بسم الله الرحمن الرحيم

مقرر هندسة البرمجيات
١٤٩٤

الوحدة الثامنة

صيانة البرمجيات وتوثيقها
Software Maintenance and Documentation

إعداد: م. هناء قشطة

الفصل الدراسي الأول

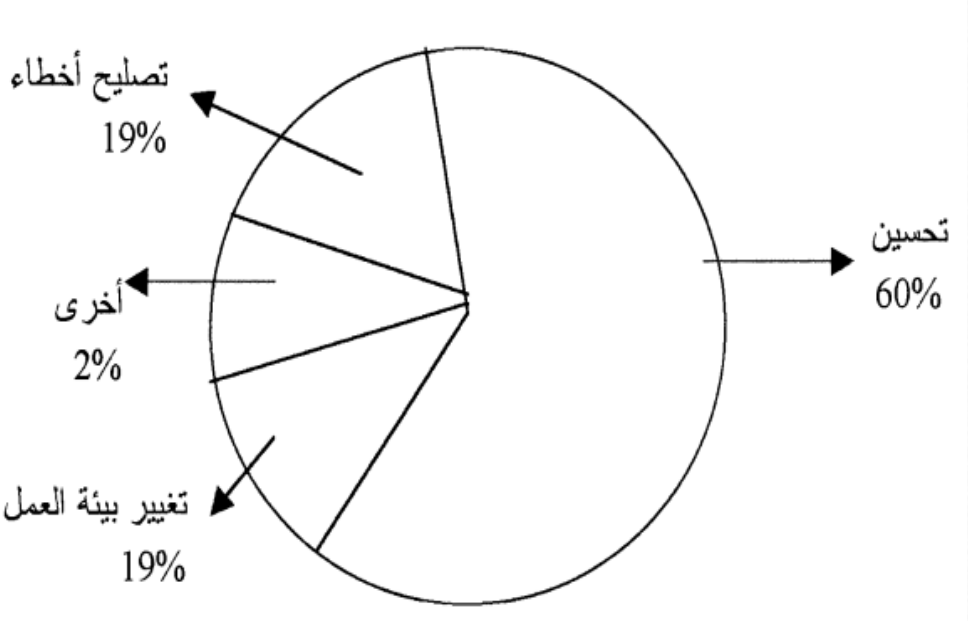
٢٠٢١م-٢٠٢٢م

أهداف اللقاء

- تحديث البرمجيات بصيانتها.
- تقدير تكلفة صيانة البرمجيات وتحديثها.
- تحليل وشرح العوامل التي تؤثر في صيانة البرمجيات.
- تصنيف طرق توثيق البرمجيات.
- توضيح أهمية قاموس البيانات في عملية الصيانة.
- تعداد واستخدام قاموس البيانات.

٢. صيانة البرمجيات

- **يقصد بصيانة البرمجيات:** الجهد المبذول لإحداث تغيير على المنتج بعد التصنيع، لتصليح الأخطاء، أو لتغيير بيئة العمل، أو لأمور أخرى طارئة.

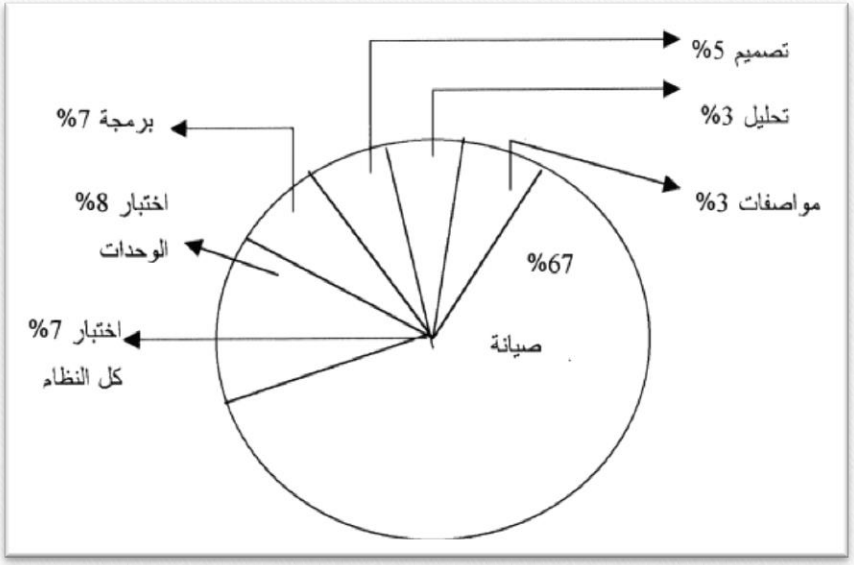


الوقت المصروف على مراحل الصيانة

٢. صيانة البرمجيات

أهمية صيانة البرمجيات:

