مدى استخدام منهجيات وادوات هندسة البرمجيات في انجاز مشاريع التخرج لطلبة أقسام الحاسوب بجامعة الزاوية

دراسة ميدانيه

أ. عبدالرحمن ابراهيم العربي
قسم الحاسوب – كلية العلوم الزاوية
جامعة الزاوية

الملخص:

نظرا لأهمية هندسة البرمجيات وتطبيقاتها في المشاريع البرمجية الناجحة حيث نجاح أي منتوج برمجي يعتمد على تطبيق لمفاهيم هندسة البرمجيات. وفشل جل المشاريع البرمجية ناتج عن القصور في استخدام مفاهيم هندسة البرمجيات وباعتبار الباحث يقوم بتدريس مقررة هندسة البرمجيات منذ سنوات قرر القيام بدراسة بحثيه تتمثل في مدى تطبيق مفاهيم هذا المقرر عمليا

للطالب من خلال انجازهم لمشروع برمجي في مرحلة التخرج فقام الباحث بدراسة ميدانيه وذلك بوضع نموذج استبيان وتعبئته وبرنامج احصائي لتفريغ النتائج بعد التدقيق والبحث في معظم هذه المشاريع من فترة تدريس هذا المقرر بقسام الحاسوب وتعبأتها من خلال مشاريع برمجيه قام بها طلاب أقسام الحاسوب خلال سنوات سابقه ومشاريع حالية وذلك بالتدقيق والمرور على تفاصيل هذه المشاريع التي ينطلق منها الخريج كمبرمج جيد وتم استخدام منهجية العمل الميداني للبحث والتدقيق من الباحث وبعد المرور عينات كبيره من مشاريع تخرج للطلبة بأقسام الحاسوب قام الباحث بتفريغ النماذج التي عبت من هذه المشاريع في نموذج احصائي يشمل نسب مئوية والحصول عل نتائج خاصة بهذا البحث وبناءً علي النتائج التي توصل اليها الباحث سواء كانت اجابيه أو سلبيه و التي تمثل ان 80% من مشاريع التخرج لا تستخدم هندسة البرمجيات و 17% تشير الى بعض المنهجيات وفقط 3% استخدمت هذه المنهجيات بشكل جيد.

وهنا كان الهدف هو اتخاذ سبل بديله تساعد الطالب في تطبيق كم معين من مفاهيم هندسة البرمجيات والحصول على الهدف المنشود وهو برمجيات ذات جودة عالية وبتكلفه محدودة وتتمتع بخصائص البرمجيات الجيدة .

عليه قام الباحث بوضع حلول للنتائج السلبية وكتابة بعض التوصيات لغرض تعليم الطالب كيف يتعامل مع مواكبة التطور العلمي بأساليب هندسة البرمجيات المتاحة سواء في المشاريع الواقعية او المشاريع البرمجية التي ينجزها خرجي اقسام علوم الحاسوب وتحقيق الاهداف الاساسية لهندسة البرمجيات للوصول الى برمجيات ذات كفاءه وجوده عالية

وهي الحصول عل برمجيات ذات جودة عالية وبتكلفة محدودة و بزمن مناسب وباقل مخاطر تكون لها استفادة مستقبليه وتفعيلها على الواقع وليس ان يناقش الطالب المشروع وينتهي الموضوع بل جعل مشروع التخرج بمواصفات عالية تمكن الطالب من التنافس به وإدخاله في سوق

العمل وان ترجع عوائد هذه المشاريع بالربح للمؤسسة التعليمية وان يكون ابتكار علمي مفيد للطالب والمشرف وان يشارك به في البحوث والابتكارات العلمية.

الكلمات المفتاحية : هندسة البرمجيات ، جودة البرمجيات ، مشاريع برمجية ، تكاليف البرمجيات لغة النمذجة الموحدة ، مهندس البرمجيات ، جدولة المشاريع البرمجية ، مشاريع تخرج

مقدمة:

1-مقدمه في هندسة البرمجيات ومنهاجياتها وأدواتها:

1-1 تعريف هندسة البرمجيات

هي فرع من فروع علم الحاسوب يهتم بكل أشكال إنتاج البرمجيات الجيدة ذات الجودة العالية وبتكلفه محدودة سهلة الاستخدام والصيانة ، ويجب أن يتبنى مهندس البرمجيات الاقتراب المنظم و النظامي في عمله، ويستخدم منهجيات والأدوات المناسبة والتقنيات الضرورية بدءاً من المراحل المبكرة لخصائص النظام حتى صيانة هذا النظام بعد استخدامه.

