر جيد حيث يمنع دمج كود ضخم غير سليم**
9. **الفشل دليل على إعادة العمل: تدفع حالات الفشل في الاختبارات إعادة صياغة التصميم والكود للتخلص من الأخطاء**

**WhyuseTDDandwhat are the benefits?**

**يوفر طريقة منهجية لتطوير برامج ذات جودة أعلى وموثوقية أكبر وبشكل أسرع وأقل تكلفة**

**ويستخدام عديد من الأسباب :-**

* 1. **على كتابة اختبارات دقيقة تحدد البرنامج قبل تنفيذ كود الفعلي TDD جودة الكود أفضل: يركز**

**وهذا يؤدي إلى كتابة كود أكثر قابلية للاستخدام**

* 1. **توثيق الاختبارات: تعتبر حالات الاختبار موثقة لمتطلبات النظام وتمنع حدوث أخطاء عند إجراء تغييرات لاحقة**
  2. **اختبارات متكررة: تضمن الاختبارات المتواصلة جودة الكود وتقلل من حدوث الأخطاء**
  3. **الفشل المبكر: تكشف حالات الفشل في مرحلة تطوير مبكرة من عملية التطوير**
  4. **يسرع عملية التطوير: على الرغم من أن كتابة الاختبارات في البداية قد تبدو تستغرق وقتًا إضافيًا إلا أن يمكن أن يقلل من الوقت المستغرق في إصلاح الأخطاء في وقت لاحق TDD**
  5. **التعاون بين أعضاء الفريق TDD يعزز التعاون:  يمكن أن يعزز**

**benefits TDD ?**

1. **زيادة الإنتاجية: يمكن أن يؤدي استخدامها إلى زيادة في الإنتاجية على المدى الطويل**
2. **تحسين التواصل:يمكن أن يساعد على تحسين التواصل بين المطورين**
3. **تقليل التكاليف: يمكن أن يقلل من تكاليف تطوير وصيانة البرنامج على المدى الطويل**

**What are the drawbacks of TDD?**