- تعتبر صيانة البرمجيات ذات أهمية قصوى في هندسة البرمجيات، و بالتالي تعتبر سهولة الصيانة أحد المتطلبات الرئيسية لهندسة البرمجيات، لإنتاج برمجيات تلبي حاجة المستخدم، و ذات جودة عالية، و بتكلفة معقولة.
- عمليًا تواجه الشركات مشكلات عديدة في هذا المجال، مما يكبدها تكاليف عالية، حيث ٦٧% من الجهد و الوقت و التكلفة المادية و الموارد تنفقها الشركات في صيانة البرمجيات الموجودة .



(التكلفة النسبية لمراحل تطوير البرمجيات).

٢. صيانة البرمجيات

- يتوقع أن تزداد هذه النسبة في المستقبل، مما سيجعل الشركات غير قادرة على تخصيص الموارد اللازمة لإنتاج برمجيات جديدة.
- لماذا تتطلب البرمجيات صيانات كثيرة، تستهلك جهدًا و موارد هائلة؟
- ذلك أن معظم البرمجيات الموجودة حاليًا يتراوح عمرها ما بين ١٠ – ١٥ عامًا، و خلال هذه الفترة ظهرت تغيرات في أساليب التصميم و البرمجة، مما يتطلب باستمرار إعادة تصميم هذه البرمجيات و تطويرها في ضوء التقنيات و الأساليب الحديثة.
- أيضًا مما يزيد صيانتها تعقيدًا، أن هذ البرمجيات قد تما تصميمها في الماضي، و أدخلت عليها تغييرات متلاحقة، دون الأخذ بعين الاعتبار لمتطلبات هندسة البرمجيات، كاستخدام التصميم الهيكلي، و التوثيق، و غيرها، يجعل صيانة هذه البرمجيات عملية معقدة تتطلب وقتًا و جهدًا كبيرين.

٢. صيانة البرمجيات

- يختلف مفهوم صيانة البرمجيات عن المفهوم العام للصيانة، و الذي يقصد به إصلاح الأعطال، فصيانة البرمجيات لا تعني إصلاح الأخطاء التي تظهر في البرامج فقط، بل **يشمل مفهوم صيانة البرمجيات أنشطة محددة هي:** الصيانة العلاجية، والصيانة التكييفية، والصيانة التحسينية، والصيانة الوقائية، وهذه الأنشطة تتم عادة بعد إنتاج البرمجيات واختبارها ووضعها قيد التشغيل.

١. **الصيانة العلاجية Corrective Maintenance**

صيانة تشخيص وتصحيح الأخطاء الموجودة في البرمجيات التي تظهر خلال تشغيلها، والتي لم يتم اكتشافها خلال مراحل الاختبار Testing، أثناء تطوير النظام.

٢. **الصيانة التكييفية Adaptive Maintenance**

صيانة يتم خلالها إدخال تعديلات على البرمجيات بهدف مواءمتها مع التطورات التي تحدث في مجالات الحوسبة، و مواءمتها مع متطلبات التشغيل ضمن بيئات جديدة، و باستخدام تقنيات و أساليب جديدة. (مثلاً ظهور: أجيال جديدة من الأجهزة، أنظمة تشغيل مطورة...).

٢. صيانة البرمجيات

٣. الصيانة التحسينية Perfective Maintenance

صيانة يتم خلالها إدخال تعديلات و إضافات إلى البرمجيات التي يتم تشغيلها واستخدامها بشكل ناجح حاليًا، وذلك بهدف إدخال إمكانيات جديدة إلى البرمجيات، وتعزيز قدراتها ووظائفها، لتلبي رغبات جديدة للمستخدمين أو تحسين رغبات موجودة حاليًا.

٤. الصيانة الوقائية Preventive Maintenance

صيانة تهدف إلى تجنب مشكلات قد تحدث في المستقبل، نتيجة التطور المتوقع في التجهيزات، وبيئات التشغيل الجديدة، أو لتحسين المتطلبات في المستقبل، أو لتوفير متطلبات لازمة مستقبلا.

٢. صيانة البرمجيات

خصائص صيانة البرمجيات

- يمكن دراسة خصائص صيانة البرمجيات وفق ثلاث مداخل

وهي:

١. الأنشطة أو العمليات اللازمة لصيانة البرمجيات، وأثر أساليب هندسة البرمجيات في كفاءة هذه الأنشطة.

٢. التكاليف اللازمة لصيانة البرمجيات.

٣. العوامل المؤشرة في صيانة البرمجيات.

