

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

جامعة كـري

كلية علوم الحاسوب وتقانة المعلومات

مدخل هندسة البرمجيات

أستاذ : أبو بكر عبد الرحيم الجاك



محتويات الوحدة

الصفحة	الموضوع
3	مقدمة
3	تمهيد
4	أهداف الوحدة
6	١. ماهية هندسة البرمجيات
8	٢. المهام النموذجية لهندسة البرمجيات
11	٣. مبادئ هندسة البرمجيات
14	٤. أنواع البرمجيات
19	٥. أدوات هندسة البرمجيات بمساعدة الحاسوب

20	1.5 ميزات أدوات هندسة البرمجيات بمساعدة الحاسوب
21	2.5 قصور أدوات هندسة البرمجيات بمساعدة الحاسوب
22	3.5 التصنيف العام لأدوات هندسة البرمجيات المساعدة
24	6. التحديات الرئيسية التي تواجه هندسة البرمجيات
25	7. مهنة البرمجة
28	8. المسؤولية المهنية والأخلاقية لمهندسي البرمجيات
33	الخلاصة
34	لمحة مسبقة عن الوحدة التالية
36	إجابات التدريبات
37	مسرد المصطلحات
45	المراجع

مقدمة

تمهيد

عزيزي الدارس ،،

سنتعرف من خلال هذه الوحدة إلى مفاهيم أساسية في هندسة البرمجيات ، والتي تعد الأساس لهذه المادة . ما هي هندسة البرمجيات ، سنقوم بتوضيح شقي المصطلح ، الشق الهندسي والشق البرمجي ، في القسم الثاني من الوحدة نتناول المهام النموذجية لهندسة البرمجيات ، وهي مهام تهتم بجميع أطوار دورة حياة تطوير البرمجيات بدءاً من المراحل الأولى لتحليل النظام حتى صيانته ، القسم الثالث يتناول مجموعة من المبادئ الأساسية التي تتميز بها هندسة البرمجيات، ثم تنتقل بك الوحدة إلى القسم الرابع وهي أنواع البرمجيات وهنا نستطيع أن نميز نوعان رئيسان: برمجيات النظام ، البرمجيات التطبيقية ، في القسم الخامس نتناول أدوات هندسة البرمجيات بمساعدة الحاسب ، وهي طائفة واسعة من الأنواع المختلفة من البرامج التي تستخدم في مساندة الأنشطة الخاصة بالعمليات البرمجية ، القسم السادس يتناول التحديات الرئيسية التي تواجه هندسة البرمجيات في القرن الحادي والعشرين والتي تتمثل في التحدي المتعلق بالموثوق ، تحدي تحديد عدم التجانس ، تحدي المخرجات ، القسم السابع يتناول مهنة البرمجة وفريق العمل الخاص بهذه المهنة المكون من محللو النظم ومصممو النظم ثم المبرمجون ، القسم الثامن

والأخير يعالج موضوع المسؤولية المهنية والأخلاقية لمهندسي البرمجيات.
تجد عزيزي الدارس في ثنايا الوحدة بعض التدريبات التي يساعدك تنفيذها على فهم محتوى المادة هذا
بالإضافة إلى أسئلة التقويم الذاتي التي تهدف إلى ترسيخ الفهم وتعزيزه لديك و تعميقه .

أهداف الوحدة



عزيزي الدارس،

بنهاية دراسة هذه الوحدة ينبغي أن تكون قادراً على أن :

- تعرّف هندسة البرمجيات.
- صف دور هندسة البرمجيات في هندسة وبناء البرمجيات.
- تشرح الفرق بين علم هندسة البرمجيات وعلوم الحاسبات الآلية.
- تصف العلاقة بين هندسة البرمجيات وهندسة النظم..
- تعدد عمليات البرمجيات.
- تعرّف أدوات هندسة البرمجيات بمساعدة الحاسب الآلي.
- تشرح أهمية أدوات هندسة البرمجيات بمساعدة الحاسب الآلي في بناء البرمجيات.
- تسرد المهام النموذجية لهندسة البرمجيات.
- تعدد مبادئ هندسة البرمجيات.
- تسمي أهم أنواع البرمجيات.
- تشرح التحديات الرئيسية التي تواجه هندسة البرمجيات.
- تصف المسؤولية المهنية والأخلاقية لمهندسي البرمجيات والعمل بها.

توطئة

في عصرنا الحالي والذي يُعد عصر المعلوماتية يعتمد الإنسان في معظم أمور حياته سواء في بيته أو في مكان عمله أو في الأماكن العامة على التعامل مع العديد من الأجهزة والمعدات التي تعتمد في عملها على البرمجيات والتي تعتمد بدورها على أنظمة الحاسب الآلي ، لذا أصبح من المهم تصميم هذه الأنظمة وبرامجها بأقل التكاليف من خلال التوزيع السليم للموارد المتاحة والتصميم الجيد لهذه الأنظمة والبرامج بحيث تعمل بكفاءة وموثوقية عالية وبطريقة ذات فعالية لتعلب دورا فعالا في الاقتصاد الوطني

والعالمي ، ومن هنا ظهر مفهوم هندسة البرمجيات.

وقد استخدمت هندسة البرمجيات كمفهوم نظري من حين لآخر في أواخر الخمسينات وبداية الستينات من القرن الماضي، أما الاستخدام الرسمي لها فكان في مؤتمر عُقد من قبل اللجنة العلمية في منظمة حلف شمال الأطلسي عام ١٩٦٨م حول البرمجيات ، وقد أخذ هذا المصطلح في الانتشار منذ ذلك الحين ولاقي اهتماماً متزايداً في نواحي مختلفة.

قبل ظهور هندسة البرمجيات كان تطوير البرمجيات يفتقر إلى الكثير من المعايير المتعارف عليها اليوم مما نتج عنه ذلك فشل العديد من المشاريع البرمجية ، أو زيادة الوقت اللازم لإنجازها ، أو صعوبة في صيانتها. وحتى تلك النظم البرمجية التي تم تصميمها وإنجازها كانت غالباً ما تعاني من نقص في الوثائق.

وظهور العديد من الأخطاء غير المتوقعة إضافة لدعم وثائقي ضعيف وعدم تلبية الشروط والمتطلبات التي صُممت لأجلها بشكل كامل. ولحل هذه المشاكل والصعوبات كان من اللازم إعادة النظر وتقويم جميع الجوانب المتعلقة بتصميم وتطوير البرمجيات مما أدى إلى ظهور هندسة البرمجيات مع بداية عام ١٩٦٨م نتيجة للنقاشات حول ما عرف في ذلك الوقت "بأزمة البرمجيات" حيث تم لأول مرة الإشارة إلى طرق تصميم وتطوير البرامج، وقد ساعد على ذلك . حينها . ظهور الجيل الثالث من أجهزة الحاسب الآلي القوية ، حيث إن قوتها جعلت النظم البرمجية غير الممكنة سابقاً ممكنة ، ونتج عن ذلك أنظمة ضخمة وكبيرة وأُعيد من الأنظمة السابقة ، مما أدى إلى توسع مجالات استخدام الحاسوب في معظم مجالات الحياة.

١. ما هي هندسة البرمجيات؟ (What is Software Engineering?)

يوجد العديد من التعريفات التي تم اقتراحها من العديد من المهتمين بهندسة البرمجيات ويمكن إجمال هذه التعريفات في أن "هندسة البرمجيات" هي : علم يهتم ببناء الأنظمة البرمجية الكبيرة والمعقدة بواسطة فريق من مهندسي البرمجيات باتباع طرق ومبادئ هندسية منظمة وصحيحة واستخدام وسائل وأدوات وتقنيات تجعل من تصميم وتطوير وبناء واختبار هذه النظم عملية منظمة يمكن تكرارها في حدود الإمكانيات المتاحة مادياً وبشرياً وزمنياً ، وذلك من خلال دراسة دورة حياة النظام مع إعطاء أشكالاً متعددة لعمليات تصميمه وتطويره وبناءه واختباره بحيث يكون المنتج البرمجي النهائي على درجة عالية من الجودة والموثوقية وقليل التكاليف بقدر الإمكان مع تسليمه للعميل في الوقت المناسب على أن يعمل بكفاءة على أجهزة الحاسب الآلي المختلفة.

وبصفة عامة ، يهتم علم هندسة البرمجيات بجميع نواحي إنتاج نظم البرمجيات التي تلبي متطلبات

المستخدمين تحت قيود معينة ، وذلك من خلال تطبيق النظريات والمعرفة بأسلوب فعال وبكفاءة موثوقة، وتدخل الطرق المستخدمة في هندسة البرمجيات في جميع أطوار دورة حياة نظم البرمجيات من المراحل الأولى لتحليل النظام حتى صيانته مروراً بعمليات وضع المواصفات ، والتصميم ، والبناء ، والفحص ، والتشغيل. كما يوظف علم هندسة البرمجيات الطرق والأساليب والعمليات والقياسات الهندسية لتحقيق الأهداف وإتمام الفائدة من الأدوات المعدة لإدارة عمليات تطوير البرمجيات ، مع الاهتمام بالجودة والتحكم في النوعية.

وبالتدقيق في مصطلح " هندسة البرمجيات " نجده يحتوي على شقين : **الشق الأول : هندسي** : وفيه يطبق المهندسون النظريات والأساليب والأدوات الملائمة بفعالية للحصول على حلول مثالية للمشاكل التي تواجههم مع التقيد بالقيود التنظيمية والمالية التي تحيط بالمشكلة ، ودائماً ما يحاولون اكتشاف الحلول للمشاكل حتى عندما لا توجد نظريات أو أساليب مطبقة لتدعمهم ، وذلك من خلال تبنيهم أسلوب تصنيفي منظم لعملهم مع التركيز على اختيار الطريقة الأكثر ملائمة لتنفيذ النظام البرمجي طبقاً للمواصفات المطلوبة والطريقة الأكثر ابتكاراً.

الشق الثاني : فهو برمجي : وفيه يتم الاهتمام بجميع **مظاهر إنتاج البرمجيات** ، حيث إن هندسة البرمجيات لا تهتم فقط بالعملية الفنية لتطوير البرمجيات ولكن تهتم أيضاً بالنشاطات الأخرى مثل إدارة مشروع البرمجيات ، وتطوير الأدوات والأساليب والنظريات لدعم إنتاج البرمجيات.