1-2 أهداف وأسس هندسة البرمجيات

التحسين في جودة البرمجيات وزيادة الإنتاجية في إعداد البرمجيات. الموثوقية وسهولة الاستخدام وقابلية الصيانة والتحديث وتحسين طرق الاتصال بين المستخدم و معد البرنامج والتقليل من تكلفة المنتجات البرمجية و الأسس تعتمد هندسة البرمجيات على معرفة المجالات التالية: علم الحاسوب – علم الإدارة – مهارات الاتصال – الأسلوب الهندسي – علم الاقتصاد.

1-3 أدوات هندسة البرمجيات

الطرق والأدوات الآتية يتم استخدامها من قبل مهندس البرمجيات لإعداد البرمجيات:

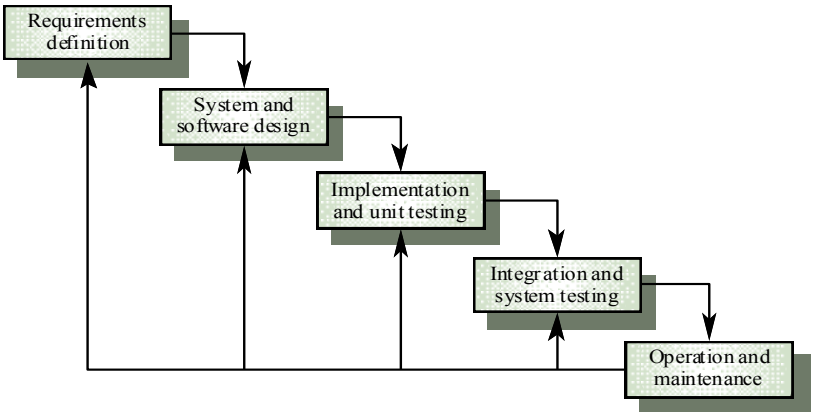
- أدوات التحليل لتحديد متطلبات النظام. مثل CASE tools و ادوات غانت و أدوات التصميم لتمثيل الحل.
- أدوات ومفاهيم الإدارة لتخطيط زمن متابعة المشاريع.
- إجراءات صيانة البرمجيات.
- الادوات الآلية لزيادة جودة البرمجيات وإنتاجية المبرمج والتحكم الإداري للمشاريع البرمجية.
- ادوات قياس إجراءات اعتماد الجودة.
- طرق لتقدير التكلفة ومدى الاعتماد على المنتوج.
- استخدام المكونات الجاهزة
- استحداث منهجيات جديدة لإعداد المنظومات لغرض تقليل الوقت والتكلفة و اعادة هندسة البرمجيات

4-1 نماذج عمليات البرمجيات Software process model

هي تمثيل مجرد لوصف العملية من منظور معين، ومن النماذج العامة لعمليات البرمجيات:

1) نموذج الشلال The waterfall model :

عبارة عن مراحل واضحة المعالم منفصلة لوضع المواصفات والتطوير.

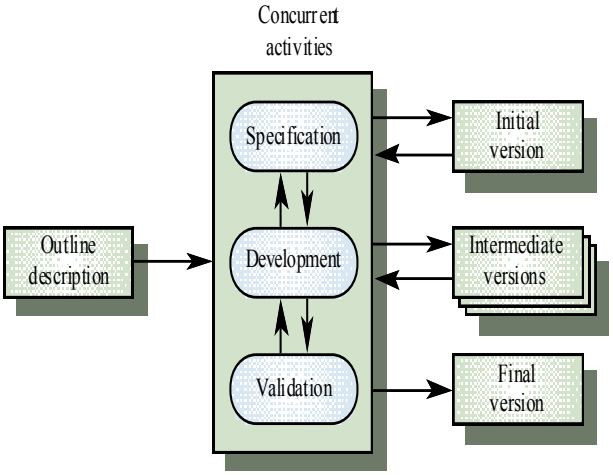


(شكل1 نموذج شلال المياه)

وقد تختلف هذه المراحل من مطور الى اخر ومن نظام الى اخر وقد تتجزأ او تدمج بعض المراحل ولاكن لا تخرج عن هذا النطاق و من عيوب نموذج الشلال : صعوبة تقدير وتكييف التغييرات أثناء العملية. و التقسيم غير المرن للمشروع إلى مراحل منفصلة يزيد من صعوبة الاستجابة عن تغيير متطلبات المستهلك،

2) التطوير الإرتقائى Evolutionary development

التطوير الارتقائي هو نموذج تطوير لإنتاج البرمجيات تتداخل فيه المواصفات مع التطوير. والهدف هو العمل مع المستهلك (الزبون) والتوصل إلى النظام النهائي من خلال خطوط عامة تمهيدية للمواصفات ويجب البدء بمتطلبات مفهومة جيداً.