٢. صيانة البرمجيات

الأنشطة المتعلقة بصيانة البرمجيات

- تبدأ الأنشطة المتعلقة بصيانة البرمجيات فور استلام طلب الصيانة Maintenance Request.

(١) إذا كان البرنامج المصدري هو التوثيق الوحيد المتاح للبرمجيات المطلوب صيانتها، تتم الأنشطة وفق التسلسل التالي:

— تحليل البرنامج المصدري لمحاولة فهمه، والتعرف على هياكل البيانات، وخطوات المعالجة، والوظائف التي يقوم بها.

— تحديد التعديلات والتغييرات التي يجب إدخالها إلى البرنامج، وكتابة الصيغة البرمجية اللازمة لها.

٢. صيانة البرمجيات

الأنشطة المتعلقة بصيانة البرمجيات

- مراجعة البرنامج للتأكد من صحة إدخال التعديلات، ومن عدم وجود أخطاء فيها أو ناتجة عنها.
- هذه الأنشطة تعتمد على التقدير والتجربة والخطأ، وتسمى الصيانة غير المنهجية أو غير الهيكلية Unstructured Maintenance.
- هذا النوع من الصيانة يؤدي إلى إضاعة الجهود والإحباط نتيجة عدم اتباع منهجية واضحة في التطوير.

٢. صيانة البرمجيات

الأنشطة المتعلقة بصيانة البرمجيات

(٢) في حالة توفر التوثيق الكامل للبرنامج المطلوب صيانته (حيث أنه تم تطويره وفق منهجية محددة بشكل واضح)، فإن الصيانة تتم وفق التسلسل التالي:

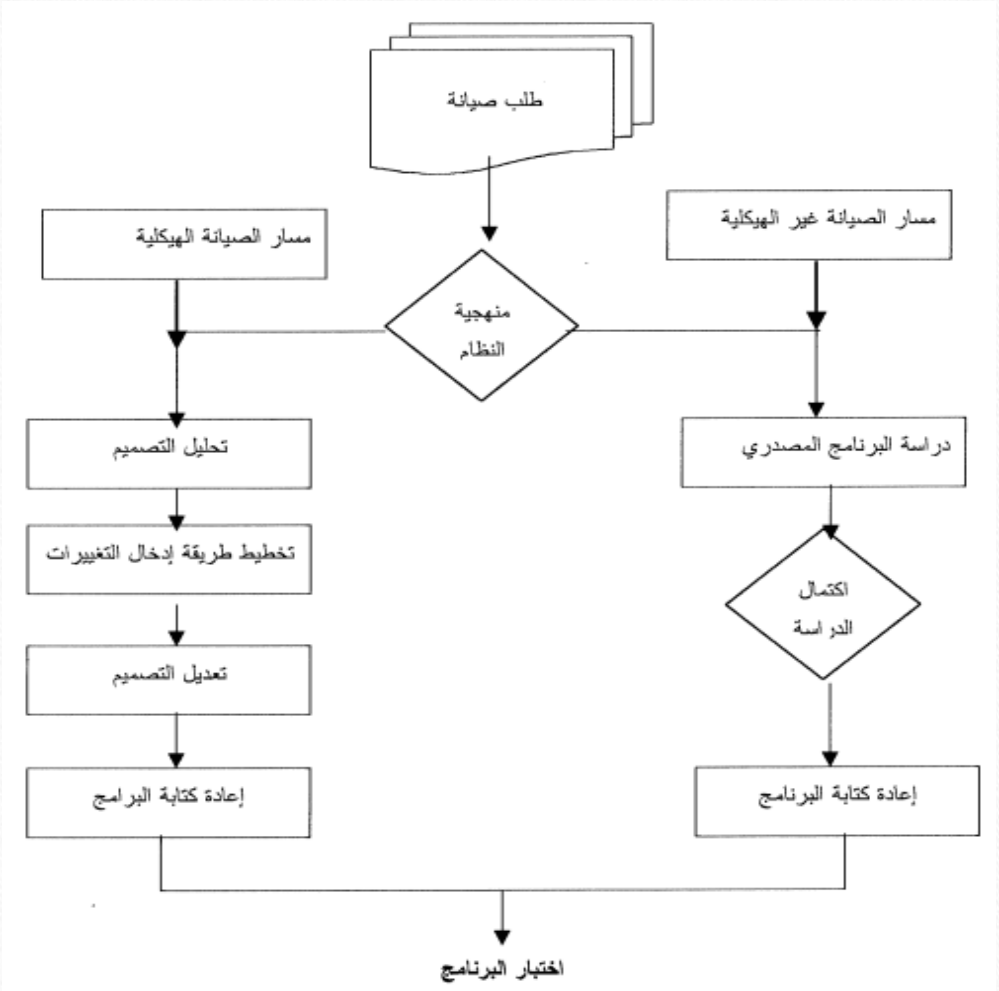
١. تقييم وثائق التصميم لتحديد خصائص النظام من حيث هيكليته وأداؤه وواجهة استخدامه.
٢. تحديد تأثير التعديلات اللازمة، ووضع خطة لأسلوب إدخال هذه التغيرات.
٣. تعديل التصميم الحالي باستخدام أساليب هندسة البرمجيات، والمتعلقة بتصميم البرمجيات، ثم مراجعة التصميم المعدل، قبل البدء بعملية البرمجة.
٤. إعادة كتابة البرنامج المصدري للنظام لإدخال التعديلات.
٥. اختبار النظام بعد التعديل.

