مقدمة في هندسة البرمجيات

محمد الأحمد

فهرس المحاضرة

01 What is SE?
02 SE Ethics

تعريف البرمجيات "Software"

هي مجموعة برامج (programs) ووثائق مساعدة (documentation) وأحياناً بحاجة لمعلومات (data).

تعتبر الوثائق جزء أساسي من أي برنامج وهي عبارة عن تسجيل كل خطوة من خطوات التي نقوم بها خلال مرحلة التصميم والتكويد والاختبار ضمن وثائق وذلك لجعل البرنامج قابلية للتطوير ومعالجة الأخطاء.

تعريف هندسة البرمجيات "Software Engineering"

فرع من فروع العلم والتي تهتم بتطوير العلم بطريقة هندسية. وهو علم تطبيقي هدفه الأساسي تطوير نظم برمجية لتقدم خدمات. تختلف كل برمجية بمجموعة الخدمات التي تقدمها ولا يتم تطوير هذه البرمجيات عشوائياً إنما يوجد خطوات تتم ضمن عملية معينة، هذا العلم يخبرنا كيف نطور هذه النظم بطريقة نظامية يكون فيها الأخطاء قليلة والكلفة مقبولة والجودة عالية ضمن قيود زمنية لوجود مشكلة في الوصول للسوق. تختلف هندسة البرمجيات عن بقية الهندسات بكون انتاجها يعتمد على منتجات عقلية وليست فيزيائية التي لا تفنى بوجود الزمن ولكنها منتجات منطقية نتاج للعقل البشري والعقل البشري قاصر لذا يوجد أخطاء دوماً.

ملاحظة: لا يمكن تطوير نظام برمجي بدون أخطاء.

أقسام المنتجات البرمجية

منتجات عامة (Generic Products)

عندما تقوم شركة برمجية بإصدار منتجات برمجية وبيعها أي أنها ليست مخصصة لجهة معينة والمطور هو من يحدد الخدمات المطلوبة من هذه المنتجات البرمجية، مثال: البرمجيات التي تنتجها شركة **Microsoft**.

منتجات خاصة (Customized Products)

مخصصة لزبون معين لتلبية احتياجاته الخاصة، مثال: برنامج لشركة نقل.

الفرق بينهما أن التوصيف (وهو المطلوب من المنتجات البرمجية مكتوب ضمن وثائق) بحالة المنتجات العامة يكون فيه المطورين هم المسؤولون عن وضع التوصيف أما المنتجات الخاصة فالزبون هو من يضع التوصيف.

أهمية صناعة البرمجيات

تعد صناعة البرمجيات صناعة ناجحة لا تحتاج لرؤوس أموال أو معدات وكل ما تحتاجه هو عقول مفكرة وتعود بأرباح عالية، مثال: دولة الهند.

كيفية ظهور مصطلح هندسة البرمجيات

في عام 1946 ظهر أول حاسوب، وفي عام 1968 نتيجة أزمة البرمجيات "كثرة الأخطاء وارتفاع كلفة إصلاحها" عقدت اللجنة العلمية للناتو مؤتمر علمي وضعت فيه أسس هندسة البرمجيات وبعدها أخذ بالتطور على مر السنين.

وبالتالي هو علم حديث حيث يوفر أفضل الأدوات والطرق والوسائل والاستراتيجيات اللازم استخدامها لتطوير نظام برمجي ضمن قيود الكلفة والوقت والجودة.

كلفة البرمجيات

قبل 2000 كانت كلفة العتاد الصلب غالية جداً ثم أخذت أسعارها بالتناقص أما البرمجيات فكانت أسعارها منخفضة ثم بدأت تتزايد.

Cycle of Software Developing

1) التخصيص (Specification): توصيف المتطلبات.
2) التصميم (Design)
3) التوكيد (Coding)
4) الاختبار (Testing)

بعد الانتهاء من هذه المراحل يصبح المنتج البرمجي جاهز ليدخل في مرحلة التشغيل (Maintenance) وهي أطول مرحلة تشكل مرحلة الصيانة وتعد أكثر مرحلة مكلفة إذا لم نكن قد اتبعنا طرق هندسة البرمجيات الجيدة أي يكون لدينا الكثير من الأخطاء فنحتاج لتكلفة عالية لإصلاحها، هندسة البرمجيات تؤكد وجوب تطوير النظام وفق هذه المراحل لتفادي وجود الكثير من الأخطاء لكن لابد من وجود بعض الأخطاء لأنه لا يوجد نظام برمجي كامل مهما كانت خبرة الشركة المطورة كما تقوم بعض الشركات بتجاهل الأخطاء لأغراض تسويقية لإجبار الزبون على شراء النسخة الأحدث.