(شكل2 نموذج التطوير الارتقائي)

- في التطوير الارتقائي يتم وضع خطوط وصف عامة ثم تتم متابعة العمل بأنشطة متلاقية مساعدة لتحديد المواصفات التي تعطى إصداراً أولياً بمكن أن يتفاعل مرة أخرى مع تحديد المواصفات لتغييرها وتعديلها وهكذا، ثم من تحديد المواصفات تبدأ أعمال التطوير التي توفر إصدارات وسيطة تتفاعل بدورها مع التطوير وتؤثر فيه وتتأثر به، ثم يتم التحقق من النظام للوصول إلى الإصدار النهائي.

تتفاعل أيضاً مراحل تحديد المواصفات والتطوير والتحقق مع بعضها البعض.

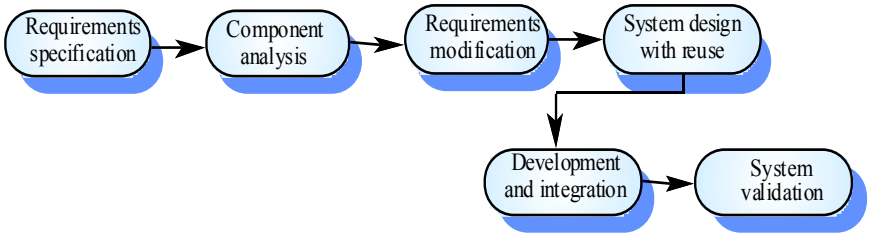
- من مشاكل وعيوب التطوير الارتقائي : نقص وضوح العمليات، فقر هيكلية النظم ، وتحتاج مهارات خاصة بلغات تسمح بالنماذج الأولية السريعة.

من مزايا التطوير الارتقائي: سهولة تعامله للنظم الصغيرة والمتوسطة الحجم ذات التفاعلية، وبعض الأجزاء من النظم الكبيرة مثل واجهة المستخدم

3) التطوير المعتمد على إعادة الاستخدام Reuse-based development :

بتجميع النظام من مكونات موجودة.

يعتمد على إعادة الاستخدام التقليدي حيث يتم تجميع النظام من عدة مكونات موجودة أو من نظم أخرى (Commercial-off-the-shelf).



(شكل 3 نموذج المعتمد على اعادة الاستخدام)

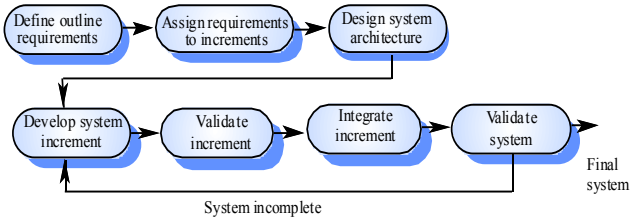
تمتاز هذه المنهجية بانها اصبحت الاكثر اهمية لاكن من عيوبها ان الخبرة فيها قليلة و محدودة

5-1 تكرار العمليات Process iteration

قد يتم التكرار المتتالي في أي عملية أو في أي نموذج وهناك منهجيتان للتكرار هما:

1-5-1 التطوير المتزايد: Incremental development

بدلا من التوصل إعداد النظام مرة واحدة يقسم التطوير والتسليم إلى أجزاء متزايدة يقوم كل منها بتوفير وظيفة مطلوبة، وتكون لمتطلبات المستخدمين أعلى أولوية، وتوضع المتطلبات ذات الأولوية الأعلى في الأجزاء المبكرة من النظام. ما ان يبدا تطوير جزء حتى يبدأ تجميد هذه المتطلبات حتى يمكن الاستمرار في الجزء التالي