٢. صيانة البرمجيات

الأنشطة المتعلقة بصيانة البرمجيات

- هذا النوع من الصيانة يسمى الصيانة البنيوية أو الهيكلية Structured Maintenance، حيث يطبق هذا الأسلوب في الأنظمة التي طورت باستخدام منهجية هندسة البرمجيات (ص١٩٨ شكل٣ الصيانة الهيكلية والصيانة غير الهيكلية).
- أخيرًا فإن أساليب هندسة البرمجيات لا تضمن عدم حدوث مشكلات، ولكنها تساعد على تخفيض عدد المشكلات، وحجم المشكلة الطارئة، وتسهيل علاجها، وتسهيل التعديل علي النظام، وتطويره، وتحسين جودته الكلية.

٢. صيانة البرمجيات



الصيانة الهيكلية و الصيانة غير الهيكلية

٢. صيانة البرمجيات

مهام صيانة البرمجيات

تبدأ مهام صيانة البرمجيات قبل فترة طويلة من إعداد طلب الصيانة:

- ففي البداية تقوم شركات صناعة البرمجيات بتأسيس الهيكل التنظيمي اللازم لعمليات الصيانة.
- ثم تصميم الإجراءات اللازمة لتقييم الصيانة البرمجيات وإعداد التقارير.
- تحديد تسلسل الأحداث الذي تمر به طلبات الصيانة.
- تحديد مسك (تسجيل أو حفظ) السجلات المتعلقة بأنشطة الصيانة، تحديد المعايير اللازمة لمراجعة أنشطة الصيانة وتقييمها.
- **مهام الصيانة تتم كما يلي:**

٢. صيانة البرمجيات

١. تنظيم أنشطة الصيانة:

تختلف الهياكل التنظيمية بشكل واسع في شركات تطوير البرمجيات، بالتالي لا يوجد هيكل محدد، فلكل حالة طبيعتها الخاصة واحتياجاتها.

لا يتم أداء أنشطة الصيانة من قبل وحدة تنظيمية متخصصة، بل من قبل أقسام التحليل و البرمجة بشكل تكليفات خاصة طبقًا للعمل، فمن الضروري تخصيص مسؤوليات الصيانة لطرف ما حتى لو بشكل غير رسمي.

في الشركات الصغيرة يتم إرسال طلبات الصيانة إلى:

- مراقب الصيانة يحولها إلى مدير النظام.
- مدير النظام، الذي يقوم بدراستها وتحديد اللازم، ثم يعرضها على لجنة خاصة تسمى لجنة رقابة التغييرات Control Authority Change.

٢. صيانة البرمجيات

- لجنة مراقبة التغييرات هي لجنة فنية مهمتها دراسة التغييرات المقترحة للبرمجيات الحالية، والموافقة، أو رفضها. (تضمن لجنة مراقبة التغييرات عدم تغيير النظام لصالح جهة معينة مما يؤثر سلبًا على المستخدمين الآخرين).
- ثم يقوم فريق الصيانة بتنفيذ التغييرات المطلوبة.

هذا التنظيم لعملية الصيانة يساعد على تحديد المسؤوليات وتحسين سير أنشطة الصيانة.

٢. صيانة البرمجيات

٢. إعداد التقارير Reporting:

- طلبات الصيانة يجب إعدادها جميعًا بشكل قياسي وفق نموذج موحد يسمى نموذج طلب الصيانة Maintenance Request Form أو يسمى تقرير المشكلة Problem Report.
- يقوم المستخدم بإعداد التقرير.
- عندما يريد إجراء تعديلات.
- في حالة اكتشاف خطاء معين، حيث يجب على المستخدم إعداد وصف كامل للحالات التي تؤدي إلى هذا الخطأ (يشمل البيانات المدخلة و القوائم و أي مواد تصف المشكلة).
- بالنسبة للتقارير المتعلقة بالصيانة التكييفية أو التحسينية فيجب أن ترفق بوصف مختصر للمتطلبات المطلوبة.
- نموذج طلب الصيانة هو وثيقة خارجية يتم استخدامها أساسًا لتخطيط عمليات الصيانة.

