

مراجع المحاضرة : كتاب هندسة البرمجيات - تأليف د.أحمد شعبان دسوقي - أ.د.السيد محمود الريبيعي



## محتويات الوحدة

الصفحة	الموضوع
49	مقدمة
49	تمهيد
50	أهداف الوحدة
53	1. المرحلة الأولى مرحلة تجميع المتطلبات
55	2. المرحلة الثانية مرحلة التطوير
55	1.2 مرحلة التصميم
56	2. 2 مرحلة التنفيذ
57	3. المرحلة الثالثة مرحلة الاختبار وتشخيص الأخطاء
57	1.3 الخطوة الأولى التحقق (Verification)
58	2.3 الخطوة الثانية المصادقة (Validation)
60	4. المرحلة الرابعة مرحلة الصيانة والارتقاء
60	5. نشاطات المظلة (Umbrella Activities)
60	1.5 إعداد الوثائق وإنتاجها
61	2.5 متابعة المشروع البرمجي ومراقبته
61	3.5 ضمان جودة المنتج البرمجي
62	4.5 إدارة تكوين البرنامج
62	5.5 إدارة المخاطر
65	الخلاصة
66	للمحة مسبقة عن الوحدة التالية
67	إجابات التدريبيات
59	مسرد المصطلحات
72	المراجع

## مقدمة

### تمهيد

عزيزي الدارس ،

أهلاً بك في هذه الوحدة من مقرر " هندسة البرمجيات (١)" هذه الوحدة تتناول دورة حياة (عمليات) البرمجيات ، وهي دراسة تفصيلية للمراحل المختلفة لدورة حياة البرمجيات. تركز المرحلة الأولى من مراحل حياة (عمليات) البرمجيات على دراسة متطلبات حل المشكلة، وتنتهي هذه المرحلة بكتابه "وثيقة توصيف المتطلبات" التي تشمل المواصفات الفنية للمشروع البرمجي بناءً على متطلبات العميل.

تنقسم المرحلة الثانية إلى مراحلتين : مرحلة التصميم حيث تعد "وثيقة توصيف المتطلبات" من معطيات هذه المرحلة و مرحلة التنفيذ حيث تعد أيضاً "وثيقة توصيف المتطلبات" من معطيات هذه المرحلة و تتم هذه المرحلة على خطوتين ؛ الأولى عملية البرمجة والثانية عملية التكامل.

المرحلة الثالثة من مرحلة الاختبار وتشخيص الأخطاء وهي مرحلة مستمرة خلال جميع مراحل تطوير المشروع البرمجي، وتشمل هذه المرحلة خطوتين : الأولى خطوة التحقق ، والثانية خطوة المصادقة.

المرحلة الرابعة وهي مرحلة الصيانة والارتقاء حيث تبدأ من هذه المرحلة أهم مراحل عمر المنتج وهي مرحلة الصيانة.

وهي عبارة عن مجموعة من الأنشطة المستقلة عن أي نشاط برمجي هيكلی وتجري طوال عملية بناء المنتج البرمجي.

تجد عزيزي الدارس في ثابا الوحدة بعض التدريبات التي يساعدك تنفيذها على فهم محتوى المادة هذا بالإضافة إلى أسئلة التقويم الذاتي التي تهدف إلى ترسيخ الفهم وتعزيزه لديك و تعديله .

## أهداف الوحدة



في نهاية هذه الوحدة ينبغي أن يكون قادراً على أن :

- تعدد عمليات البرمجيات.
- تصف هيكلأً عاماً لعمليات البرمجيات.
- تشرح دورة حياة البرمجيات.
- تدون أهم مراحل تطوير البرمجيات وأهمية كل منها في عملية التطوير.
- تحدد مدخلات ومخرجات مرحلة تجميع المتطلبات.
- تصف بعض الطرق الممكن استخدامها في مرحلة تجميع المتطلبات.
- تحدد مدخلات ومخرجات مرحلة التصميم.
- تصف بعض الطرق الممكن استخدامها في عملية تصميم البرمجيات.
- تحدد مدخلات ومخرجات مرحلة التصميم (البرمجة والتكامل).
- تشرح الفرق بين عمليتي المصادقة على والتحقق من صلاحية البرمجيات .
- تسرد أهم الأخطاء التي يمكن اكتشافها أثناء عملية اختبار البرمجيات.
- تعبر عن أهمية صيانة البرمجيات ، ويسرد أهم الأعمال التي قد تشملها.
- تعبر عن أهمية نشاطات المظلة في عملية تطوير البرمجيات ، ويسرد أهمها.