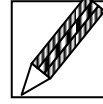
وبناءً على ما سبق يتضح أن **دور هندسة البرمجيات هو : تطبيق منهج مرتب وقابل للقياس لعمليات تطوير وتشغيل وصيانة البرمجيات ، أي تطبيق الهندسة على البرمجيات** ، وهو ذات التعريف الذي وضعه معهد المهندسين الكهربائيين والإلكترونيين (IEEE) لهندسة البرمجيات ، وذلك من خلال الإجابة على العديد من الأسئلة التي تخص هندسة وتطوير البرمجيات ، والتي من أهمها :

- ما هي **الطرق والمبادئ الهندسية المنظمة والصحيحة** التي يمكن تطبيقها على بناء البرنامج؟.
- ما هي **الوسائل والأدوات والتقنيات** التي تجعل من تصميم وتطوير وبناء واختبار البرنامج عملية منظمة يمكن تكرارها في حدود الإمكانيات المتاحة مادياً وبشرياً وزمنياً؟
- كيف **نتمكن اقتصادياً** من بناء برنامج موثوق يعمل بكفاءة ليس على جهاز حاسب واحد بل على العديد من أجهزة الحاسب المختلفة؟.
- ما هي **القياسات والإجراءات** التي يجب إجراؤها لضمان جودة المنتج البرمجي؟.
- كيف سيتم **دعم البرنامج على المدى الطويل** عندما يطلب المستخدم إجراء تعديلات أو تحسينات عليه؟.
- ما هو **المنهج الواجب اتباعه** لدراسة تقنيات الإدارة اللازمة لتخطيط المشاريع البرمجية وتنظيمها واستخدامه كآلية للتحكم في المشروع بشكل يقود إلى تسليم منتج برمجي عالي الجودة قابل

للتشغيل في الوقت المحدد؟.

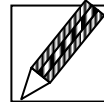
تدريب (١)

ما هو الفرق بين علم هندسة البرمجيات وعلوم الحاسبات الآلية؟



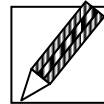
تدريب (٢)

ما هو الفرق بين هندسة البرمجيات وهندسة النظم؟



تدريب (٣)

ما هي عمليات البرمجيات؟



٢. المهام النموذجية لهندسة البرمجيات

(Typical Software Engineering Tasks) :

هناك مجموعة من المهام النموذجية التي تتعلق بهندسة البرمجيات يجب أن تتبع في تطوير

المشاريع البرمجية من قبل فريق متكامل من مهندسي البرمجيات ، وهذه المهام هي كالتالي :

• تحليل المشكلة.	• تحديد المتطلبات.	• تصميم البرمجية.
• الترميز (عملية البرمجة).	• اختبار وتكاملية الشفرة البرمجية.	
• التثبيت وتوزيع البرمجية.	• توثيق البرمجية.	• صيانة البرمجية.
• التدريب	• تقدير الموارد	• إدارة المشروع

وبنظرة ثاقبة للمهام المذكورة أعلاه نجد أن هندسة البرمجيات تهتم بجميع أطوار دورة حياة تطوير

البرمجيات (Software Life Cycle) بدءاً من المراحل الأولى لتحليل النظام حتى صيانتها ، وسوف نتناول

أطوار دورة حياة تطوير البرمجيات بالتفصيل لاحقاً ، ولكن سوف تغطي فكرة بسيطة هنا عن أهم الخطوات

المتبعة (الأطوار) في تطوير البرمجيات من وجهة نظر هندسة البرمجيات :

- مبدئياً لا بد من التعرف على متطلبات النظام فإذا كان هناك زبون أو عدة زبائن فإنه لا بد من حدوث مقابلة بين الزبائن ومتخصصي دراسة وتحليل المتطلبات في فريق التطوير ، وإن لم يكن هناك زبون

محدد ولكن هناك متطلبات لفئة عامة من الزبائن فإن دراسة **متطلبات واحتياجات السوق** يتم عملها وهذه الطريقة مناسبة لتطوير برمجيات عامة.

- بعد دراسة وتحليل المتطلبات والانتها من كتابة وثائق التحليل يتم الانتقال إلى وضع التصميم الأمثل لترجمتها إلى شفرة برمجية قابلة للتنفيذ ، وذلك من خلال **المصممين** في فريق العمل ولا بد أن يكونوا ملمين إلماماً كاملاً **بطرق التصميم المختلفة** والفروق المختلفة بينها ولا بد أن يكون لديهم إلمام بخواص النظم الموجودة مثل أنظمة التشغيل ، ونظم المعلومات المختلفة المتاحة ، وواجهات التطبيق المختلفة الرسومية والبرمجيات المساعدة التي يمكن أن تساعد في عملية الترميز (البرمجة).

- بعد الانتهاء من تخطيط وثائق التصميم يتم الانتقال إلى **توزيع المهام على المبرمجين** في فريق العمل لتحويل تلك المخططات وترجمتها إلى شفرة برمجية قابلة للتنفيذ باستخدام إحدى لغات البرمجة عالية المستوى (High Level Programming Languages) ، ومن ثم تبدأ عملية **التكامل للنظام** وهي عملية تجميع المهام البرمجية بعد ترميزها عن طريق فريق التطوير مع برمجيات جاهزة تخدم في نفس المشروع.

- بعد الانتهاء من مرحلة برمجة وتكامل النظام تبدأ **مرحلة اختبار النظام البرمجي** من حيث الأخطاء البرمجية ، وكذلك من حيث مطابقته للمواصفات المطلوبة وجودة الأداء. ومصطلح جودة البرمجية (Quality Assurance) يتعلق بعملية تطوير المنتج البرمجي والطريقة التي تم تطويره بها بحيث تكون وفقاً لمعايير مقاييس الجودة والتي يتم وضعها من خلال **مجموعة مراقبة الجودة** من فريق التطوير بعد تحديد متطلبات المشروع البرمجي مباشرة وقبل البدء في التنفيذ الفعلي للمشروع، وتتم عملية مراقبة الجودة عند الانتهاء من أي جزئية أو مكون من المشروع حيث يتم اختباره وفقاً للمعايير المتفق عليها سابقاً ، وعند وجود أي خلل أو خطأ فعلى الفريق المختص إصلاحه.

- عملية التوثيق ليست مرحلة منفصلة ولكنها تتم بالتوازي مع المراحل السابقة ، وهي لا تشمل عملية التعليق على الشفرة البرمجية فقط ولكنها تشمل العديد من العمليات مثل **كتابة ملفات المساعدة ، وكتيب المستخدم ، وكتيب التدريب، وكتيب المرجع ، وكتيبات التثبيت** ولا غرابة أن يكون حجم ما يكتب من سطور لتوثيق النظام يزيد على مرتين من عدد سطور النظام نفسه.

- بعد عمليات التحليل ، والتصميم ، والبرمجة ، والاختبار والتوثيق ، لا بد أن **توزع هذه البرمجيات وتثبت على أجهزة العملاء.**

- ومع بداية إصدار وتوزيع المنتج البرمجي تبدأ مرحلة الصيانة، ومصطلح الصيانة هنا يصف **النشاطات** التي تتم بعد إصدار المنتج البرمجي والتي تشمل **عملية تصحيح الأخطاء** ، وتطوير البرمجية لتعمل مع **أوساط جديدة وحاسبات جديدة وأنظمة تشغيل جديدة** ، **وزيادة فاعلية البرمجية**، وفي كثير من الحالات تكون **الصيانة هي الأكثر كلفة** من ناحية المصاريف والوقت في دورة حياة تطوير البرمجية.

وبالطبع فإن كل العمليات السابقة لا بد من إدارتها من خلال إدارة واعية وعلى دراية كاملة بجميع مراحل تطوير مشاريع البرمجيات ، حيث تمثل عملية إدارة المشروع النشاط المهم والحرص لتطوير البرمجيات.

٣. مبادئ هندسة البرمجيات (Software Engineering Principles)

لهندسة البرمجيات أهمية كبيرة ، إذ أنها تساعد في (١) **زيادة المردود لشركات النظم البرمجية** ، **ومستخدمي البرمجيات** ، من خلال (٢) **التوزيع السليم للموارد المتوفرة** ، (٣) **والتصميم الجيد لهذه البرمجيات** مما يقلل من **تكاليف وزمن إنتاجها** ، (٤) **مع زيادة وتحسين مستوى جودتها وموثوقيتها** ، ويتم ذلك من خلال تطبيق مجموعة من **المبادئ الأساسية** التي تتميز بها والتي من أهمها ما يلي :

■ **الدقة والصورية (Rigor And Formality)** : يمكن الحصول على منتج موثوق ومضبوط الكلفة من خلال الدقة في عملية الإنتاج وتوجد درجات متنوعة للدقة أعلاها الصورية التي تتطلب أن تكون المنتجات مصممة ومقيمة بطريقة رياضية، وبالتالي **تقتضي الدقة الصورية** ولكن العكس غير صحيح إذ يمكن أن نكون دقيقين بطرق غير صورية (رياضية). يحتاج أي عمل لتنفيذه إلى خطوات وإذا نفذنا هذه الخطوات اعتماداً على خبرات وتجارب ونظريات عندها ستزداد الدقة، فإذا لم تتوفر هذه الاعتمادات (الخبرات والتجارب والنظريات) نعتمد عند ذلك على التمثيل الرياضي وهذه هي الصورية واللغات البرمجية أدوات صورية للتعبير عن وظائف ثم تحول المترجمات اللغة الصورية إلى لغة الآلة.

■ **فصل الاهتمامات (Separation of Concern)** : يعتمد هذا المبدأ على اتباع سياسة "فرق تسد" حيث يتم تقسيم المشكلة إلى مشاكل أصغر غير مرتبطة ببعضها البعض أو يكون ارتباطها صغيراً ، ثم التركيز على حل كل مسألة على حدة. **ويتم فصل الاهتمامات حسب الطرق التالية :**

- **حسب الزمن** : جدولة الأعمال وفق محور الزمن.
- **حسب وجهات النظر**: مثلاً عند تحليل متطلبات نظام برمجي ، قد يكون من المفيد التركيز على

المعطيات التي تتعامل معها البرمجية من جهة والتركيز على الوظائف والتحكمات التي تقوم بها البرمجية من جهة أخرى.