٢. صيانة البرمجيات

إعداد التقارير Reporting:

- النماذج التي يتم إعدادها من قبل إدارة تطوير البرمجيات فتمثل الوثيقة الداخلية، فهي تقرير المتغيرات Software Change Report الذي يجب أن يتضمن:
  - الجهد اللازم لتلبية طلب الصيانة.
  - طبيعة التغييرات اللازمة.
  - أولوية الطلب Priority of the Request.
  - أي بيانات تتعلق بالتغييرات المطلوبة.
- يرسل هذا التقرير إلى لجنة مراقبة التغييرات للموافقة على التغييرات قبل البدء بالتخطيط لتنفيذها.

٢. صيانة البرمجيات

٣. تسلسل احداث الصيانة Flow of Maintenance Events:

- عند استلام طلب صيانة في البداية يتم تحديد نوع الصيانة (تصحيحية، تكييفية، تحسينية، وقائية).
- في حالة الصيانة التصحيحية يتم تقييم أثر الأخطاء الموجودة:
- **شديد التأثير:** يتم تخصيص فريق من المبرمجين تحت إشراف مدير النظام لحل المشكلة فورًا.
- **غير شديد التأثير:** تقييم الطلب و تصنيفه ضمن فئات، ثم جدولته حسب أهميته و توفر الموارد اللازمة لتنفيذه.
- في بعض الحالات تكون الأخطاء ذات تأثير شديد جدًا، بحيث تتطلب إجراءات فورية، فيتم التعديل على البرنامج المصدري، دون الاهتمام بما ينتج من آثار جانبية، و دون تحديث وثائق النظام، (لا ينصح بهذا الأسلوب إلا في الكوارث) و يجب توثيق التغييرات التي تمت بعد الانتهاء من معالجة هذا النوع من المشكلات.

٢. صيانة البرمجيات

٣. تسلسل احداث الصيانة Flow of Maintenance Events:

- في حالة الصيانة التكييفية أو التحسينية:
  - يتم تقييم التكييفات و التحسينات المطلوبة، و تصنيفها حسب أولويتها قبل وضعها قيد التنفيذ.
  - ليس بالضرورة الموافقة على جميع الطلبات التحسينية، لعوامل منها:

    استراتيجية الشركة – الموارد المتاحة – دورة حياة النظام – غيرها.
- جميع أنواع الصيانة يتطلب تنفيذها نفس المهام:
  - إجراء التعديلات في تصميم البرمجيات.
  - عمليات المراجعة.
  - إدخال التعديلات اللازمة إلى البرنامج المصدري.
  - إجراء اختبارات الوحدات الوظيفية – الاختبارات التكاملية – غيرها.

٢. صيانة البرمجيات

٣. تسلسل احداث الصيانة Flow of Maintenance Events:

- العمليات الفنية لصيانة البرمجيات هي عبارة عن إعادة و تكرار للعمليات الفنية الخاصة بتطوير البرمجيات.
- المهمة الأخيرة هي المراجعة النهائية للتأكد من ان البرمجيات أصبحت تلبي فعلاً المتطلبات المذكورة في طلب الصيانة.
- بعد الانتهاء من إنجاز جميع المهام يتم إجراء مراجعة الحالة Situation Review التي تهدف للإجابة على الأسئلة التالية: ص٢٠٢.

- تعتبر هذه المراجعة مفيدة لتخطيط المستقبلي لعمليات الصيانة، لأنها تمثل تغذية راجعة ضرورية للإدارة القائمة بهذه العمليات.

٢. صيانة البرمجيات

٤. مسك سجلات الصيانة Maintenance Record Keeping:

- تسجيل بيانات أحداث الصيانة يوفر قاعدة بيانات ضرورية تساعد إدارة الصيانة على تقييم فعالية أساليب الصيانة المستخدمة، و جودة البرمجيات، و تساعد على تحديد التكاليف الفعلية للصيانة.
- أهم عناصر البيانات التي يجب تسجيلها عند توثيق عمليات الصيانة: ص٢٠٣.
- فيتم تجميع هذه البيانات لكل عملية من عمليات الصيانة، و تخزن في قاعدة بيانات ليجري تحليلها و تقييم أداء الأنشطة.
- يمكن قياس الجهد المبذول في الصيانة بطريقة أخرى، حيث تدل الدراسات أن درجة تعقيد البرنامج و نمط البرمجة هما أكثر العوامل تأثيرًا في حدوث الصيانة التصحيحية.