## توطئة

سبق وأن عُرِفنا في الوحدة الأولى عمليات البرمجيات بأنها : هي مجموعة من الأنشطة المرتبطة (الطبقية) المتراقبة (Constraints Activities) والقيود (Consistency Activities) والموارد (Resources) التي تهدف إلى توصيف (Specifying) وتصميم (Designing) وتنفيذ (Implementing) وختبار (Testing) وصيانة وترقية (Maintenance and Evolution) منتج برمجي يفي بجميع متطلبات العميل، غالباً ما يقوم بهذه الأنشطة مهندسو البرمجيات. ويمكن وصف عمليات البرمجيات بوضع هيكل عام لها (Common Process Framework) ، وذلك من خلال وضع عدد محدد من نشاطات الهيكل (Framework Activities) والتي يمكن تطبيقها على جميع المشاريع البرمجية بغض النظر عن حجمها أو تعقيدها ، وعدد منمجموعات المهام (Task Set) الالزام للكل نشاط هيكل ، هذا بالإضافة إلى مجموعة من نشاطات هندسة البرمجيات والتي يطلق عليها " نشاطات المظلة "، وهي عبارة عن مجموعة من الأنشطة مستقلة عن أي نشاط برمجي هيكل وتحرج طوال عملية بناء المنتج البرمجي (مثل : متابعة المشروع البرمجي ومراقبته ، ضمان جودة المنتج البرمجي ، إدارة تكوين البرنامج ، إدارة المخاطر ، وخلافه) ، كما سيتم توضيحه فيما بعد.

فمثلاً ، يُعد الاتصال بالعميل نشاط هيكل يحتاج إلى مجموعة من المهام منها على سبيل المثال : إعداد وثيقة عمل وجداول أعمال للجتماع الرسمي مع العميل ، وإجراء أبحاث لتحديد الحلول المقترحة والطرق الموجودة لمعالجة الموضوع وتجمیع متطلبات العميل ، مراجعة متطلبات العميل مع جميع المعنيين بالأمر ، وخلافه من المهام.

وتشير تلك العمليات (Software Process) - في الوقت نفسه . إلى مراحل تطوير البرمجيات. وبمفهوم علم هندسة البرمجيات (Software Engineering) يطلق على تلك العمليات - أيضاً - اصطلاح عام وهو دورة حياة البرمجيات (Software Life Cycle)

. وعلم هندسة البرمجيات هو علم يهدف إلى إنتاج برمجيات خالية من الأخطاء وذات جودة عالية في وقت محدد وبميزانية محددة ، وبطريقة اقتصادية بحيث يفي بجميع متطلبات الجهة المستفيدة.

وبغض النظر عن مجال التطبيق أو حجمه أو درجة تعقيده ، وبدون الدخول في تفصيلات النماذج (Models) التي تصنف عمليات المراحل مثل : Waterfall Model و Spiral Model ، والتي سوف نتناولها في الوحدة الثالثة من هذا الكتاب ، يمكن تقسيم مراحل تطوير البرمجيات إلى المراحل الرئيسية التالية :

- ① مرحلة تحليل المشكلة (Problem Analysis Phase).
- ② مرحلة التطوير (Development Phase) وتشمل مرحلتي التصميم (Implementation Phase) و التنفيذ (Design Phase).

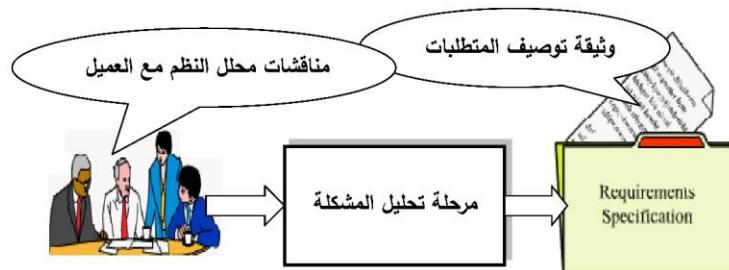
③ مرحلة الاختبار وتشخيص الأخطاء (Testing and Debugging Phase).  
ويأتي بعد كل هذه المراحل أهم مرحلة وأطولها وأكثرها تكلفة وهي :

- ④ مرحلة صيانة وترقية البرنامج (Maintenance Phase).