- **حسب الخواص والمواصفات :** يتضمن معالجة المواصفات المطلوبة من البرمجية على حدة، فعلى سبيل المثال إذا كان المطلوب بناء نظام برمجي فعال وصحيح ، فيتم أولاً تصميمه بطريقة متأنية ومنظمة تضمن صحته ثم يتم التغيير في النظام لتحسين فعاليته.
- **حسب الأقسام :** وهو عبارة عن فصل التعامل مع الأقسام المكونة للنظام البرمجي كل على حدة.

■ **التجزئة (Modularity) :** وهي تقسيم النظام البرمجي المعقد إلى أقسام أبسط تسمى كتلاً ، ويهدف مبدأ التجزئة في هندسة البرمجيات إلى :

- القدرة على تجزئة النظام البرمجي إلى كتل (التجزئة التتالي (الشجري) للنظام البرمجي).
- القدرة على تركيب النظام البرمجي من كتل (التركيب التصاعدي للنظام).
- القدرة على فهم الكتل كلاً على حدة لتعديلها (الاستقلالية).

يجب أن يتحقق في الكتل :

- تماسك قوي داخلي للكتل (الكتلة مبنية بشكل منطقي لتحقيق هدف محدد).
- ترابط ضعيف بين الكتل (الاستقلالية).

■ **التجريد (Abstraction) :** وهو تحديد المظاهر المهمة وتجاهل تفاصيلها أثناء دراسة مسألة معينة (فصل الشيء المهم عن الشيء غير المهم في المسألة) ، وهي نظرة نسبية مرتبطة بالهدف الذي نعمل عليه.

■ **توقع التغيير (Anticipation of Change) :** وهو إحصاء الأماكن التي يمكن أن يتم تعديل عليها في البرمجيات ، ومن أكثر هذه الأماكن تغييراً :

- **الخوارزميات :** يمكن أن تكون هناك عدة خوارزميات تقوم بنفس الوظيفة (الفرز مثلاً) وبالتالي فاستبدال إحدى هذه الخوارزميات بأخرى أفضل منها، هو أمر محتمل ، ولذلك يفضل كتابة الخوارزمية في وحدة برمجية منفصلة.
- **تمثيل المعطيات :** يتغير أداء البرنامج بتغير بنية المعطيات التي يستخدمها.
- **الآلات التجريدية:** يمكن اعتبار البرامج التي نكتبها يتم تنفيذها على آلات تجريدية تتفهم التعليمات والأوامر المكتوبة بلغة عالية المستوى، أي أن لكل لغة عالية المستوى آلة تجريدية تتفهم تعليمات اللغة وتنفذها بشكل تجريدي وبعيد عن العتاد الصلب.

- **الوسط الاجتماعي** : التغيير في الوسط الاجتماعي يؤدي إلى تغيير النظام البرمجي المرافق (مثلاً تغيير العملة يؤدي إلى تعديل النظم البرمجية البنكية الموجودة) .
- **الطرفيات** : يتعلق هذا النوع بالبرمجيات التي تحتاج للتعامل مع طرفيات خاصة كأنظمة التحكم. وبالتالي تغيير الطرفيات يتطلب تغيير النظم البرمجية المرافقة.

- **العمومية (Generality)** : يجب علينا عند طرح مشكلة أو مسألة أن نحاول إيجاد مسألة أعم تحوي المسألة المطروحة، لأنه قد يكون حل المسألة الأعم أسهل من المسألة الأصلية، ويمكن أن يكون الحل العام قابلاً لإعادة الاستخدام. وكما يمكن أن يكون الحل العام موجوداً في المكتبات البرمجية الجاهزة، ويعتمد هذا المبدأ عندما يكون الهدف من إنتاج النظام البرمجي تسويقياً.
- **إعادة الاستخدام (Reuse)** :

يشير هذا المصطلح إلى تطوير برمجية معتمده على استخدام أجزاء جاهزة تم إعدادها وفقاً لمعايير الجودة حيث تم اختبارها من قبل مما يزيد من إنتاجية المشروع وتوفير المزيد من الوقت، ويمكن النظر لهذا المصطلح - أيضاً - على أنه استخدام ما هو متاح فعلياً لتحقيق ما هو مطلوب.

٤. أنواع البرمجيات

البرمجيات تحيط بنا في كل مكان : في المناطق الصناعية ، وفي الاستعمالات الخاصة ، وفي أنظمة الاتصالات ، وفي أنظمة النقل ، وفي مجالات أخرى، فالبرامج تأتي مختلفة في أشكالها وأحجامها، فمنها ما يدمج في الهواتف المحمولة ، ومنها ما تقوم بتصميم مركبات الفضاء. أما عن **تصنيف البرمجيات** ، نستطيع أن نميز نوعان رئيسيان:

- **برمجيات النظام (System Software)** : هي البرامج التي تمثل الأدوات التي تساعد في بناء أو دعم البرمجيات التطبيقية. وتتميز برمجيات الأنظمة بالتفاعل الكثيف مع عتاد الحاسب الآلي وباستعمال كثيف والتشارك في الموارد من قبل المستخدمين ، مثل : أنظمة التشغيل ، و مترجمات اللغات ، وخلافه.
- **البرمجيات التطبيقية (Application Software)** : هي البرامج التي تساعد في تأدية بعض المهام المفيدة أو الممتعة بشكل مباشر وأمثلة ذلك : الألعاب ، برامج الصرافة الآلية (ATMs) ، برامج التحكم في الطائرة وبرامج البريد الإلكتروني وبرامج معالجة النصوص ، وبرامج الجدولة. وضمن تصنيف البرمجيات التطبيقية ، من المفيد أن نميز أصناف البرمجيات التالية:
- **برمجيات الألعاب (Games Software)**.
- **برمجيات نظم المعلومات (Information Systems Software)** : وهي من أهم وأوسع مجالات التطبيقات البرمجية ، حيث تقوم هذه النظم بالاتصال بقاعدة بيانات كبيرة أو أكثر تحتوي على كميات

ضخمة من البيانات والمعلومات الخاصة بهذه النظم وبطريقة تسهل الحصول على المعلومات واتخاذ القرارات ، والتفاعل مع المستخدم لإتمام عملياته بيسر وموثوقية كبيرة ، ومن هذه الأنظمة على سبيل المثال : نظام حجز مقعد بالخطوط الجوية ، نظام المعاملات البنكية ونظام شئون الموظفين والرواتب ، خلافة من الأنظمة.

- **برمجيات نظم الوقت الحقيقي (Real – Time Systems Software) :** هي الأنظمة التي يجب أن تعطي إجابة ضمن شروط زمنية محددة ، وتتألف مكونات برمجيات الزمن الحقيقي من : مجسات لجمع وتصفية البيانات من المحيط الخارجي ، ومحول يقوم بتحويل البيانات إلى الشكل الذي يقبله النظام ، بالإضافة إلى مكوّن تحكم وإخراج إلى المحيط الخارجي ، ومراقب تحكم ينسق بين جميع هذه المكونات لتستمر المحافظة على الاستجابة بالزمن الحقيقي. ومثال على ذلك برمجيات التحكم في مراكز الطاقة ، والأقمار الصناعية ، والعمليات الصناعية المختلفة.

- **البرمجيات المدمجة بالأجهزة (Embedded Software) :** تقوم هذه البرمجيات بأداء وظائف خاصة ضمن الأجهزة الذكية من خلال برمجة ذاكرات للقراءة فقط بهذه البرمجيات وتركيبها ضمن هذه الأجهزة. مثال على هذه الوظائف : التحكم في سرعة محرك الغسالة الكهربائية ، وكذلك التحكم في اختيار برامج تشغيلها.

- **البرمجيات المكتبية (Office Software) :** وهي برمجيات موجه لأجهزة الحاسبات الشخصية (PC Computer) لأتمتة الأعمال المكتبية مثل : مجموعة برمجيات ميكروسوفت أوفيس كبرنامج معالجة الكلمات وتنسيق النصوص ، وبرنامج الجداول الحسابية ، وتقديم العروض التقديمية ، وخلافة . وكذلك برمجيات الإنترنت والبريد الإلكتروني ، وخلافة من هذه البرمجيات.

- **البرمجيات العلمية والهندسية (Engineering and Scientific Software) :** تتميز هذه الأنظمة بمحاكاة الأنظمة وتعتمد على الخوارزميات العددية ، وتتراوح تطبيقاتها من إيجاد جذور معادلة من الدرجة الثانية إلى تحليل إحداثيات مركبات الفضاء وإجراء التفاعلات النووية .

- **برمجيات الذكاء الاصطناعي (Artificial Intelligence Software) :** تعتمد برمجيات الذكاء الاصطناعي على مجموعة من الحقائق (Facts) والعلاقات (Relationships) . وتقوم هذه البرمجيات بتحديد الحل المناسب من خلال الاستنتاج والاستدلال المنطقي (Logical Deductions) وكفاءة عالية ، حيث يتم تمثيل العلاقات بين الأشياء وتجميعها وتنظيمها للوصول إلى استنتاج منطقي للحقائق التي تمثلها تلك العلاقات. ومثال ذلك برمجيات التعرف على الأشكال (الصور والصوت).

- **البرمجيات المغلفة (Shrink-wrap Software) :**

يطلق هذا المصطلح (البرمجيات المغلفة) على البرمجيات التطبيقية المعدة للتوزيع التجاري

(Commercial Software Packages) (مثل : ميكروسوفت أوفس). ويتم تحويل البرمجيات التطبيقية المعدة للاستخدام الداخلي (In-House Application Software) إلى برمجيات تطبيقية معدة للتوزيع التجاري (برمجيات مغلقة) بإضافة العديد من الوحدات البرمجية الجديدة الصغيرة إليها مثل برمجية حماية الشيفرة ، وبرمجية التثبيت وبرمجية الرخص ، وخلافه ، وبعد ذلك يتم وضعها كحزمة برمجية تطبيقية تجارية قابلة للتثبيت على أقراص مدمجة مغلقة تُعد للبيع التجاري. ومن أهم الخصائص أو الاعتبارات الهامة التي يجب أن تؤخذ في الاعتبار عند تطوير هذه البرمجيات ما يلي :

① سهولة التكيف مع منصات تشغيل الحاسب الآلي (Multi-Platform Resilience) : لا بد أن يراعى في عملية التطوير للحزمة بأن تكون صالحة للعمل على منصات التشغيل المختلفة للحاسب (Platform Independent) ، وإذا كانت غير موائمة لمنصات معينه ، يجب العمل على تكييفها للعمل معها عن طريق إضافة وحدات برمجية تقوم بهذا الغرض.