٢. صيانة البرمجيات

٥. التقييم Evaluation:

- تعتمد دقة التقييم على توفر البيانات المتعلقة بعملية الصيانة، حيث يمكن استخدام عدد من المقاييس للتعرف على مستوى أداء عمليات الصيانة، ومن هذه المقاييس: ص٢٠٤.
- توفر هذه المقاييس أساسًا كميًا لتقييم أداء عمليات صيانة البرمجيات في المؤسسة.
- تساعد على اتخاذ القرارات المتعلقة باختيار أساليب التطوير و لغات البرمجة.
- تساعد على التنبؤ بحجم أعمال الصيانة في المستقبل، و تخصيص الموارد اللازمة لذلك.

٢. صيانة البرمجيات

المشكلات المتعلقة بصيانة البرمجيات

- معظم المشكلات المتعلقة بصيانة البرمجيات تعود إلى قصور، أو ضعف الطرق المستخدمة في تخطيط و تطوير هذه البرمجيات، وعدم تطبيق أساليب هندسة البرمجيات كالرقابة و المراجعة.
- هناك أمثلة نموذجية لهذه المشكلات: ص٢٠٥.

٣. كلفة صيانة البرمجيات

- ازدادت تكاليف صيانة البرمجيات في السنوات الماضية، حيث في:
  - السبعينات حوالي ٣٥% من ميزانية البرمجيات.
  - الثمانينات حوالي ٦٠% من ميزانية البرمجيات.
  - السنوات القادمة متوقع أن تصل ٨٠% من ميزانية البرمجيات.
- تقاس تكاليف الصيانة بالمبالغ المالية اللازمة لها، وبالإضافة للمقياس المالي توجد مقاييس أخرى مهم، مثل: ص٢٠٦.

٣. كلفة صيانة البرمجيات

- **يقسم الجهد المبذول في البرمجيات إلى فئتين أو نوعين من الأنشطة:**
- **أنشطة منتجة:** تتضمن عمليات أساسية مثل التحليل، و التقييم، و التصميم، و التعديل، و البرمجة.
- **أنشطة غير منتجة:** تتضمن الوقت اللازم لفهم البرنامج المصدري، وتفسير هياكل البيانات المستخدمة فيه، والتعرف على خصائص واجهات الاستخدام، وحدود الأداء، وغيرها.
- يمكن تقدير تكاليف الأنشطة المنتجة باستخدام نفس الطرق المتبعة في هندسة البرمجيات، لتحديد تكلفة عمليات التطوير المختلفة (كالتحليل، والتصميم، وغيرها...).
- أما الأنشطة غير المنتجة الخاصة بعمليات الصيانة، فيصعب تقديرها، وهي ترتبط أساسًا بمدى تطبيق منهجية هندسة البرمجيات أثناء تطوير البرمجيات، حيث كلما كان مستوى تطبيق منهجية هندسة البرمجيات عاليًا، كلما كان الوقت و التكاليف المبذولة في هذه الأنشطة قليلة.

٤ـ العوامل المؤثرة في صيانة البرمجيات

- أحد أهداف هندسة البرمجيات هو إمكانية الصيانة **وهي:** سهولة فهم و تصحيح و تكييف و تحسين (تعزيز) البرمجيات.
- تتأثر صيانة البرمجيات بالعديد من العوامل أهمها: ص٢٠٧.
- تعتبر إمكانية الصيانة من المفاهيم التي يصعب قياسها كميًا و بشكل مباشر، لذلك يتم اتباع القياس غير المباشر لها، من خلال السمات المختلة لأنشطة الصيانة التي يمكن قياسها.
- من مقاييس الصيانة Maintainability Metrics، قياس الجهد المبذول في صيانة البرمجيات و يسمى Maintainability Index MI، حيث يستخدم MI لقياس الصيانة المطلوبة للبرنامج وفق معادلة رياضية، تعتمد على مقاييس التعقيد، حيث كلما زاد التعقيد زادت الصيانة.
- معادلة MI و شرح عناصرها: ص٢٠٨

$$MI = 171 - 5.2 * Ln(ave\ V) - 0.23 * ave\ V(G) - 16.2 * Ln(ave\ loc) + 50 * \sin(sqrt(2.4 * per_{2}CM))$$

٤. العوامل المؤثرة في صيانة البرمجيات

- العوامل المؤثرة في صيانة البرمجيات يجب التحقق منها خلال عمليات المراجعة التي يجب إجراؤها في نهاية كل خطوة من خطوات تطوير البرمجيات، حيث يتم التأكد من تطبيق المتطلبات والمعايير التي تضمن التوصل إلى برمجيات قابلة للصيانة في مراحل التطوير كالتحليل والتصميم.
- تعتبر المراجعة النهائية للبرمجيات (تسمى Configuration Review) في نهاية مرحلة الاختبار الأكثر شمولية، حيث تهدف إلى التأكد من اكتمال، ووضوح و سهولة فهم جميع العناصر المكونة للبرمجيات.