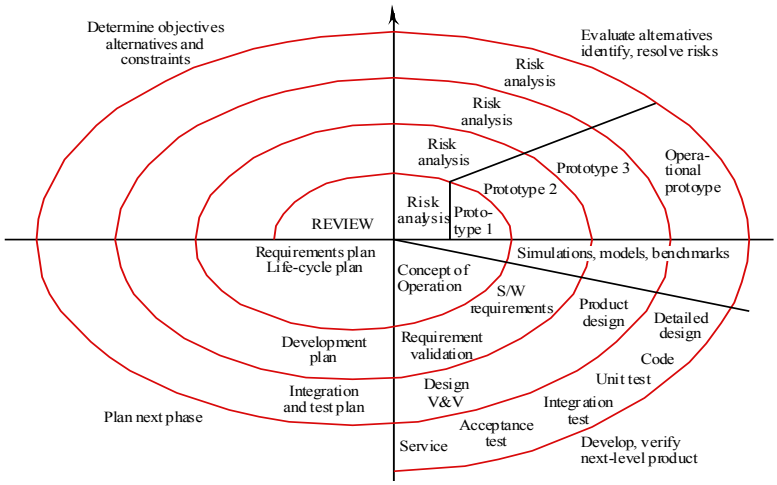
(شكل 4 نموذج التطوير المتزايد)

ويمتاز التطوير المتزايد يمكن لتقدير الزبون ان يوضع في الاعتبار مع كل جزء وبهذا تتاح وظيفة النظام كما تعمل الاجزاء المبكرة من النظام كنماذج اولية تساهم في تحديد دقة للمتطلبات للمراحل التالية ويمتاز النظام بقلة المخاطر في الانهيار الكلي للمشروع و تكون لخدمات النظام ذات الاولوية الاعلى تنال الكثير من الفحص والاختبارات

كذلك استخدام (البرمجة السريعة)Extreme programming هي منهجية تعتمد على التطوير والتسليم لمكونات وأجزاء صغيرة جداً من وظائف النظام بالطريقة المتزايدة وتعتمد على تحسين الشفرة الثابتة ، وتضمين الزبون فى فريق التطوير

1-5-2 التطوير الحلزوني Spiral development

يمثل تطوير العمليات على هيئة لولبية بدلا من تتابع متتال للأنشطة أو التتابع المتتالي مع الرجوع عكسياً لمزيد من التحسين، وتمثل كل حلقة من الحلزون مرحلة من مراحل العملية، و لا توجد مراحل ثابتة مثل توصيف المتطلبات أو التصميم، ويتم اختيار الحلقات في الحلزون بناء على ما هو مطلوب ، ويتم تقدير المخاطر Risks وتحليلها وحل أمورها خلال العمليات.



(شكل 5 نموذج التطوير الحلزوني)

اما من اساليب هندسة البرمجيات ما يعرف بلغة النمذجة الموحدةUML والتي توفر عدة نماذج مختلفة لوصف النظام. القائمة التالية تعرضها كلها مع جملة واحدة توجز الغرض من كل نموذج:

وقائع الاستخدامUse Cases – "كيف سيتفاعل نظامنا مع العالم الخارجي؟"

مخطط الصنفيات Class Diagram "ما هي الكائنات التي نحتاجها؟ وما علاقتها؟"

مخطط التعاون Collaboration Diagram "كيف تتعامل الكائنات مع بعض؟"

مخطط التتابع Sequence Diagram "كيف تتعامل الكائنات مع بعض؟"

مخطط الحالة State Diagram "ما الحالات التي يجب أن تكون عليها الكائنات؟"

مخطط التحزيم Package Diagram– "كيف سنقوم بقولبة عملنا؟"

مخطط المكونات Component Diagram – "كيف سترتبط مكونات برنامجنا؟"

مخطط التجهيز Deployment Diagram – "كيف سيتم تجهيز البرنامج؟"

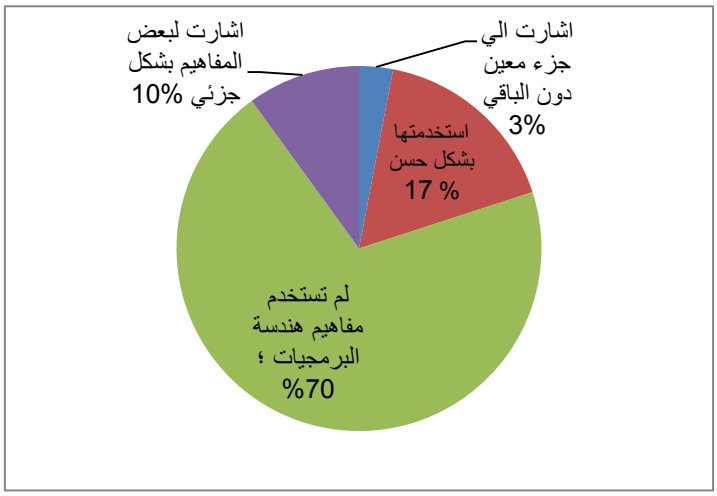
كذلك تهتم هندسة البرمجيات بادرة النسخ والتنوعات والاصدارات و إعادة هندسة النظام وإدارة تكوين البرمجيات ومقاييس الجودة وموثوقية البرمجيات واحتساب التكاليف