وتعُد نتائج كل مرحلة من هذه المراحل معطيات للمرحلة التي تليها ، فمثلاً نتائج مرحلة تحليل المشكلة هي معطيات لمرحلة التصميم ، ونتائج مرحلة التصميم هي معطيات لمرحلة التنفيذ ، وهكذا وسوف نتناول الآن هذه المراحل بإيجاز وبدون الدخول في التفاصيل الخاصة بكيفية إنجاز كل مرحلة ، حيث إننا سوف نتناول معظم هذه المراحل بالشرح والتفصيل في باقي أجزاء هذا الكتاب.

## 1. المرحلة الأولى: مرحلة تجميع المتطلبات

تركز هذه المرحلة (أو العملية) على دراسة متطلبات حل المشكلة ، حيث إنها تُعد مرحلة التوصيف الدقيق للمشروع البرمجي وتهيئه لمرحلة التصميم النهائي، ويتم ذلك عبر إعداد مجموعة من المخططات (Charts) والنمذاج (Models) واللواائح (Lists) من خلال طرح مجموعة من الأسئلة المتخصصة من قبل محلل النظم للعميل لمعرفة كافة احتياجات وشروط المشروع البرمجي ، وذلك طبقاً لمنهجية محددة تسمى منهجية التحليل، وتنتهي هذه المرحلة بكتابة وثيقة رسمية (**Formal Document**) تشتمل على المواصفات الفنية للمشروع البرمجي بناءً على متطلبات العميل مثل نطاق المعلومات الخاص بالمشروع (**Information Domain**) والأداء الوظيفي (**Functionality**) ، الربط البيني بين البرمجية والمستخدمين (**User Interface**) ، ومستويات الأمان المطلوبة (**Security Levels**) ، بالإضافة إلى المتطلبات الاستثنائية (**Exceptional Requirements**). ويطلق على هذه الوثيقة "وثيقة توصيف المتطلبات" (**Requirements Specification Document**) ، والتي ستكون بمثابة العقد النهائي بين مطوري المشروع البرمجي والعميل. والشكل رقم 2.1 يمثل شكل توضيحي لهذه المرحلة ، حيث إن معطيات هذه المرحلة تتمثل في متطلبات العميل ونتائجها تتمثل في كتابة "وثيقة توصيف المتطلبات".



الشكل رقم 2.1 : شكل توضيح لمرحلة تحليل المشكلة.

- وبصفة عامة تتضمن مرحلة (أو عملية) تحليل المشكلة التعريف بالمشكلة من خلال العديد من التواحي والتي من أهمها :
- **المدخلات (Inputs):** يجب على المحلل معرفة الطرق التي يفضلها العميل في عملية إدخال البيانات إلى النظام ، فهل عملية إدخال البيانات سوف تتم عن طريق القراءة من ملف ، أم سوف تكون تفاعلية بين المستخدم والنظام عن طريق قوائم اختيار بواسطة الفأرة ، أو تكون تفاعلية ولكن بطريقة مفتوحة عن طريق لوحة المفاتيح ، أو غيرها من طرق إدخال البيانات.
  - **المخرجات (outputs):** يجب - أيضاً - على المحلل الاهتمام بمعرفة متطلبات العميل من حيث شكل وهيئة نماذج المخرجات (تقارير النظام) وصيغها المختلفة ، هل تكون على شكل أرقام فقط أو أرقام وحروف ، أو خليط من الأرقام والحوروف والرسومات ، أو ما شابه ذلك.
  - **المعالجة (Processing) :** يجب أن يدقق المحلل في كيفية عملية المعالجة من حيث نوعية البيانات والحسابات والقرارات التي سوف تعمل على هذه البيانات لإنجاز المخرجات المطلوبة ، وكذلك من حيث بيئة التشغيل التي سوف يتم تشغيل النظام من خلالها (نظام التشغيل (Operating System) ، وهل عملية التشغيل سوف تتم على حاسب آلي شخصي (Personal Computer) ، أو من خلال استخدام الشبكات المحلية (LANs).