٥. توثيق البرمجيات Software Documentation

- **توثيق البرمجيات:** هو تسجيل جميع المعلومات المتعلقة بتطوير البرمجيات، وتتضمن هذه المعلومات النماذج Models المستخدمة خلال مرحلة التحليل والتصميم، والتوصيف التفصيلي لمكونات هذه النماذج.
- يجب أن تخزن وتحفظ هذه الوثائق بطريقة منتظمة تسهل على كل من المستخدم والمطور وكادر الصيانة الرجوع إليها عند الحاجة.
- يطلق على مجموعة من الوثائق المتعلقة بالبرمجية والتي تعد خلال مراحل تطويرها المختلفة دليل البرمجية او دليل النظام System Directory.
- يمكن أن يتضمن دليل النظام عدة إصدارات للوثيقة الواحدة، تعكس التغييرات المختلفة التي تمت خلال حياة النظام.

٥. توثيق البرمجيات Software Documentation

مكونات دليل النظام

- مكونات توثيق النظام، أو الوثائق المختلفة التي توجد في دليل النظام، ويطلق أيضًا على هذه الوثائق مصطلح فقرات تكوين البرمجية، وهي كالتالي:
  - توصيف النظام.
  - توصيف متطلبات البرمجيات.
  - توصيف خصائص تصميم البرمجيات.
  - البرامج المصدرية للنظام.
  - وثائق اختبار النظام.
  - أدلة تركيب وتشغيل البرمجية.
  - البرامج التنفيذية للنظام.
  - توصيف قاعدة البيانات للنظام.
  - وثائق صيانة النظام.
  - المعايير والإجراءات المستخدمة لتطبيق أساليب هندسة البرمجيات ومنهجيتها.

٥. توثيق البرمجيات Software Documentation

- إن القيام بصيانة النظام يتطلب إدخال التعديلات على البرامج المصدرية، وأيضًا إجراء تغييرات في التصميم، وإجراء اختبارات جديدة.
- بالتالي يجب تحديث وثائق دليل النظام نتيجة لعمليات الصيانة، لأهمية المحافظة على دليل النظام محدثًا Up To Date.

٥. توثيق البرمجيات Software Documentation

إدارة مكونات البرمجيات

- تهدف إدارة مكونات البرمجيات إلى ضبط التغييرات التي يمكن أن تحدث خلال مراحل هندسة البرمجيات، وبالتالي يتم تطبيقها خلال جميع المراحل، ويمكن أن يطلق عليها إدارة تغيير البرمجيات.
- بشكل عام تهدف إدارة مكونات البرمجيات إلى:
  - تحديد التغييرات التي يجب ادخالها في البرمجية.
  - تخطيط التغييرات لتحديد الطريقة والأسلوب الافضل لذلك.
  - التأكد من اجراء التغييرات بصورة صحيحة.
  - اعلام الجهة المستفيدة من البرمجية بالتغييرات التي تمت فيها.

٥. توثيق البرمجيات Software Documentation

مهام إدارة مكونات البرمجيات

- تهدف إدارة مكونات البرمجيات إلى الرقابة على التغييرات التي يمكن أن تتم على البرمجيات، وتتضمن إدارة المكونات القيام بالوظائف أو المهام **التالية**:
  - تحديد وتعريف عناصر أو فقرات المكونات.
  - الرقابة على الإصدارات المختلفة لهذه العناصر.
  - الرقابة على التغييرات التي يمكن أن تحدث خلال دورة حياة البرمجيات.
  - مراجعة المكونات للتأكد من إعدادها بشكل صحيح وفق الأهداف المرجوة منها.
  - إعداد التقارير اللازمة لتعريف ذوي العلاقة بالتغييرات التي تمت على هذه البرمجيات.

٥. توثيق البرمجيات Software Documentation

١. تحديد و تعريف مكونات البرمجيات

تتطلب إدارة مكونات البرمجيات و الرقابة على التغييرات التي تتم فيها:

- تحديد و تعريف كل عنصر أو فقرة من هذه المكونات.
- إعطائها التسميات المناسبة، و تعريف الخصائص أو السمات الخاصة بكل منها، مثل اسم الوثيقة و وصف محتواها، و غيرها.
- تنظيمها ضمن مجموعتين:

- **مجموعة المكونات الأساسية**

يتم إنشاء المكونات الأساسية غالبًا من قبل مهندس البرمجيات خلال مراحل تطوير النظام (تحليل، تصميم، برمجة، اختبار)، بالتالي فإن وثائق مثل توصيف المتطلبات، البرنامج المصدري للوحدة الوظيفية، و غيرها، يمكن اعتبارها مكونات أساسية.