خلاصة نتائج الدراسة و جمع المعلومات :

بعد المرور على اكثر عدد ممكن من مشاريع التخرج لطلاب اقسام الحاسوب و جمع الاحصائيات من حيث مدى استخدام مفاهيم هندسة البرمجيات في المشاريع البرمجية من عدمه توصل الباحث الى :

نسبة الاستخدام 17% من المشاريع استخدمتها بشكل حسن و 10 % اشارت لبعض المفاهيم بشكل جزئي غير مفصل اما 3% كانت اشارت الي جزء معين دون الباقي اما 70% لم

تستخدم مفاهيم هندسة البرمجيات برغم من حداثتها وكانت تعتمد على مقرر تحليل وتصميم النظم لمراحل المشروع



شكل (6) مقطع احصائي لنتائج الدراسة

كما لاحظ الباحث بعض التشابه الكبير في المشاريع باستخدام نسق النموذج الانحداري أو ما يعرف بنموذج شلال المياه الذي يتعلمه الطالب في مقرر تحليل وتصميم النظم، بينما المشروع يفضل استخدام نموذج تطوري من نماذج هندسة البرمجيات مثلا التطوير بإعادة الاستخدام او النموذج الارتقائي بما يتناسب مع طبيعة النظام .

كما أن الجداول الزمنية لا توجد في بعض المشاريع و الافضل استخدام ادارة المشاريع البرمجية ولوحظ من خلال البحث عدم الاهتمام بالجود والتكلفة.

أما ما يخص الكود البرمجي نسبه كبيره لا يوجد توثيق لحقول والمتغيرات البرمجية واحيانا يكتب الكود البرمجي بلغة غير مناسبه او عدم استخدام نوع مناسب من قواعد البيانات مما يعكس اثار سلبيه على المشروع.

بعض المشاريع تعاني من اخطاء ناتجه عن الفهم الخاطئ لمتطلبات النظام أي الطالب في مرحلة التحليل يتحصل على معلومات غير صحيحة من اشخاص يمثلون النظام والطالب لا يتحقق من ما جمع من معلومات او يكون استخدم اسلوب غير مناسب في جمع المعلومات

التوصيات:

أولاً اوصي بتدريس مقرر هندسة البرمجيات لصيق بمشروع التخرج والاهتمام باستخدام مفاهيم هندسة البرمجيات السالفة الذكر طوال مراحل المشروع البرمجي المذكورة في الاستبيان من بداية المشروع حتي اكتماله مستخدمين كل ما تم ذكره والاعتماد على لغة النمذجة الموحدة مما يسهل في فهم المشروع و تحويله من لغة برمجة الى لغة برمجة اخرى واستخدام مفاهيم هندسة البرمجيات مثل مفهوم الوراثة و اعادة الاستخدام من اجل الوصول الى برمجيات ذات جودة عالية وتكلفة محدودة معتمد عليها و قابلة للصيانة و سهلة الاستخدام موثقة بشكل جيدوالاهتمام بموضوع الجدولة الزمنية من حيث اختيار المسار الافضل كما انصح باستخدام نموذج تضخم العيوب defect amplification model لتوضيح نشوء الأخطاء وكشفها خلال مراحل التصميم الأولى والتصميم التفصيلي و اجراءات اكتشاف وملاحقة الأخطاء والاحتفاظ بالوثائق الورقية المرفقة للمشروع والتأكيد علي استخدام معطيات حقيقيه عند تنفيذ الكود والمشاركة في وضع مواصفات البرمجة ومراجعة نشاطات هندسة البرمجيات للتأكد من موافقتها مع تعريف عملية البرمجة تدقيق منتجات العمل المتعلقة بالبرمجية للتأكد من توافقها مع ما جرى تعريفه في عملية البرمجة و التحقق من توثيق ومعالجة الانحرافات في العمل البرمجي وضع وثائق بها نماذج UMLمتكاملة و الاعتماد علي ادارة المشاريع البرمجية للوصول للزمن والوقت الافضل من خلال المسار الحرج للمشروع.