الصعوبات التي نواجهها في هندسة البرمجيات

- ✓ التغاير أو ما يسمى عدم التجانس ووجود العديد من التقنيات والتي يجب تعلمها لتطوير نظام برمجي يعمل على عدة منصات (IOS, Android, ...).
- ✓ النمو السريع في هذا المجال: احتمالية ظهور تقنيات جديدة في كل لحظة.
- ✓ نتيجة اعتماد المؤسسات المتزايد على البرمجيات مثل البنوك والطائرات فيجب أن تكون هذه الأنظمة موثوقة لدخولها في مجالات حساسة، لذلك يوجد ضغط لتطوير منتجات ذات جودة عالية تكون قابلة للاعتماد وموثوقة.
- ✓ مراعاة اختلاف مجالات استخدام البرمجيات فمثلاً عند تطوير الألعاب يجب مراعاة سهولة الاستخدام بعكس الأنظمة البنكية التي يجب أن تكون ذات وثوقية عالية وبالتالي كل مجال يحدد منهجية معينة لتطويره بحسب متطلبات هذا المجال أي لا يوجد منهجية محددة لتطوير نظام وإنما عدة منهجيات يمكن اختيار أحدها أو المزج بينها للوصول للمنهجية المناسبة.

متى نقول عن نظام برمجي أنه جيد؟

- سهولة الصيانة (Maintainability): سهولة تحديد مكان الخطأ والقدرة على إصلاحه وذلك نتيجة تصميم البرنامج بشكل صحيح.
- قابل للاعتماد (Dependability): موثوق وخاصة بالأنظمة الحساسة مثل الطيار الآلي لكون هذه الأنظمة تتحكم بحياة أشخاص لذلك يجب أن يكون الخطأ معدوم قدر الإمكان.
- الكفاءة (Efficient): استهلاك الموارد الحاسوبية أقل ما يمكن أي أن البرنامج لا يحتاج الكثير من الذاكرة أو يستهلك القرص الصلب ....
- سهولة الاستخدام (Usable): لا يوجد حاجة لقراءة تعليمات الاستخدام مثلاً مواقع الويب لا تحوي دليل استخدام لكنها تصمم بطريقة يمكن لأي شخص استخدامه.

Applications Types

1. Stand-alone applications: برامج تعمل على حواسيب محلية (PC) وليست بحاجة لشبكة مثل برامج الرسم وبعض الألعاب.
2. Embedded-Systems: يوجد نظام تحكم والذي يدير عمل العتاد الصلب بالإضافة لواجهة تفاعل مع المستخدم صغيرة مثال: الغسالة.
3. Entertainment Systems: أنظمة شخصية للتسلية مثل وسائل التواصل الاجتماعي والألعاب.

Web Software Engineering

بسبب ظهور الويب أدى إلى تضخم هندسة البرمجيات وظهور علم هندسة الويب "هذا العلم يستعين ب 99٪ من علم هندسة البرمجيات بالإضافة إلى هندسة الشبكات " يعد الويب بيئة تشغيلية فأصبحت أغلب النظم تعمل على الويب أما آخر تطور فهو ظهور Cloud Computing وهو يسمح باستخدام بعض الخدمات عن طريق شرائها من قبل الشركات أي أصبحت البرمجيات تعمل كخدمات (Service).

ظهور الويب في التسعينات أدى إلى تغير البرمجيات من تطبيقات PC ثابتة إلى خدمات موزعة بهدف الوصول الأسرع والتغيير الأسهل.

Reusability:

خلال عملية التطوير يوجد مبدأ هام وهو Reusabilityإعادة الاستخدام يمكن التعبير عنه بـ(لا تخترع الدولاب) أي إعادة استخدام كل ما هو متوفر لأن أي برمجية مستخدمة ومجربة وتم إصلاح مشاكلها فهي بالتالي أكثر نضجاً من البرمجية التي ستكتبها، من الغلط تطوير كل شيء من الصفر بل يمكن الاستفادة من البرمجيات الناضجة وهي بالغالب تكون جزئية بسبب كون أغلب أنظمتنا معقدة ومكونة من عدد من الأنظمة الجزئية.

CASE Tools:

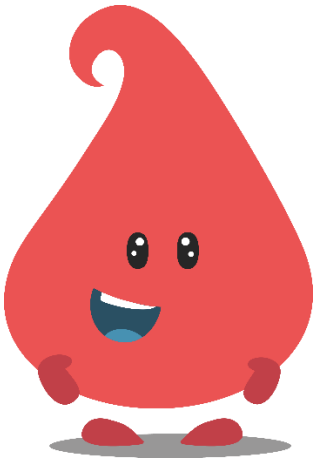
Computer-Aided Software Engineeringالأدوات المساعدة في هندسة البرمجيات وهي نظم برمجية يجب استخدامها من أجل تسريع عملية التطوير مثال: أثناء التطوير يتم رسم مخططات فعند وجود برنامج لرسم المخططات لا حاجة لأن نقوم برسمها يدوياً، عند وجود بيئة تطوير معينة يمكن استخدامها أيضاً كالـ IDE، فقدر الإمكان خلال عملية التطوير يتم الاستفادة من هذه الأدوات ويجب الاستفادة منها لتسريع عملية التطوير لكون الزمن عامل حساس.

أمثلة عن CASE Tools: الكومبيلرات، برنامج Word .....

مراجع المادة:

Software Engineering by Summerville, Ian, 9th Edition or above.

Software Engineering: A Practioner's Approach. Roger S. Pressman.6th edition or Above.

Software Engineering, by Mohammad Ahmad, SVU.

انتهت المحاضرة..