② الخيارات الإقليمية وخيارات اللغة (Localization) : يجب أن تتوافق هذه البرمجيات مع منصات التشغيل ذات اللغات المتعددة (Multi-Languages Platforms) عن طريق إضافة وحدة برمجية (Multi-Languages Service Pack) يتم من خلالها اختيار الخيارات الإقليمية وخيارات اللغة بسهولة تامة مما يساعد على انتشار البرمجية وبالتالي على تسويقها.

③ إدارة عملية الترقية (Patch Management) : وهي العملية الخاصة بتنسيق عمليات التحديثات اللازمة للإصدارات ، وكيفية توزيع الرقيعات (Patches) إلى العملاء لسد الثغرات التي تنشأ في الحزمة أثناء عمليات التشغيل (مثل الرقيعات الأمنية التي توزعها شركة ميكروسوفت لسد الثغرات الأمنية التي تكتشف بعد توزيع نظام التشغيل) ، وكيفية حماية العملاء من أي فقد للبيانات أو بيانات التشكيل ، وكيفية تحديث الحزمة.

④ رخصة الاستخدام وعملية التنشيط للمنتج (Activation License Control and Product) : عملية تنشيط المنتج البرمجي هي عبارة عن إجرائية للتحقق من صلاحية رخصة استخدامه (License Activation Procedure) لمنع استخدام المنتج لغير المرخص لهم ، لذا يجب تضمين هذه العملية في البرمجيات التجارية للحفاظ على حق الملكية ، ويجب أن تتضمن أيضاً طريق سهلة لعملية التنشيط من قبل العملاء، ويوجد العديد من أنواع رخص التشغيل وهي رخص محددة بوقت معين وتتم لنسخ المنتج التقويمية (Time Limited Evaluation Edition) (مثل البرمجيات المشتركة التي توزع من خلال الإنترنت للاستخدام لمدة شهر) ، أو رخص استخدام غير محددة الوقت (مثل رخص التشغيل الخاصة بحزمة ميكروسوفت أوفس).

⑤ حماية البرنامج المصدر (Source Code Protection) :

لا بد من حماية برنامج المصدر من العابثين الذين يحاولون الحصول عليه ، وذلك باللجوء إلى وسائل التشفير والحماية المختلفة.

⑥ وجود وسائل المساعدة (Help) :

لا بد من احتواء الحزمة على طرق لمساعدة العميل في استخدام البرنامج بمعظم اللغات العالمية المشهورة ، وكذلك إرشادات في حالة حدوث مشاكل (Troubleshooting) ، ويجب أيضاً تخصيص مواقع على شبكة الإنترنت للمنتج لتقديم المساعدة للعملاء.

⑦ التثبيت (Installation) :

وفقاً للأبحاث الحديثة الخاصة بتثبيت البرمجيات وجد أن حوالي ٣٠% من فشل حزم البرمجيات ينشأ أصلاً من عدم التثبيت الصحيح ، لذا لا بد من التفكير جدياً في طريقة قوية لعملية التثبيت لتجنب الأخطاء التي تؤدي إلى فشل التثبيت ، ولا بد من الأخذ في الاعتبار أنواع الحاسبات المختلفة ، والتفكير في العمليات غير العادية في عملية التثبيت كوضع مفاتيح للتسجيل والرخص ، وعملية التنشيط.

أسئلة تقويم ذاتي

١. ما هي البرمجيات المغلفة؟ كيف يمكنك تحويل البرمجيات التطبيقية المطورة للاستخدام الداخلي لتصبح برمجيات مغلفة.

٢. الجدول التالي يحتوي العمود الأول منه على الأنماط المختلفة للبرمجيات والعمود الثاني منه على تعريفات لهذه الأنماط وهي غير مرتبة، أعد بناء الجدول بحيث يكون هناك توافق بين الأنماط والتعريفات.

أنماط البرمجيات	تعريفها
<ul style="list-style-type: none"> برمجيات النظام برمجيات تطبيقية برمجيات الألعاب برمجيات نظم الوقت الحقيقي البرمجيات المدمجة بالأجهزة البرمجيات المكتنية البرمجيات العلمية والهندسية برمجيات الذكاء الاصطناعي 	<ul style="list-style-type: none"> برمجيات موجهة لأتمتة الأعمال المكتبية. برمجيات لمحاكاة الأنظمة وتعتمد على الخوارزميات العددية. برمجيات تعتمد على مجموعة من الحقائق والعلاقات حيث يتم تحديد الحل المناسب من خلال الاستنتاج والاستدلال المنطقي. برمجيات تعطي إجابة ضمن شروط زمنية محددة. تقوم هذه البرمجيات بالاتصال بقواعد البيانات بطريقة تسهل الحصول على المعلومات واتخاذ القرارات. برمجيات تقوم بأداء وظائف خاصة ضمن الأجهزة. برمجيات للتسليّة وزيادة الاداء والمعرفة.

٥. أدوات هندسة البرمجيات بمساعدة الحاسوب

(CASE Tools)

CASE : هي كلمة مؤلفة من أوائل حروف عبارة (Computer Aided Software Engineering)

(هندسة البرمجيات بمساعدة الحاسب) وهي تشمل طائفة واسعة من الأنواع المختلفة من البرامج التي تُستخدم في مساندة الأنشطة الخاصة بالعمليات البرمجية مثل تحليل المتطلبات وإعداد النماذج التحليلية وتصحيح الأخطاء والاختبار ، وتتوافق جميع الطرق في أيامنا هذه مع تقنية CASE ومنها أجهزة التمثيل بالرموز وفي وحدات التحليل المستخدمة في تدقيق نماذج النظام وفق قواعد توثيقية عملية لإنشاء النظام. وقد تشمل أدوات CASE كذلك على معدات الترميز (التكويد) والتي تقوم بصورة آلية بإعداد تكويد للبرامج من واقع نموذج النظام إضافة إلى إعداد الخطوط الإرشادية الخاصة بتنفيذ العملية وتقديم النصح لمهندسي البرامج حول الخطوة التالية التي يجب القيام بها.

وهذا النوع من أدوات CASE والذي يهدف إلى إسناد ودعم عملية التحليل والتصميم يسمى في بعض الأحيان بأداة CASE العليا ، لأنه يقدم المساندة للمراحل المبكرة من عملية البرمجة وخلافا لذلك فإن أدوات CASE المعدة لتقديم المساندة والدعم لعملية التنفيذ والاختيار كأنظمة تصحيح الأخطاء وتحليل البرامج وإعداد متطلبات الاختيارات وأدوات تحرير البرامج تسمى أحياناً بأدوات CASE السفلى .

ويمكن القول بأن أدوات هندسة البرمجيات بالاستعانة بالحاسب الآلي هي : عبارة عن برامج مصممة لمساعدة المبرمجين على التغلب على تعقيدات العمليات البرمجية المختلفة والمساعدة على أتمنتها ، وهي تمثل مجموعة من الضوابط والآليات التي تساعد في تحليل ، وتصميم ، ونمذجة وترميز ، واختبار ، وتوثيق البرمجيات. ويُعد برنامج محرر النصوص كمثال وأداة قيمة من أدوات تطوير البرمجيات لتنظيم الملف أو البرنامج ، وكذلك برنامج محرر الرسوم كمثال آخر لتوثيق الرسوم الخاصة بالتصميم للنظم الكبرى من البرمجيات ، والمثير للجدل أن هذه الوسائل المساعدة لم يتم اكتشافها وتقنينها بصورة كاملة ، فكلما ظهرت احتياجات برمجية جديدة أخرى أصبح من الضروري اكتشاف وسائل وأدوات جديدة مناسبة لها.

1.5 ميزات أدوات هندسة البرمجيات بمساعدة الحاسوب

(CASE Tools Advantages)

يمكن تلخيص أهم ميزات أدوات هندسة البرمجيات كالتالي :

① **السرعة الفائقة (Increased Speed)** : تتميز أدوات هندسة البرمجيات **بالسرعة الفائقة في أتمتة العمليات وتخفيض الوقت** اللازم لاستكمال العديد من المهام خاصة التي تتطلب المخططات الرسومية والبنود الخاصة بها. وتقدر التحسينات في الإنتاجية بعد التطبيق في مدى يتراوح من (٢٠٠ - ٣٠٠٪).

② **الدقة المتزايدة (Increased Accuracy)** : تساعد أدوات هندسة البرمجيات في **اكتشاف الأخطاء** والذي تُعد خطوة مهمة في إزالة هذه الأخطاء حيث يوفر اكتشاف الأخطاء وإزالتها الوقت والجهد في المراحل المبكرة من إعداد النظام ، و كلما زاد حجم النظام أصبح من الصعب اكتشاف الأخطاء مما يؤدي إلى زيادة الوقت والجهد المطلوب للتنسيق والإدارة بين فرق العمل المختلفة.

③ **تخفيض الوقت اللازم للصيانة (Reduced Lifetime Maintenance)** : وكنتيجة للتحليل الجيد، التصميم الجيد، توليد الشفرات آلياً، اختبار البرمجيات آلياً، التوثيق الآلي، يتحسن أداء النظام وبالتالي فإن الجهد اللازم للصيانة ينخفض إلى حد كبير. وكذلك يمكن توفير العديد من المنابع لتطوير أنظمة جديدة ، وتقوم أدوات مساعدة البرمجيات (برمجيات إعادة الهندسة **Re-engineering Programs**) باكتشاف الأجزاء من البرمجيات والتي يمكن إعادة استخدامها مما يزيد من كفاء العمل وتقليل الجهد المطلوب.

④ **التوثيق الأفضل (Better Documentation)** : وباستخدام أدوات هندسة البرمجيات المساعدة هناك العديد من كميات التوثيق التي يتم توليدها لتمثل ملاحظات على كيفية تطوير النظام وعمل صيانة له.

⑤ **البرمجة في أيدي غير المبرمجين :**

(Programming in the Hands of Non-programmers) :

مع التطور السريع والاتجاه نحو تكنولوجيا البرمجة الشيئية ، وقواعد بيانات خادم العميل يمكن أن تتم عملية البرمجة بأناس ليس لديهم خلفية كاملة عن البرمجة، حيث يصبح المهم هو فهم الهدف الأساسي من البرنامج والقدرة على تحليل مكونات البرنامج وتفاصيله والتي تستخدم في توليده عن طريق هذه الأدوات (أدوات التطوير منخفضة المستوى).