التعامل مع مشاريع التخرج كأنها انتاج علمي والعمل بها و العمل عل جعل مكتبه الكترونيه تجمعهم لغرض الاستفادة منهم قدر المستطاع.

ملاحق البحث :

النماذج المستخدمة في جمع البيانات

تم تصميم نموذج خاص لكل مشروع تخرج مكتمل تمت مناقشته سابقا مع العلم ان جميع الطلاب قاموا بدراسة مقرر هندسة البرمجيات في الفصل السادس و تم المرور على أغلب مشاريع التخرج بقسم الحاسوب بكلية العلوم

نموذج يمثل معلومات عامة عن مشروع تخرج

اسم المشروع :.....

الكلية (.....)

اسماء طلبة المشروع :.....

اسم المشرف :.....

الفصل الدراسي : .....

اهداف المشروع : .....

هل المشروع قابل للتطوير : نعم لا

هل المشروع مستقل او تكمله لمشروع سابق :نعم لا

هل المشروع يحوي على كود برمجي مرفق : نعم لا

هل المشروع لمشكلة واقعيه : نعم لا

هل يوجد توثيق للحقول والمتغيرات البرمجية نعم لا

هل المشروع يواكب أخر التطورات من الكيان المادي نعم لا

هل المشروع له قاعدة بيانات مصممه بشكل جيد نعم لا

هل المشروع خالي من الاخطاء البرمجية نعم لا

هل المشروع استخدمت فيه تقنيات تطوير الويب نعم لا

أي ملاحظة اخري .....

نموذج تفصيلي لكل مشروع في شكل استبيان

نماذج تطور لهندسة البرمجيات	ملم	غير ملم	حالة وسط حسب طبيعة المشروع	ملاحظة
1)) نموذج الشلال				
2)) التطويرالارتقائي				
3)) التطوير المعتمد على إعادة الاستخدام				
4)) نموذج التطوير المنهجي الاساسي				
5)) نماذج قديمة مثل النموذج الانحداري				

نماذج تكرار العمليات	ملم	غير ملم	حالة وسط حسب طبيعة المشروع	ملاحظة
التطوير المتزايد:				
– التطوير الحلزوني				
أي نوع اخر من التطوير				
هندسة البرمجيات بمساعدة الكمبيوتر CASE :				
البرمجة السريعة				
استخدام مفاهيم UML	ملم	غير ملم	حالة وسط	ملاحظة
وقائع الاستخدام Use Cases				
مخطط الصنفيات Class Diagram				
مخطط التعاون Collaboration Diagram				
مخطط التتابع Sequence Diagram				
مخطط الحالة State Diagram				
مخطط التحزيم Package Diagram				
مخطط المكونات Component Diagram				
مخطط النشر Deployment Diagram				

استخدام الخطط التالية	ملم	غير ملم	حالة وسط حسب طبيعة المشروع	ملاحظة
خطة ادارة المشاريع البرمجية Project Management				
خطة ادارة الجودة Quality Management				
ادارة الاشخاص Managing People				
ادارة المخاطر The risk management				
هل يوجد جدوله للمشروع Project scheduling				
هل يوجد الخط الزمني لأنشطة المشروع Activity timeline				
هل تم تطبيق المخططات البيانية و شبكة الانشطة Bar charts and activity networks				
خطة التحقق Validation plan				
خطة ادارة التكوينconfiguration plan				
خطةالصيانهmaintenance plan				

مراجع البحث:

1- *Somerville, Ian. Software Engineering, Addison-Wesley, 8th Edition,(2007).*

2- *Linda Rosenberg, Ted Hammer, Jack Shaw. Software Metrics (2010)*

3- *Lan K. Bray Soren Lauesen Software Requirements Styles and Techniques , Addison Wesley 2010 An introduction To Requirements Engineering , (2012)*

4- *Bernd Oestereich Developing , Software with UML – Object-oriented analysis and design in practice, Addison Wesley Professional.,*

5- *Perdita Stevens with Rob Poole Using UML. Software Engineering with Objects and Components (Updated Edition, Addison-Wesley 1st edition (April 1, 2009)*