- **مجموعة المكونات المركبة**

تتكون عادة من مجموعة من المكونات الأساسية كمخططات التدفق، و توصيف قاعدة البيانات، و غيرها.

٥. توثيق البرمجيات Software Documentation

٢. رقابة الإصدارات Version Control

- تتضمن هذه المهمة مجموعة من الإجراءات والأدوات المستخدمة لإدارة ومتابعة الإصدارات المختلفة لعناصر مكونات النظام.
- يتم تمثيل الإصدارات المختلفة للبرمجيات بإعطائها ما يسمى برقم الإصدار.
- كل إصدار هو عبارة عن مجموعة كاملة من المكونات (البرنامج المصدري، هياكل البيانات، ...)، ويمكن أن يكون للإصدار الواحد تنوعات مختلفة، حيث مثلاً يمكن ان يكون هناك للإصدار الواحد نوعان يستخدمان في بيئات تشغيل مختلفة.

٥. توثيق البرمجيات Software Documentation

٣. رقابة التغيير Change Control

تتضمن هذه المهمة مجموعة من الإجراءات والأدوات التي تساعد على ضبط التغييرات التي تحدث في البرمجيات، سواء خلال تطويرها أو بعد ذلك في أثناء صيانتها.(شكل ٦ ص ٢١٣)

٤. مراجعة مكونات النظام System Configuration Audit

تهدف هذه المراجعة الى التأكد من أن التغييرات التي أجريت في البرمجة فد نفذت بصورة صحيحة، وخلال هذه المراجعة يجري تقييم جميع عناصر المكونات..

٥. اصدار التقارير Reporting

تهدف هذه المهمة في إدارة المكونات الى تزويد مستخدمي النظام وجميع الجهات ذات العلاقة به، بالمعلومات المتعلقة بالتغييرات التي تمت أو التي يجري إدخالها، أو التي ستتم في المستقبل القريب على هذا النظام، وذلك لكي يكونوا على اطلاع دائم بذلك.

٦. قاموس البيانات Data Dictionary

- القاموس هو قائمة منتظمة تضم جميع عناصر البيانات الخاصة بالنظام وتوفر تعريفات دقيقة لها بحيث يمكن لكل من مطور النظام ومستخدميه تكوين فهم مشترك لمدخلات النظام ومخرجاته ومكوناته ومخازنه.
- يساعد قاموس البيانات على تنميط جميع التسميات المستخدمة في النظام وتوفير توصيفات دقيقة وواضحة لها ولذلك فهو يمثل مرجعاً رئيسياً من مراجع النظام.

٦. قاموس البيانات Data Dictionary

- يتم توصيف هياكل البيانات في القاموس من خلال الرموز الموضحة في الجدول التالي:

الرمز	المعنى
=	يكافئ
+	و
ملاحظة	تكتب الملاحظات بين نجمتين
(مكون اختياري)	تكتب المكونات الاختيارية بين قوسين دائريين
{ مكون متعدد القيم } N, 1	تكتب المكونات متعددة القيم ضمن قوسين كبيرين
[مكون 1 / مكون 2]	يستخدم فقط مكون واحد من بين المكونات المكتوبة بين أقواس مربعة

٦. قاموس البيانات Data Dictionary

INVOICE= INVOICE-NO + { CUSTOMER-NAME / CUSTOMER-PHONE }

+ INVOICE- AMOUNT + (SHIPPING-DATE)

10

1{ + INVOICE _ LINE }

* يجب إصدار ثلاث نسخ من الفاتورة *

INVOICE-LINE = LINE-NO + ITEM NAME + QUANTITY + PRICE +VALUE

٦. قاموس البيانات Data Dictionary

- يتبين لنا من المثال السابق أن تدفق البيانات المسمى فاتورة يحتوي على خمس مكونات: الأول هو رقم الفاتورة والثاني اسم الزبون أو رقم هاتفه ونلاحظ أن أي منهما يكفي ولا ضرورة للاحتفاظ بكليهما والعنصر الثالث هو قيمة الفاتورة والعنصر الرابع اختياري يمكن عدم كتابته أما العنصر الخامس فهو تفاصيل اسطر الفاتورة ويمكن وصف المكون INVOICE – LINE ، ويمكن أن يكون على النحو التالي:

INVOICE – LINE = LINE –NO + ITEM NAME + QUANTITY + PRICE + VALUE

Questions or Comments?