⑥ **الفوائد الغير ملموسة (Intangible Benefits)** : تفيد أدوات التطوير في مشاركة المستخدم والتي تساهم في قبوله الجيد للنظام الجديد وهذا يساهم إلى حد كبير في تخفيض منحى التعليم الأولي.

أسئلة تقويم ذاتي

ما هي أدوات هندسة البرمجيات بمساعدة الحاسب الآلي ؟ وما هي أهم مميزات وقصور استخدامها؟.



٢,٥ قصور أدوات هندسة البرمجيات بمساعدة الحاسوب

(CASE Tools Limitations)

يمكن تلخيص أهم القصور الموجودة في أدوات هندسة البرمجيات بمساعدة الحاسب في التالي :

① **مزج الأدوات (Tool Mix) :** من المهم جدا اختيار ملائم لمزيج من الأدوات للحصول على مميزات وتكاليف مخفضة. ولابد من اختيار الأدوات غير المعتمدة على نوعية العتاد وإمكانية مشاركة نتائج أداة تطوير تم استخدامها مع أدوات أخرى لضمان تكاملية الأدوات مع بعضها.

② **التكاليف (Cost) :** الأدوات المساعدة ليست رخيصة الثمن ، وفي الحقيقة فإن شركات التطوير على نطاق ضيق لا تستخدم هذه الوسائل معتقدين بأنها مفيدة فقط في تطوير الأنظمة الكبيرة ، حيث أن تكلفة تزويد مطوري الأنظمة بهذه الوسائل مكلف. ويعتبر (العتاد ، البرمجيات ، الاستشارات) كلها عوامل تدخل في معادلة التكلفة.

③ **منحنى التعلم (Learning Curve) :** وفي معظم الحالات فإن إنتاجية المبرمج تصل إلى أدنى مرحلة لها في المرحلة الأولية للتطبيق وذلك لاحتياج المستخدمين للوقت الكافي للتعلم ، بالرغم من أن مستشاري هذه الأدوات يقدمون العديد من الخبرات لمستخدميها حيث يقومون بالتدريب وكذلك وجود العديد من مواقع الخدمات الخاصة بهذه الأدوات على شبكة الإنترنت والتي تساعد في تعجيل منحنى التعلم لهذه الأدوات.

٣,٥ التصنيف العام لأدوات هندسة البرمجيات المساعدة

(CASE Tools General Classifications) :

- ① **أدوات التطوير المساعدة عالية المستوى (Upper Case Tools) :**
- وهي تمثل أدوات أنشطة العمليات المبكرة للمتطلبات ، التحليل والتصميم. وتساعد المحللون في تخزين ، وتنظيم ، وتحليل نماذج العمل ، وترتيب الأسبقيات التالية :
- إستراتيجيات العمل الحالية والمستقبلية.
 - النظم المكتملة وإستراتيجيات تطبيقها.
 - قواعد البيانات والشبكات المراد تطويرها.
 - التطبيقات المراد تطويرها.

- ② **أدوات التطوير المساعدة منخفضة المستوى (Lower Case Tools) :**
- وهي تمثل أدوات دعم الأنشطة التالية مثل البرمجة واكتشاف العلل والاختبارات حيث تساعد

المبرمجين في زيادة الإنتاجية والجودة ، و تمتد هذه الأدوات إلى تفاصيل التصميم للمساعدة في توليد التطبيقات من خلال الخدمات التالية :

- المساعدة في سرعة اختبار البرنامج واكتشاف الأخطاء.
- توليد الكود للبرنامج من مواصفات التحليل والتصميم.
- توليد الشاشات وقواعد البيانات.

⌚ أدوات هندسة البرمجيات بمساعدة الحاسب لدورة حياة التطوير المتقاطعة :

(Cross Life Cycle Case Tools) :

وهي تشمل وسائل إدارة المشاريع التي تساعد المديرين في (التخطيط ، وضع الجداول الزمنية ، وإعداد التقارير ، وتوزيع الموارد).

والجدول رقم (1) يوضح أهم تصنيفات أدوات هندسة البرمجيات بمساعدة الحاسب وأمثلة عليها.

جدول(1) : تصنيف أدوات هندسة البرمجيات بمساعدة الحاسب وأمثلة عليها

نوع الأداة (Tool Type)	أمثلة على الأداة (Examples)
أدوات التخطيط (Planning Tools)	<ul style="list-style-type: none"> • أدوات بيرت (PERT Tools) • أدوات التقييم (Estimation Tools) • الجداول الإلكترونية (Spreadsheets)
أدوات التحرير (Editing Tools)	<ul style="list-style-type: none"> • محررات النصوص (Text Editors) • معالجات الكلمات (Word Processors) • محررات المخططات (Diagram Editors)
أدوات إدارة التغيير (Change Management Tools)	<ul style="list-style-type: none"> • أدوات تتبع المتطلبات (Requirements Traceability Tools) • نظم تغيير التحكم (Change Control Systems)
أدوات إدارة التكوين (Configuration Management Tools)	<ul style="list-style-type: none"> • نظم إدارة الإصدار (Version Management Systems) • أدوات بناء النظم (System Building Tools)
أدوات النمذجة الأولية (Prototyping Tools)	<ul style="list-style-type: none"> • اللغات عالية المستوى (Very High-level Languages) • مولدات واجهات المستخدم (User Interface Generators)

نوع الأداة (Tool Type)	أمثلة على الأداة (Examples)
أدوات دعم الطرق (Method Support Tools)	<ul style="list-style-type: none"> • محررات التصميم (Design Editors) • قواميس البيانات (Data Dictionaries) • مولدات الشفرة (Code Generators)
أدوات معالجة اللغات (Language Processing Tools)	<ul style="list-style-type: none"> • المترجمات (Compilers) • المفسرات (Interpreters)
أدوات تحليل البرامج (Program Analysis Tools)	<ul style="list-style-type: none"> • مولدات التداخل (Cross reference Generators) • المحللات الإستاتيكية (Static Analyzers) • المحللات الديناميكية (Dynamic Analyzers)
أدوات الاختبار (Testing Tools)	<ul style="list-style-type: none"> • مولدات اختبار البيانات (Test Data Generators) • مقارنات الملفات (File Comparators)
أدوات اكتشاف وتصحيح الأخطاء (Debugging Tools)	<ul style="list-style-type: none"> • نظم التصحيح التفاعلية (Interactive Debugging Systems)
أدوات التوثيق (Documentation Tools)	<ul style="list-style-type: none"> • برامج شكل الصفحة (Page Layout Programs) • محررات الصور (Image Editors)
أدوات إعادة الهندسة (Re-engineering Tools)	<ul style="list-style-type: none"> • نظم التداخل (Cross-reference Systems) • نظم برامج إعادة الهيكلة (Program Restructuring Systems)

٦. التحديات الرئيسية التي تواجه هندسة البرمجيات

(Key challenges Facing Software Engineering) :

تواجه هندسة البرامج ثلاثة تحديات كبرى في القرن الحادي والعشرين ، وهي :

① التحدي المتعلق بالموروث (The Legacy Challenge) :

لقد تطورت غالبية الأنظمة البرمجية الرئيسية المستخدمة حاليا منذ سنوات عديدة خلت ، ورغم ذلك لا تزال تلك الأنظمة تؤدي وظائف حساسة. إن التحدي المتعلق بالموروث هو التحدي الذي يعني أساسا بصيانة وتحديث البرامج بطريقة يمكن معها تفادي التكاليف الباهظة والاستمرار في الوقت ذاته في تقديم الخدمات الأساسية في قطاع الأعمال.

⌚ تحديد عدم التجانس (The Heterogeneity Challenge) :

تزداد الحاجة وباستمرار إلى أنظمة تعمل كأنظمة موزعة عبر شبكات تشتمل على أنواع مختلفة من الحاسبات الآلية ذات الأنظمة المساندة المختلفة وهذا التحدي يعوق تطوير أساليب إعداد برامج يمكن الاعتماد عليها بدرجة كافية للتعامل مع هذا التحدي.

⌚ تحدي المخرجات (The Delivery Challenge) :

تعد العديد من الأساليب المتبعة في هندسة البرامج مضيعة للوقت. كما أن الزمن الذي تستغرقه يحد وبشكل كبير من فرص تطوير البرامج وضمان جودتها . على الرغم من كل ذلك ، فإنه ينبغي على قطاع الأعمال والتجارة في أيامنا هذه أن يتميز بسرعة الاستجابة والتغيير السريع، كما ينبغي أن تتغير وبنفس الوتيرة برامج المساندة الخاصة بها. إن تحدي المخرجات هذا يرتبط بالزمن اللازم للحصول على مخرجات من أنظمة كبيرة ومعقدة دون التفريط في الجودة والتنوعية. وبالطبع فإن هذه المسألة ليست مسألة منفصلة فقد يلزم مثلاً إجراء تغيير سريع في نظام قديم العهد من أجل جعل النفاذ إليه من خلال الشبكة أمراً سهلاً. وللتعامل مع هذه التحديات يلزمنا توفر أدوات وأساليب جديدة واستخدام طرق إبداعية من أجل القيام بالبرمجة واستخدام الطرق الحالية في هندسة البرمجيات جنباً إلى جنب مع تلك الطرق الإبداعية.

٧. مهنة البرمجة (Programming Career)

مهنة البرمجة من المهن الهامة والمطلوبة في السوق العربية بصفة خاصة والعالمية بصفة عامة ، ولكن بشرط أن يكون المبرمج على كفاءة عالية وقدرة على استخدام معظم الأدوات البرمجية وتوظيفها بصورة مثالية، ومن يريد العمل بمهنة البرمجة يجب أن يكون مستخدماً متمرساً للحاسب الآلي وله خبرة طويلة في التعامل مع شتى أنواع برمجياته ليس فقط كمستخدم عادي ولكن كشخص قادر على فهم كيفية تصميم وإنشاء هذه البرمجيات ، مع القدرة على اكتساب المهارات الخاصة بأدوات البرمجة وتطويرها ، وكذلك يكون له عقل يجيد التعامل مع الأسس الرياضية ، حيث إن مهنة البرمجة لا تعتمد على مجرد أداء المهام فقط وإنما تتطلب فكراً خصباً وذهناً حاضراً وحجاً للإبداع في العمل والمثابرة عليه.