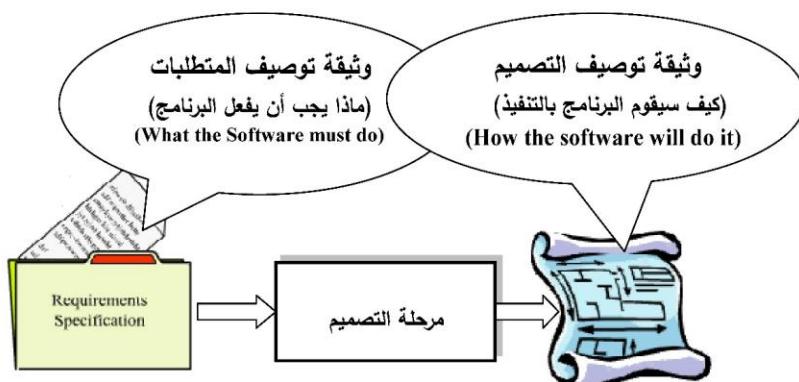
## 2. المرحلة الثانية مرحلة التطوير

تنقسم مرحلة التطوير إلى مراحلتين هما :

### 1.2 مرحلة التصميم

تُعد وثيقة توصيف متطلبات العميل هي معطيات هذه المرحلة (أو العملية) ، حيث يتم توزيع المهام على المختصين من فريق العمل من المصممين ، الذين يقومون في هذه المرحلة بتحديد الخوارزميات التي سوف يستخدمونها ، وكذلك رسم مخططات التصميم التي تتناسب مع المتطلبات المتفق عليها سابقاً مع العميل ، فهناك العديد من القوالب والنماذج يتم التصميم على أساسها ، فتصنف بعضها حسب تحليل البيانات وعرضها ، والبعض حسب التسلسل الزمني أو الفترة الزمنية المحددة ، وأخرى على حسب بيئه التصميم وغيرها، وتشتمل خريطة التدفق للبيانات (Data Flowchart) إحدى أهم الطرق المستخدمة في مرحلة التصميم ، حيث تستخدم لتوضيح المدخلات والمخرجات وتسلسل العمليات والمعدات المستخدمة لحل المشكلة، فإذا كانت المشكلة معقدة وترتبط بأقسام عديدة تخص العميل فإنه يتم إعداد ما يسمى بخريطة التدفق للنظام (System Flowchart) والتي توضح العلاقات المشتبعة التي تربط بين هذه الأقسام المختلفة ونظام تدفق البيانات بين هذه الأقسام بعضها البعض ، وعلى هذه الخريطة يتم إيصال محظات العمل والأفراد العاملين والمعدات وشكل الوثائق والأقسام المؤثرة في هذه الخريطة يمكن للمبرمج استخدام خريطة التدفق المختصرة أو خريطة التدفق المفصلة وأيضاً من الأدوات التي يمكن للمصمم استخدامها لشرح التعليمات المكونة للخوارزمية قائمة القرارات (Decision Table) التي يتم فيها بيان العلاقات المختلفة وأسباب ومكان التفريع للعمليات هذا وبعد إتمام عملية التصميم يتم تسجيل كل ما يتعلق بهذه المرحلة في "وثيقة توصيف التصميم" (Design Specification Document) ، مثل تحديد مكونات البرمجية والبناء الهيكلي للبرمجة وربط البياني مع المستخدمين ، هيكل وجداول البيانات ،

والخوارزميات ، وذلك بحيث يضمن إنجاز الوظائف الأساسية لتحقيق متطلبات العميل .  
والشكل رقم 2.2 يوضح هذه المرحلة.



شكل 2.2 : شـكل توضيحي لمرحلة التصميم.

## 2.2 مرحلة التنفيذ

تُعد وثيقة توصيف متطلبات التصميم هي معطيات هذه المرحلة (أو العملية) ، حيث يتم ترجمتها إلى منتج برمجي متكامل باستخدام إحدى لغات البرمجة المناسبة أو باستخدام أدوات البرمجيات المعتمدة على الحاسوب (CASE Tools) على أيدي فريق من المبرمجين ، وتنتمي هذه المرحلة على خطوتين :

- الخطوة الأولى : وهي عملية البرمجة (Programming) ، حيث يتم خلالها تجزئة المشروع البرمجي إلى أجزاء تركيبية صغيرة (Modules) للتعقب على التعقيدات (Complexities) التي يمكن أن تنشأ في حالة التعامل معه ككتلة برمجية واحدة، وبعد ذلك توزع هذه الأجزاء على فريق المبرمجين ، بحيث يكون كل مبرمج مسؤولاً عن برمجة واختبار وتوثيق جزئية معينة من هذه الأجزاء ، بطريقة تسمح بالتكامل مع الأجزاء الأخرى .