ولتصميم وتطوير البرامج والتطبيقات بطريقة قياسية ، يجب أن يتكون فريق العمل الخاص بذلك من التالي :

- محللو النظم (System Analysts) : وهم الأشخاص القائمون علي دراسة وتحليل متطلبات النظام ومدخلاته ومخرجاته ، وكذلك تحديد الموارد اللازمة لتنفيذه ، بالإضافة إلى بيان كيفية التنفيذ وشرح ديناميكية العمل وتنظيم العلاقات المختلفة بين الكائنات الموجودة بالنظام .

• **مصممو النظم (System Designers):** يأتي دورهم بعد مرحلة التحليل وتحديد الاحتياجات ، حيث يكون النظام بحاجة الآن إلي كيفية التطبيق من حيث الشكل العام وتصميم كائنات ونماذج النظام وبنية كل كائن علي حده. وفي هذه المرحلة يتم تصميم نماذج وأشكال الشاشات ومواضعها وطرق عرضها وربطها مع بعضها البعض.

• **المبرمجون (المطورون) (Programmers (Developers)) :** ويأتي دورهم بعد مرحلتي التحليل والتصميم حيث يتم التنفيذ الفعلي للنماذج والشاشات المصممة وكتابة الشفرات (Source Code) المسؤولة بدورها عن تشغيل النظام .

فمثلا إذا كنا بصدد إنشاء نظام لإدارة شركة ما من الناحية المالية والتجارية ، فسيقوم المحللون بدراسة الدورة المستندية لهذه الشركة ، وكيفية تعاملها مع الشركات الأخرى ، وديناميكية العمل من حيث المستندات المستخدمة في دورات العمل المختلفة...الخ ، وكيفية تدفق البيانات من مرحلة إلي الأخرى ، وبالتالي تحليل النظام ككل بشكل متكامل ثم. بعد ذلك يأتي دور المصممين حيث يتم تصميم نماذج وأشكال الشاشات ومواضعها وطريقة عرضها وربطها ببعض والتي سيبرمجها المبرمجون ، وبعد ذلك يأتي دور المبرمجين حيث يتم التنفيذ الفعلي لما تم تصميمه سابقا حيث يتم كتابة الشفرات اللازمة لإنشاء كل النماذج وربطها بعضها ببعض.

ومن خلال ما سبق ، يتضح أن المبرمج هو الشخص القائم على كتابة الشفرات اللازمة لبث روح الحياة في النظام وجعله وحدة واحدة مترابطة يؤدي في النهاية - عند تشغيله من قبل المستخدم - جميع المهام الذي صمم من أجلها ، وبالتالي فالمبرمج هو حلقة الوصل بين الحاسب الآلي والمستخدم ، فكلاهما لا يعرف لغة الآخر ، ولكن المبرمج يعرف لغة الاثنين.

والمبرمج ليس من تعلم لغة برمجية فحسب ، بل المبرمج هو من يعرف فن البرمجة ، أي كيف يضع الإستراتيجية المناسبة في المكان المناسب ، وهذه بحد ذاتها موهبة ربانية كالرسم والنحت ، أما ما تبقى من العمل البرمجي فلا يتعدى تطويع الذخائر والأدوات البرمجية وتشكيل الفكرة في قالب ذي طابع فني برمجي لإنتاج المنتج البرمجي.

ولكي يبدأ أي شخص بامتهان مهنة البرمجة كوظيفة يجب عليه أولاً التعرف على أنواع لغات البرمجة من حيث نقاط القوة والضعف في كل منها ، وكذلك التطبيقات الخاصة بكل منها ، وتعلم مبادئ البرمجة ومفاهيمها الأساسية والمشاركة بين جميع لغات البرمجة ، وهو الهدف الأساسي من هذا الكتاب الذي بين يديك ، فإذا كنت تريد أن تبدأ في عالم البرمجة فعليك بدراسة هذا الكتاب وبالتسلسل المذكور به

أسئلة تقويم ذاتي



١. ما هي أهم التحديات الرئيسية التي تواجه هندسة البرمجيات؟ وكيف يمكن مواجهتها من وجه نظرك؟

٢. الجدول التالي يحتوي العمود الأول منه على الفرق الأساسية لتطوير البرمجيات والعمود الثاني على وظائف هذه الفرق، أعد بناء الجدول بحيث يكون هناك توافق بين الفرق ووظائفها

فرق تطوير البرمجيات	وظائفها
محلو النظم	• يقوم بتصميم نماذج وأشكال الشاشات ومواقعها وطرق عرضها وربطها مع بعضها البعض.
مصممو النظم	• يقوم بتحليل متطلبات النظام ومدخلاته ومخرجاته وكذلك تحديد الموارد اللازمة لتنفيذه.
المبرمجون	• يقوم بكتابة الشفرات المسؤولة بدورها عن تشغيل النظام.

8. المسؤولية المهنية والأخلاقية لمهندسي البرمجيات

(Professional and Ethical Responsibility for Software Engineers) :

مثلهم في ذلك مثل غيرهم من المهندسين ، فإنه يتوجب على مهندسي البرمجيات القبول بحقيقة أن مهامهم تتضمن مسؤوليات أكبر من مجرد استخدام المهارات الفنية حيث أن عملهم يتم ضمن إطار قانوني واجتماعي معين، ومن الواضح أن هندسة البرمجيات مقيدة بمجموعة من القوانين المحلية والوطنية والدولية، وتبعاً لذلك فإن عليهم أن يتصرفوا بطريقة مسئولة من الناحيتين الأخلاقية والمعنوية إذا كان لهم أن يحوزوا على الاحترام كمحترفين في مجالات عملهم، ومن البديهي كذلك أن يتمسك هؤلاء المهندسون بالقواعد والمعايير المتعارف عليها في مجال الشرف والنزاهة والأمانة والاستقامة. كما ينبغي عليهم عدم استخدام مهاراتهم وقدراتهم بطريقة غير شريفة أو تسيء إلى سمعة المهنة، ومع ذلك فإن هناك مجالات لا تكون فيها معايير السلوك المقبول ملزمة بموجب القانون بل ترتبط بمفاهيم فضفاضة للمسؤولية المهنية، وهذه بعض منها :

- **السرية (Confidentiality) :** يجب على المهندسين في العادة احترام خصوصية وسرية المعلومات الخاصة بعملائهم وأرباب العمل بصرف النظر عما إذا تم توقيع اتفاقية خاصة بالسرية أم لا .
 - **الاختصاص (Competence) :** يتوجب علي المهندسين الامتناع عن إعطاء معلومات خاطئة عن مستوي تأهيلهم واختصاصهم ، كما أن عليهم عدم قبول أي عمل خارج عن نطاق اختصاصهم وهم يعلمون ذلك.
 - **حقوق الملكية الفكرية (Intellectual Property Rights) :** يجب أن يكون المهندسين مطلعين على القوانين المحلية التي تحكم استخدام الممتلكات الفكرية مثل براءات الاختراع وحقوق النشر والطبع والتأليف وغيرها، كما يجب عليهم أن يحرصوا على التأكد من حماية الممتلكات الفكرية لأرباب عملهم وعملائهم.
 - **إساءة استخدام أجهزة الحاسب الآلي (Computer Misuse) :** ينبغي على مهندسي البرمجيات عدم استخدام مهاراتهم الفنية في الإساءة إلي أجهزة الحاسب الآلي التي تعود للآخرين وإساءة الاستخدام تتراوح بين أمور تافهة نسبياً (مثل ممارسة الألعاب على جهاز صاحب العمل) وأمر خطير جداً (كنشر الفيروسات مثلاً).
- وفى هذا الخصوص تلعب الجمعيات والمؤسسات المهنية دورا هاما ، فالمنظمات مثل : "اتحاد أصحاب الآلات الحاسبة" ، و"معهد المهندسين الكهربائيين والإلكترونيين" ، و"جمعية الحاسبات البريطانية" تقوم بنشر لوائح السلوك المهني أو قواعد أخلاق المهنة. كما يتعهد أعضاء هذه المنظمات بالتقييد بتلك اللوائح والقواعد عند التوقيع على العضوية.
- وينبغي على مهندسي البرمجيات الالتزام بجعل مهنة تحليل وتصميم وتطوير واختبار وصيانة البرامج مهنة محترمة وذات فائدة وجدوى، ووفق التزامهم بصحة وسلامة ورفاهية الجمهور فإن على مهندسي البرامج الالتزام بالمبادئ الثمانية التالية :
- ① **الجمهور (Public) :** يجب على مهندسي البرمجيات العمل بما يتوافق مع المصلحة العامة.
 - ② **العميل وصاحب العمل (Client And Employer) :** يجب على مهندسي البرمجيات العمل بطريقة تخدم مصالح عملائهم وأرباب الأعمال وتكون متوافقة مع المصلحة العامة.
 - ③ **المنتج (Product) :** يجب على مهندسي البرمجيات التأكد من أن منتجاتهم والتعديلات المتعلقة بها تفي بأعلى المستويات المهنية الممكنة.
 - ④ **النزاهة (Judgment) :** يجب على مهندسي البرامج المحافظة على التكامل والاستقلالية في آرائهم المهنية.
 - ⑤ **الإدارة (Management) :** يجب على المدراء والقياديين في مجال هندسة البرمجيات المساهمة والتعهد بالالتزام بالمنهج الأخلاقي في إدارة عملية تطوير وصيانة البرمجيات.

⌚ **المهنة (Profession)** : يجب على مهندسين البرمجيات تحسين سمعة وصورة المهنة وبما يتوافق مع المصلحة العامة.

⌚ **الزملاء (Colleagues)** : يجب على مهندسي البرامج أن يكونوا منصفين وعادلين مع زملائهم وان يقدموا الدعم والمساندة لهم.

⌚ **الذات (Self)** : يجب على مهندسي البرامج المشاركة في برامج التعليم الدائم والمتعلق بممارسة مهنتهم كما أن عليهم تعزيز الأسلوب الأخلاقي في ممارسة هذه المهنة.

أسئلة تقويم ذاتي



1 ضع علامة (✓) أمام الإجابة الصحيحة وعلامة (X) أمام الإجابة الخطأ. صحح الإجابة الخطأ؟ :

- هندسة النظم جزء من هندسة البرمجيات.
- مصطلح هندسة البرمجيات يحتوي على شقين أحدهما "هندسي" تطبق فيه النظريات والأساليب للحصول على الحلول المثالية للمشاكل والآخر "برمجي" يهتم بجميع مظاهر إنتاج البرمجيات.
- هندسة البرمجيات لا تهتم فقط بالعملية الفنية لتطوير البرمجيات ولكن تهتم أيضا بالنشاطات الأخرى مثل إدارة مشروع البرمجيات.
- علوم الحاسبات تركز على أنظمة الحاسبات ولغات البرمجة لدراساتها وتطويرها بينما هندسة البرمجيات تهتم بالمشاكل العملية لإنتاج البرمجيات.
- تحديد المتطلبات وقيود تشغيلها وتطوير النظام البرمجي والتحقق من أداء النظام البرمجي عبارة عن الأنشطة الأربعة الأساسية في جميع عمليات البرمجيات.
- أدوات هندسة البرمجيات بالاستعانة بالحاسب عبارة عن أنظمة برمجية مصممة لإسناد الأنشطة الروتينية لعمليات إنتاج البرامج.

٢. رتب (ترتيباً تصاعدياً) المهام التالية التي يجب أن تتبع في تطوير المشاريع البرمجية :

إدارة المشاريع ، صيانة البرمجية ، تصميم البرمجية ، اختبار تكاملية البرمجية ، تحديد المتطلبات، توثيق البرمجية، تقدير الموارد، الترميز(عملية البرمجة)، التثبيت وتوزيع البرمجية، التدريب، تحليل المشكلة.



اختر المصطلح المناسب والصحيح لكل من التعريفات التالية :

- تسليم المنتج البرمجي في الوقت المحدد.
- إمكانية التشغيل والتعامل مع أنظمة أخرى عادية أو شبكية.
- يمكن إعادة استخدامها مع مشاريع أخرى لأنها مكتوبة وفقا لمعايير.
- يمكن اختبارها بسهولة.
- يمكن نقلها إلى أي نظام تشغيل أو حاسب بدون إعادة كتابة أي أجزاء جديدة.
- تعني توفر ربط بيني مناسب للمستخدم ومادة وثائقية كافية.
- يمكن صيانتها إذا حدثت أي مشاكل أثناء التشغيل.
- تعني تنفيذ البرمجية في الوقت المحدد من خلال الموارد المتاحة.
- تعني عدم تذبذب أداء البرمجية من وقت لآخر.
- يتصرف النظام بشكل يتوافق مع المتطلبات الوظيفية.

الخلاصة

اشتملت هذه الوحدة على تعريف هندسة البرمجيات والتدقيق في مصطلح "هندسة البرمجيات" بشقيه الهندسي الذي يطبق فيه المهندسون النظريات والأساليب والأدوات الملائمة ، الشق البرمجي والذي يهتم بجميع مظاهر إنتاج البرمجيات.

كما استلمت على المهام النموذجية لهندسة البرمجيات وهي تحليل المشكلة وتحديد المتطلبات وتصميم البرمجية والترميز اختبار تكاملية الشفرة البرمجية والتنشيط وتوزيع البرمجية وتوثيق البرمجية وصيانة البرمجية. مبادئ هندسة البرمجيات وهي الدقة والفورية ، فصل الاهتمامات ، التجزئة ، توقع التغيير ، العمومية ، إعادة الاستخدام. أنواع البرمجيات وهي برمجيات النظام والبرمجيات التطبيقية وبرمجيات الألعاب وبرمجيات نظم المعلومات وبرمجيات نظم الوقت الحقيقي والبرمجيات المدمجة بالأجهزة والبرمجيات المكتبية والبرمجيات العلمية والهندسية وبرمجيات الذكاء الصناعي.

أدوات هندسة البرمجيات بمساعدة الحاسب الآلي والتي تمثل مجموعة من الضوابط والآليات التي تساعد في تحليل ، وتصميم ، ونمذجة وترميز ، واختبار ، وتوثيق البرمجيات. للتحديات الرئيسة التي تواجه هندسة البرمجيات والمتمثلة في التحدي المتعلق بالموثوقية - تحديد عدم التجانس - تحديد المخرجات.

مهنة البرمجة كما تناولت أخيراً التصميم وتطوير البرامج والتطبيقات بطريقة قياسية ، يجب أن يتكون فريق العمل الخاص بذلك من : محلي النظم ، مصممي النظم ، المبرمجين. المسؤولية المهنية والأخلاقية لمهندسي البرمجيات وهي لوائح السلوك المهني وهي السرية والاختصاص وحقوق الملكية الفكرية وإساءة استخدام أجهزة الحاسب الآلي. كم أن هناك مبادئ يجب أن يلتزم بها مهندس البرامج وهي الجمهور والعميل والمنتج والنزاهة والإدارة، المهنة والزملاء والذات. أرجو أن تساهم هذه الخلاصة في مساعدتك على استذكار ما ورد في هذه الوحدة

إجابة التدريبات

تدريب (١)

بصورة أساسية ، تهتم علوم الحاسبات الآلية بالنظريات والطرق التي تؤكد دراسة أنظمة الحاسبات والأدوات البرمجية ، بمعنى أنها تركز على أنظمة الحاسبات ولغات البرمجة لدراساتها وتطويرها في ذاتها ، أي أنها تُعد أداة نستخدمها عند تصميم وتطوير حل لمشكلة ما ، وذلك مثل علم الكيمياء حيث يهتم

الكيميائي بالمواد الكيميائية ذاتها من خلال دراسة تركيباتها ، وتفاعلاتها ، والنظريات التي تحكم سلوكها. في حين أن **هندسة البرمجيات** تهتم بالمشاكل العملية لإنتاج البرمجيات ، أي أن مهندس البرمجيات يعتبر أن الحاسبات الآلية أداة لحل المشاكل ، وذلك مثل المهندس الكيميائي الذي يعتبر الكيمياء أداة لإيجاد الحلول لمشاكل عامة في الصناعات الكيميائية وغيرها. ومعرفة بعض علوم الكمبيوتر ضرورية وأساسية لمهندسي البرمجيات وبنفس الطريقة فإن بعض المعرفة للفيزياء ضرورية لمهندسي الكهرباء. مثالياً ، يجب أن تعزز جميع عمليات هندسة البرمجيات بنظريات علوم الحاسبات الآلية ولكن في الحقيقة ليست هذه هي القضية أو الحالة ، بل يجب أن يستخدم مهندسو البرمجيات أحياناً أساليب خاصة لتطوير البرمجيات ، حيث إن نظريات علوم الحاسبات الآلية لا يمكن تطبيقها دائماً للمشاكل الحقيقية والمعقدة التي تتطلب حلول برمجية خاصة.

تدريب (٢)

تهتم **هندسة النظم** بجميع العمليات المتعلقة بتطوير الأنظمة المعتمدة على الحاسبات الآلية (Computer-Based Systems) بما في ذلك الأجزاء الصلبة للحاسبات الآلية (Hardware) والبرمجيات (Software) والعمليات الهندسية المتعلقة بعمليات التطوير المختلفة والتي تتضمن هندسة البرمجيات ، أي أن هندسة البرمجيات جزء من هندسة النظم. ويختص مهندس النظم بتعريف وتحديد هندسة النظام الكاملة ومن ثم تكامل الأجزاء المختلفة لخلق وابتكار النظام المكتمل.

تدريب (٣)

عمليات البرمجيات هي مجموعة من الأنشطة المرتبة (الطبقية) المترابطة (Consistency Activities) والقيود (Constraints) والموارد (Resources) تهدف إلى توصيف (Specifying) وتصميم (Designing) وتنفيذ (Implementing) واختبار (Testing) وصيانة وترقية (Maintenance and Evolution) منتج برمجي يفي بجميع متطلبات العميل. وغالباً ما يقوم بهذه الأنشطة مهندسو البرمجيات. وتوجد أربعة أنشطة أساسية في جميع عمليات البرمجيات ، وهي:

- **تحديد المتطلبات وقيود تشغيلها** وهي ما يجب أن يقوم به النظام البرمجي والقيود المفروضة أثناء عملية التطوير.
- **تصميم وتنفيذ (تطوير) النظام البرمجي.**
- **التحقق من أداء النظام البرمجي** وهو يعني التأكد من أن النظام البرمجي يفي بجميع متطلبات العميل ومقاييس الجودة المطلوبة.
- **الصيانة والارتقاء** وهو صيانة وتغيير بعض من أجزاء النظام البرمجي تجاوباً مع المتغيرات في متطلبات العميل.

لمحة مسبقة عن الوحدة التالية

عزيزي الدارس ،،

بعد أن فرغت من أهم المفاهيم الأساسية في هندسة البرمجيات في الوحدة الأولى ينتقل بك المقرر إلى الوحدة التالية التي سنتابع فيها دورة حياة (عمليات) البرمجيات من مرحلة تجميع المتطلبات إلى مرحلة التطوير ثم مرحلة الاختبار وتشخيص الأخطاء وأخيراً مرحلة الصيانة والارتقاء .
نرجو أن تكون وحدة مفيدة لك وأن تساهم معنا في نقدها وتقويمها.

مسرد المصطلحات

هندسة البرمجيات : Software Engineering

علم يهتم ببناء الأنظمة البرمجية الكبيرة والمعقدة بواسطة فريق من مهندسي البرمجيات بإتباع طرق ومبادئ هندسية منظمة وصحيحة واستخدام وسائل وأدوات وتقنيات تجعل من تصميم وتطوير وبناء واختبار هذه النظم عملية منظمة يمكن تكرارها في حدود الإمكانيات المتاحة مادياً وبشرياً وزمنياً ، وذلك من خلال دراسة دورة حياة النظام مع إعطاء أشكالاً متعددة لعمليات تصميمه وتطويره وبناءؤه واختباره بحيث يكون المنتج البرمجي النهائي على درجة عالية من الجودة والموثوقية وقليل التكاليف بقدر الإمكان ويتم تسليمه للعميل في الوقت المناسب ويعمل بكفاءة على أجهزة الحاسب الآلي مختلفة.

علوم الحاسبات الآلية : Computer Sciences

هي العلوم التي تهتم بالنظريات والطرق التي تؤكد على دراسة أنظمة الحاسبات والأدوات البرمجية ، بمعنى أنها تركز على أنظمة الحاسبات ولغات البرمجة لدراستها وتطويرها في ذاتها ، أي أنها تُعد أداة نستخدمها عند تصميم وتطوير حل لمشكلة ما ، وذلك مثل علم الكيمياء حيث يهتم الكيميائي بالمواد الكيميائية ذاتها من خلال دراسة تركيباتها ، وتفاعلاتها ، والنظريات التي تحكم سلوكها.