### تدريب (1)

تم مرحلة التنفيذ للبرمجية على خطوتين أساسيتين ، ارسم شكلاً توضيحيًا يبين كلاً من الخطوتين.



- الخطوة الثانية : وهي عملية التكامل (Integration) ، حيث يتم خلاها تجميع جميع أجزاء البرنامج المنفصلة (Individual Modules) التي تم برمجتها في الخطوة السابقة لتصبح منتجًا برمجيًا متكاملًا (Integrated Software Product).

## 3. المرحلة الثالثة: مرحلة الاختبار وتشخيص الأخطاء

مرحلة (أو عملية) الاختبار ليست مرحلة منفصلة تنفذ بعد اكتمال المشروع البرمجي أو بعد انتهاء كل مرحلة فقط ، بل يجب أن تتفق باستمرار خلال جميع مراحل تطوير المشروع البرمجي، وبصفة عامة ، تشتمل مرحلة الاختبار على خطوتين :

### 1.3 الخطوة الأولى: وهي خطوة التحقق (Verification)

وهي تتم في كل مرحلة من مراحل تطوير البرنامج، ويقصد بالتحقق في مرحلة معينة - هنا - تحديد إن كان البرنامج قد تم تحويله من المرحلة السابقة إلى هذه المرحلة بكفاءة وبدقة عالية وبدون أخطاء وإنه يحقق متطلبات المرحلة السابقة أم لا ، فعلى سبيل المثال فإن مرحلة التنفيذ تُحول وثيقة توصيف التصميم إلى برنامج متكامل قابل للتنفيذ (Executable Integrated Software) ، لذا يقصد بالتحقق في هذه المرحلة تحديد إن كان البرنامج قد تم بناؤه بالشكل الذي يحقق جميع المتطلبات الموصفة في وثيقة توصيف التصميم وبدون أي نوع من الأخطاء أم لا ، ويعرف العاملون في مجال هندسة البرمجيات التتحقق (Verification) بأنه إجراء جميع الاختبارات الالزامية لكل مرحلة للإجابة على السؤال التالي : هل نحن نبني البرنامج بصورة صحيحة ؟ (Are we building the software right?)

البرنامج .

## 2.3 الخطوة الثانية وهي خطوة المصادقة (Validation) :

وهي تتم بعد تطوير البرنامج ، ويقصد بالمصادقة – هنا – تقويم البرنامج للتأكد من أنه قد تم تصميمه بالطريقة التي يتوقعها ويرضى بها العميل وبدون أي أخطاء من أي نوع، ويعبر العاملون في مجال هندسة البرمجيات التحقق (Validation) بأنه إجراء جميع الاختبارات الازمة للإجابة على السؤال التالي : هل نحن نبني البرنامج الصحيح ؟ (Are we building the right software?) ، وخطوة المصادقة هنا تتم من خلال وجهة نظر العميل من البرنامج.

وعملية تشخيص الأخطاء (Debugging) تختلف عن عملية اختبار العيوب (Defect Testing) ، حيث إن عملية اختبار العيوب تهتم فقط بإثبات وجود عيوب بالبرنامج من عدمه بدون تحديد موضع أو كيفية إصلاح هذا العيب في حالة وجود عيوب. أما عملية تشخيص الأخطاء فهي تهتم بتحديد مواضع هذه العيوب وإصلاحها في نفس الوقت.

وسوف نتناول الآن أهم أنواع الأخطاء التي يمكن أن تنشأ أثناء مرحلة الاختبار وتشخيص الأخطاء ، حيث يمكن تصنيفها إلى الأنواع التالية :

① **أخطاء قواعد معاني اللغة (Syntax and Semantic Errors)** : وتنتج هذه الأخطاء عن كتابة تعليمات لا يتم فيها مراعاة قواعد ومعاني لغة البرمجة المكتوب بها البرنامج ، ويتم اكتشافها أثناء عملية ترجمة البرنامج المصدر (Source Code) لإنتاج البرنامج الهدف (Object Code) ، وذلك باستخدام مترجم اللغة (Compiler).