هندسة النظم : System Engineering

تهتم هندسة النظم بجميع العمليات المتعلقة بتطوير الأنظمة المعتمدة على الحاسبات الآلية (Computer-Based Systems) بما في ذلك الأجزاء الصلبة للحاسبات الآلية (Hardware) والبرمجيات (Software) والعمليات الهندسية المتعلقة بعمليات التطوير المختلفة والتي تتضمن هندسة البرمجيات ، أي أن هندسة البرمجيات جزء من هندسة النظم.

عمليات البرمجيات : Software Processes

هي مجموعة من الأنشطة المرتبة (الطبقية) المترابطة (Consistency Activities) والقيود (Constraints) والموارد (Resources) تهدف إلى توصيف (Specifying) وتصميم (Designing)

وتنفيذ (Implementing) واختبار (Testing) وصيانة وترقية (Maintenance and Evolution) منتج برمجي يفي بجميع متطلبات العميل.

جودة البرمجية Quality Assurance:

مصطلح يتعلق بعملية تطوير المنتج البرمجي والطريقة التي تم تطويره بها بحيث تكون وفقا لمعايير مقاييس الجودة والتي يتم وضعها من خلال مجموعة مراقبة الجودة من فريق التطوير بعد تحديد متطلبات المشروع البرمجي مباشرة وقبل البدء في التنفيذ الفعلي للمشروع.

التجزئة Modularity:

تقسيم النظام البرمجي المعقد إلى أقسام أبسط تسمى كتل.

التجريد Abstraction :

تحديد المظاهر المهمة وتجاهل تفاصيلها أثناء دراسة مسألة معينة (فصل الشيء المهم عن الشيء غير المهم في المسألة) ، وهي نظرة نسبية مرتبطة بالهدف الذي نعمل عليه.

توقع التغيير Anticipation of Change :

إحصاء الأماكن التي يمكن أن يتم تعديل عليها في البرمجيات.

إعادة الاستخدام Reuse :

يشير هذا المصطلح إلى تطوير برمجية معتمده على استخدام أجزاء جاهزة تم إعدادها وفقا لمعايير الجودة حيث تم اختبارها من قبل مما يزيد من إنتاجية المشروع وتوفير المزيد من الوقت. ويمكن النظر لهذا المصطلح - أيضاً - على أنه استخدام ما هو متاح فعليا لتحقيق ما هو مطلوب.

محلولو النظم System Analysts:

وهم الأشخاص القائمون علي دراسة وتحليل متطلبات النظام ومدخلاته ومخرجاته ، وكذلك تحديد الموارد اللازمة لتنفيذه ، بالإضافة إلى بيان كيفية التنفيذ وشرح ديناميكية العمل وتنظيم العلاقات المختلفة بين الكائنات الموجودة بالنظام .

معناه بالعربية	المصطلح بالإنجليزية
التجريد	Abstraction
توقع التغيير	Anticipation of Change
البرمجيات التطبيقية	Application Software
برمجيات الذكاء الاصطناعي	Artificial Intelligence Software
الخصائص المميزة للبرمجيات	Attributes of Software
التوثيق الأفضل	Better Documentation
هندسة البرمجيات بمساعدة الحاسب	CASE
نظم تغيير التحكم	Change Control Systems
أدوات إدارة التغيير	Change Management Tools
العميل وصاحب العمل	Client And Employer
مولدات الشفرة	Code Generators
الاختصاص	Competence
المتجمات	Compilers
إساءة استخدام أجهزة الحاسب الآلي	Computer Misuse

معناه بالعربية

علوم الحاسبات الآلية

السرية

أدوات إدارة التكوين

الأنشطة المرتبة (الطبقية) المترابطة

الصحة

أدوات دورة حياة التطوير المساعدة المتقاطعة

مولدات التداخل

نظم التداخل

قواميس البيانات

أدوات اكتشاف وتصحيح الأخطاء

الموثوقية

محررات التصميم

المطورون

محررات المخططات

أدوات التوثيق

المحلات الديناميكية

أدوات التحرير

الكفاءة

البرمجيات المدمجة بالأجهزة

البرمجيات العلمية والهندسية

أدوات التقويم

مقارنات الملفات

برمجيات الألعاب

العمومية

محررات الصور

تنفيذ

الدقة المتزايدة

السرعة الفائقة

برمجيات نظم المعلومات

المصطلح بالإنجليزية

Computer Sciences

Confidentiality

Configuration Management Tools

Consistency Activities

Correctness

Cross Life Cycle Case Tools

Cross reference Generators

Cross-reference Systems

Data Dictionaries

Debugging Tools

Dependability

Design Editors

Developers

Diagram Editors

Documentation Tools

Dynamic Analyzers

Editing Tools

Efficiency

Embedded Software

Engineering and Scientific

Software

Estimation Tools

File Comparators

Games Software

Generality

Image Editors

Implementing

Increased Accuracy

Increased Speed

Information Systems Software

المصطلح بالإنجليزية	معناه بالعربية
Intangible Benefits	الفوائد غير الملموسة
Intellectual Property Rights	حقوق الملكية الفكرية
Interactive Debugging Systems	نظم التصحيح التفاعلية
Interpretability	إمكانية تشغيلها مع برمجيات أخرى
Interpreters	المفسرات
Judgment	المحاكمة العقلية
Key challenges facing Software Engineering	التحديات الرئيسية التي تواجه هندسة البرمجيات
Language Processing Tools	أدوات معالجة اللغات
Learning Curve	منحنى التعليم
Lower Case Tools	أدوات التطوير المساعدة منخفضة المستوى
Maintainability and Evolubility	سهولة الصيانة وقابلية التطوير
Maintenance and Evolution	صيانة وترقية
Management	الإدارة
Method Support Tools	أدوات دعم الطرق
Modularity	التجزئة
Office Software	البرمجيات المكتبية
Page Layout Programs	برامج شكل الصفحة
PERT Tools	أدوات بيرت
Portability	إمكانية الحمل أو النقل
Product	المنتج
Profession	المهنة
Professional and Ethical Responsibility for Software Engineers	المسؤولية المهنية والأخلاقية لمهندسي
Program Analysis Tools	مسؤوليات مهندسي البرمجيات
Program Restructuring Systems	أدوات تحليل البرامج
Programmers	نظم برامج إعادة الهيكلة
Programming Career	المبرمجون
Programming in the Hands of Non-Programmers	مهنة البرمجة
	البرمجة في أيدي غير المبرمجين

معناه بالعربية	المصطلح بالإنجليزية
أدوات النمذجة الأولية	Prototyping Tools
الجمهور	Public
جودة الأداء	Quality Assurance
برمجيات نظم الوقت الحقيقي	Real – Time Systems Software
تخفيض الوقت اللازم للصيانة	Reduced Lifetime Maintenance
برمجيات إعادة الهندسة	Re-engineering Programs
أدوات إعادة الهندسة	Re-engineering Tools
الاعتمادية	Reliability
أدوات تتبع المتطلبات	Requirements Traceability Tools
أدوات تتبع المتطلبات	Requirements Traceability Tools
الموارد	Resources
إعادة الاستخدام	Reusability
إعادة الاستخدام	Reuse
الدقة والصورية	Rigor And Formality
فصل الاهتمامات	Separation of Concern
هندسة البرمجيات	Software Engineering
المهام النموذجية لهندسة البرمجيات	Software Engineering Tasks
صناعة البرمجيات	Software Industry (Development)
أطوار دورة حياة تطوير البرمجيات	Software Life Cycle
عمليات البرمجيات	Software Processes
توصيف	Specifying
المحلات الإستاتيكية	Static Analyzers
محللو النظم	System Analysts
أدوات بناء النظم	System Building Tools
مصممو النظم	System Designers
هندسة النظم	System Engineering
برمجيات النظام	System Software
مولدات اختبار البيانات	Test Data Generators
إمكانية اختبارها	Testability
اختبار	Testing

المصطلح بالإنجليزية	معناه بالعربية
Testing Tools	أدوات الاختبار
The Delivery Challenge	تحدي المخرجات
The Heterogeneity Challenge	تحديد عدم التجانس
The Heterogeneity Challenge	تحديد عدم التجانس
The Legacy Challenge	التحدي المتعلق بالموروث
Timeliness	ضبط وقت
Tool Mix	مزج الأدوات
Upper Case Tools	أدوات التطوير المساعدة عالية المستوى
Usability or User Friendliness	قابلية الاستخدام أو سهولة التعامل
User Interface Generators	مولدات واجهات المستخدم
Version Management Systems	نظم إدارة الإصدار

المراجع

أولاً : المراجع العربية :

- [1] روجر بريسمان ، "هندسة البرمجيات" ، الدار العربية للعلوم ، مركز التعريب والبرمجة ، الطبعة الأولى ، ١٤٢٥ هـ - ٢٠٠٤ م.

[2] مهندس إياد هلال ، "هندسة البرمجيات" ، المركز الألماني السوري لأعمال الانترنت (gsibc.net) ، منشور على الموقع :

<http://www.w3.org>

[3] أسماء المنقوش ، "دورة هندسة البرمجيات" ، منشور على الموقع :

<http://www.c4arab.com/>

المرجع الأساسي لهذه الدورة هو :

Shari Pfleeger, "Software Engineering - Theory and Practice", 2nd Edition

[4] مهندس عبد الحميد بسيوني ، "أساسيات هندسة البرمجيات" ، دار الكتب العلمية للنشر والتوزيع ، القاهرة ، ٢٠٠٥ م.

ثانياً : المراجع الإنجليزية :

[5] Ian Somerville , "Software Engineering", Addison Wesley, 2001.

[٦] Ronald J. Leach,, "Introduction to Software Engineering", CRC Press, 1999.

[٧] Douglas Bell , "Software Engineering A Programming Approach", 3rd Edition, Addison Wesley.

[8] Shari Pfleeger, "Software Engineering - Theory and Practice", 2nd Edition.

ثالثاً : مواقع على شبكة الإنترنت تم الاستفادة منها :

[٩] www.qucis.queensu.ca/Software-Engineering/

[١٠] www.c4arab.com/

[١١] <http://forum.amrkhaled.net/>

[١٢] <http://ar.wikipedia.org/wiki>

[١٣] <http://www.uop.edu.jo/Arabic/Faculties/>

[١٤] <http://cs.wwc.edu/>

[١٥] <http://www.pcwebopedia.com/>

-