② **أخطاء وقت التنفيذ (Run Time Errors)** : تظهر هذه الأخطاء أثناء وقت التنفيذ من خلال نظام التشغيل ، رغم من ترجمة البرنامج بنجاح بدون أخطاء في قواعد ومعاني اللغة، فالبرنامج قد يكون صحيحاً من ناحية قواعد ومعاني اللغة وبحتوى على جمل صحيحة ، ولكنها تسبب أخطاء عند تنفيذ البرنامج، من هذه الأخطاء مثلاً القسمة على الصفر تعطي قيمة لانهائية أو القسمة على قيمة كبيرة جداً تعطي قيمة صغيرة جداً ولا يمكن في أي لغة برمجة تمثل هذه القيم (Over-Flows and Under-Flows Errors)

غير معرف مسبقاً ، أو أنه قد دخل في حلقة تكرارية غير منتهية (Open Loops Errors). وبمجرد اكتشاف هذه الأخطاء يقوم نظام التشغيل بوقف تنفيذ البرنامج واعطاء رسالة تنفيذ بوجود الخطأ ففي هذه الحالة يجب تشغيل مشخص ومصحح الأخطاء (Debugger) الخاص بنفس اللغة المستخدمة لتشخيص مواضع هذه الأخطاء وتصحيحها.

② **الأخطاء المنطقية (Logical Errors)** : هي للأسف ليست أخطاء لغوية يمكن للمترجم اكتشافها أو أخطأوا في التنفيذ يمكن اكتشافها أثناء التنفيذ ، ولكنها أخطاء في تركيب المدخلات من بيانات أو في المعادلات الرياضية ، وبالتالي تكون المخرجات غير متوقعة أو غير منطقية وغير مقبولة والسبب في هذه الأخطاء هو المبرمج الذي لا يتأكد من المدخلات أو لا يتأكد من وضع المعادلات الرياضة التي سوف تتم المعالجة على أساسها ، أو أن يضع علامة القسمة بدلاً من الضرب أو علامة الطرح بدلاً من الجمع ، وخلافه.

وتعُد الأخطاء المنطقية من أصعب الأخطاء في اكتشافها ، حيث إنه في حالة وجود مثل هذه الأخطاء يجب فحص البرنامج ومطابقته مع مستندات التصميم حتى يمكن معرفة موقع هذه الأخطاء وتصحيحها.

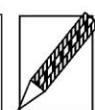
## تدريب (2)

ما الفرق بين التحقق والمصادقة؟ أيهما يجيب عن السؤال التالي؟ :

**Are we building the software right?**

وأيهما يجيب عن السؤال التالي؟ :

**Are we building the right software?**



## أسئلة تقويم ذاتي

صنف أهم أنواع الأخطاء التي يمكن أن تنشأ أثناء مرحلة الاختبار وتشخيص الأخطاء؟



## 4. المرحلة الرابعة: مرحلة الصيانة والارتقاء

كل المنتجات البرمجية الناجحة هي التي تُطَوِّر مع الوقت للتتوافق مع التغيرات التي يريدها العميل ، لذا فإن عملية تطوير البرنامج لا تنتهي بتسليم المنتج النهائي للعميل ، بل تبدأ مرحلة من أهم المراحل في عمر المنتج وهي مرحلة الصيانة ومن أهم الأعمال التي قد تشملها مرحلة الصيانة ما يلي :

- ① معالجة الأخطاء التي قد تنشأ مع التشغيل (Bugs Fixing).
- ② إضافة عمل وظيفي جديد للبرنامج (Add New Functionality).
- ③ إضافة تقارير خرج جديدة (Add New Output Reports).
- ④ تهيئة البرنامج للعمل على أنظمة تشغيل جديدة (Adapt the software to new Platforms).

## 5. نشاطات المظلة (Umbrella Activities)

يجري تكامل المراحل السابقة وعملياتها المختلفة في إطار هندسة البرمجيات بعدد من نشاطات المظلة ، والتي تُجرى طوال عملية تطوير المنتج البرمجي ، ومن أهم هذه النشاطات :

### 1.5 إعداد الوثائق وإنتاجها

ويطلق عليها مرحلة التوثيق ، وهي ليست مرحلة منفصلة تتفذ بعد اكتمال المشروع البرمجي ، بل تبدأ هذه المرحلة مع بداية المرحلة الأولى من مراحل تطوير البرنامج إلى أن يتم تسليم كامل المنتج البرمجي للعميل (Delivery Phase) ، والتي يمكن أن تُعد آخر مرحلة في تطوير البرنامج والتي يبدأ بعدها مرحلة الصيانة.

إن التوثيق السليم للبرنامج يساعد على الوقوف على آخر ما تم من تعديلات وأسبابها ، وبالتالي تكون الرؤية واضحة للتعديل الجديد ؛ لذلك يجب أن يتضمن التوثيق السليم للبرنامج ما يلي :

- ① تعریف المشكلة أو تحديد الهدف من البرنامج (وثيقة توصیف المتطلبات).
- ② وثائق ومخططات تصمیم البرنامج (وثيقة توصیف التصمیم).
- ③ تسجیل الملاحظات عن الخطوات والإجراءات الهامة في شفرة البرنامج ، فهذه الملاحظات تساعد المبرمج وقارئ البرنامج على فهم البرنامج وسهولة تتبعه واكتشاف الأخطاء وتصحیحها.
- ④ متطلبات تنفيذ البرنامج من بيته التشغيل وحدات الإدخال والإخراج المختلفة.
- ⑤ نسخة من برنامج المصدر.
- ⑥ عينة من نتائج البرنامج .

## 2.5 متابعة المشروع البرمجي ومراقبته

تُعد إدارة المشاريع البرمجية من نشاطات المظلة ضمن هندسة البرمجيات ، حيث إنها تبدأ قبل الشروع بأية نشاط تقني وتستمر طوال تعريف البرنامج وتطويره وصيانته.

## 3.5 ضمان جودة المنتج البرمجي

وذلك من خلال أخذ القياسات الضرورية طوال بناء المشروع البرمجي للمساعدة في تقويم جودة المنتج البرمجي واتخاذ القرارات التصحيحية أثناء تقديم المشروع ومن أهم القياسات المباشرة لعملية هندسة البرمجيات قياس الكلفة والجهد المطبقين ، و عدد أسطر الشيفرة البرمجية المنتجة، وسرعة التنفيذ و حجم الذاكرة المطلوب لعملية التنفيذ و عدد الأعطال المعلن عنها خلال فترة زمنية معينة ، هذا بالإضافة إلى قياسات المنتج غير المباشرة مثل درجة التعقيد ، والفاعلية ، والموثوقية ، وقابلية الصيانة ، وخلافه من صفات وخصائص المنتج البرمجي الجيد المذكورة في الوحدة السابقة.

## 4.5 إدارة تكوين البرنامج

هي نشاط مطلة يبدأ مع بداية المشروع البرمجي وينتهي فقط عند التوقف عن استخدام البرنامج ، والغرض منها هو إدارة التعديلات الحاصلة على البرنامج الذي يبنيه فريق البرمجة والتحكم فيها خلال كامل دورة حياة البرنامج بعرض تحسين السهولة التي تجرى بها التغييرات وتخفيف مقدار الجهد المبذول عندما يجب إجراء تلك التغييرات ، وذلك لزيادة الإنتاجية إلى حدتها الأقصى وتقليل الأخطاء إلى حدتها الأدنى، وهي أيضاً مسؤولة عن تعين هوية كل بند من بنود تكوين البرنامج (البرنامج المصدر **(Source Code)** ، والبرنامج التنفيذي **(Executable Code)** وثائق تصف البرنامج موجهة للمستخدمين والممارسين التقنيين ، وخلافه من الوثائق والبيانات الأخرى المتعلقة بالبرنامج .).

## 5.5 إدارة المخاطر

وذلك من خلال تحديد المهام اللازمة لتقديم المخاطر التقنية والإدارية وفهمها والقيام بالإجراءات المناسبة لها في كل مرحلة من مراحل تطوير المنتج البرمجي.

## أسئلة تقويم ذاتي



1. يتم تكامل مراحل تطوير البرمجيات وعمليات المختلفة بعدد من النشاطات التي تجرى طوال عملية تطوير المنتج البرمجي والتي يطلق عليها "نشاط المظلة" ما هي أهم هذه النشاطات.

2. وضع (✓) أمام الإجابة الصحيحة وعلامة (X) أمام الإجابة الخطا

- علم هندسة البرمجيات علم يهدف إلى إنتاج برمجة خالية من الأخطاء و ذات جودة عالية في وقت محدد بميزانية محددة وبطريقة اقتصادية بحيث يفي بجميع متطلبات الجهة المطورة.

- الاتصالات بالعميل يعد نشاطاً هيكلياً يحتاج إلى مجموعة من المهام منها (إعداد جداول أعمال للاجتماع الرسمي مع العميل ، تجميع متطلبات العميل ، مراجعة متطلبات العميل.....)

- نشاطات المظلة وهي عبارة عن مجموعة من الأنشطة مستقلة على أى نشاط برمجي هيكلي وتجرى طوال عملية بناء المنتج البرمجي.

- مرحلة تجميع المتطلبات تعد مرحلة التوصيف الدقيق للمشروع البرمجي وتهيئة لمرحلة التصميم النهائي.

- الوثيقة الرسمية تشتمل على المواصفات الفنية للمشروع البرمجي بناء على متطلبات العميل.

- وثيقة توصيف المتطلبات هي معطيات مرحلة التنفيذ.

- في مرحلة التصميم يتم تحديد الخوارزميات التي ستستخدم وكذلك رسم مخططات التصميم التي تتناسب مع المعطيات.
- وثيقة توصيف التصميم تعتبر معطيات أساسية لمرحلة التنفيذ.
- في كل الأحوال يمكن ترجمة وثيقة النصميم إلى منتج برمجي باستخدام CASE Tools.
- مرحلة الاختيار وتشخيص الأخطاء مرحلة منفصلة تنفذ بعد اكمال المشروع البرمجي.
- ليس هناك فرق على الإطلاق بين عملية تشخيص الأخطاء (Defect Test) وعملية اختبار العيوب (Debugging)
- تعتبر الأخطاء المنطقية من أسهل أنواع الأخطاء التي يمكن اكتشافها .

3. رتب المفردات التالية ترتيبا تصاعديا بحيث تمثل صحيحا لمرحلة التصميم:

- قائمة القرارات.
- خريطة التدفق للنظام
- وثيقة توصيف المتطلبات
- وثيقة توصيف التصميم.

## الخلاصة

عزيزي الدارس ، في نهاية هذه الوحدة دعنا نستعرض ما ورد فيها من مفاهيم:  
أولاً مرحلة تحليل المشكلة وهي مرحلة توصيف دقيق للمشروع البرمجي وتبينة لمرحلة التصميم النهائي ونتاج هذه المرحلة يتمثل في كتابة "وثيقة توصيف المتطلبات".  
ثانياً مرحلة التطوير وتشمل مرحلتي التصميم والتتنفيذ حيث تعد وثيقة توصيف متطلبات العميل هي معطيات هاتين المرحلتين.

مرحلة الاختبار وتشخيص الأخطاء وهي مرحلة مستمرة خلال جميع مراحل تطوير المشروع البرمجي وتشمل خطوة التحقق وهنا يحدد ما إذا كان البرنامج قد تم بناؤه بالشكل الذي يحقق جميع المتطلبات الموضوعة في "وثيقة توصيف المتطلبات" ، الخطوة الثانية المصادقة أي تقويم البرنامج للتأكد من أنه قد تم تصميمه بالطريقة التي يتوقعها ويرضى بها العميل.

مرحلة الصيانة والارتقاء أهم المراحل في عمر المنتج وتشمل معالجة الأخطاء التي قد تنشأ مع التشغيل و إضافة عمل وظيفي جديد للبرنامج و إضافة تقارير خروج للبرنامج و تهيئه البرنامج للعمل على أنظمة تشغيل جديدة ونشاطات المظلة وهي مجموعة أنشطة مستقلة عن أي نشاط برمجي هيكلية وتجري طوال عملية بناء المنتج البرمجي و من أهمها إعداد الوثائق وإنتاجها و متابعة المشروع البرمجي ومراقبته و ضمان جودة المنتج البرمجي و إدارة تكوين البرنامج و إدارة المخاطر .

## لمحة مسبقة عن الوحدة التالية

عزيزي الدارس ،

في الوحدة التالية سنتناول موضوع نماذج عمليات تطوير البرمجيات والتي سوف نتناول فيها أهم هذه النماذج بطريقة تساعده على عمل مقارنة بينها ، ومن ثم اختيار المناسب منها بناءً على نوعية متطلبات المشروع البرمجي المطلوب تطويره ، وكذلك القيود المفروضة عليه. نرجو أن تكون وحدة مفيدة لك.