دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الباب الأول

المقـــــدمـــــــــة

1-1 مقدمـــــة :

أستخدمت هندسة البرمجيات كمفهوم نظرى من حين لاخر فى اوخر الخمسينات من القرن الماضى اما الاستخدام الاول لهذا المصطلح فكان فى موتمر عقد من قبل اللجنة العلمية فى منظمة حلف شمال الاطلسى 1968 حول البرمجيات وقد اخذ هذا المصطلح بالانتشار من ذلك الحين ولاقى اهتماما متزيدا فى نواحى مختلفة .

وقد ظهرت هندسة البرمجيات بسبب عدم استخدام منهجية فى التفكير عند بناء البرمجيات مما ادى الى ظهور اخطاء كثيرة خلال بناء وصيانة البرمجيات بالتالى اصبحت تحتاج الى وقت كبير لتطويرها ولصيانتها وكلفة مالية عالية .

لذلك ظهرت هندسة البرمجيات لوضع اسس وقواعد تهدف الى تصميم وتطوير البرامج بجودة عالية تلبى احتياجات المستخدمين وامكانية وضع اسس لتميز البرامج الجيدة .

فالتغيرات التكنلوجيا والتقنية المستخدمة فى الانظمة المالية تحتاج فى الغالب الى تطوير الانظمة القديمة الى بيئة اكثر مرونة تسمح بردود فعل سريع لاحتياجات العملاء لذكك ينظر الى الجانب التقنى حيث هندسة البرمجيات تتعامل مع المصطلحات المالية والادارية المتخصصة بلادوار والاحتياجات وتودى الى تطوير الانظمة وفق لاسس وقواعد مدروسة وتقيم تلك البرمجيات وفقا لتلك الاسس ماخذة فى الاعتبار القيود المالية والبرمجية .

ان التطوير المستمر قد اصبح اهم المواصفات التى يجب توافرها فى البرامج المالية فالذى لا يعد نفسه اعداد كافيا لمقابلة تلك التغيرات التى تحدث بشكل دائم ومستمر فى سبيل التحسين يصبح مثل سيارة عتيقة وسط سباق تنافس به احدث السيارات .

لقد بدأت الادارة العامة للشئون المالية بوزارة الدفاع فى مواكبة هذا التطور فى عام 1997 م وذلك بحوسبة مجموعة كبيرة من انظمة العمل المالى واعداد البنيات التحتيه له والكوادر الموهلة لذلك العمل .

ففى العام 1997 م كانت البداية بحوسبة الفصل الثانى وحققت التجربة نجاحا مقدرا وكبيرا فى تنفيذ الاهداف المخطط لها والمتمثلة فى سرعة الاجراءات وتامين المعلومات وسلامتها واستخراج البيانات المطلوبة بسهولة ويسر .

وكانت التجربة الثانية فى العام 1999 م بانشاء مكتب الرصد والمتابعة لادخال بيانات الفصل الاول مع سجل الافراد وحقق البرامج اهدافه فى ضبط المرتبات ومحاصرة المخالفات المالية وحصر القوة .

المرحلة الثالثة كانت مرتبات الضباط ثم بعد برامج الميزانية ثم برامج المعاشات ثم بعد ذلك تم حوسبت معظم العمل المالى واخره هو الدفع الالترونى لمرتبات ضباط الصف والجنود .

لكن التغير المستمر فى هيكلة وتطوير القوات المسلحة الى جانب التطوير المستمر من وزارة المالية فى جانب تحسين الاداء المالى فى الدولة جعل لزاما على الادارة العامة للشئون المالية مواكبة التغيرات وتطوير الانظمة المالية له وتحسين اداء الرقابة المالية.

ولكن وبالرغم من إيجابيات التغير فى الهيكلة انه هناك تحديات و محددات جديدة على نظام المعلومات المحاسبي والتي تؤثر على فاعليته وكفاءته ومن أهم هذه المحددات ما يلي:

1. دخول معايير جديدة لتقيم البرمجيات لتحقيق الامان والتوكيدية والموثوقية.
2. ترابط البرامج المالية الحالية عبر الشبكة بعد ان حولها الهيكل الجديد الى عدد من البرامج التى يتم ترابطها عبر الشبكة .
3. تطوير الانظمة الموروثه .

يعرف النظام القديم بانه نظام البرمجيات التي تستخدم تقنيات ولغات كمبيوتر سابقة في حين لايزال في الخدمة وحساس في العمليات اليومية في المنظمة ' التعامل مع مثل هذا النظام يواجه العديد من التحديات علي سبيل المثال وليس الحصر الكلفه الماليه العاليه في تطوير النظام الاول وعدم كفاية الوثائق للصيانة وعدم وجود برامج وادوات دعم.

معظم النظم القديمة تلعب دور هاماً في بيئات الانتاج وتحقق النتائج المتوقعة بنجاح ومع ذلك من الصعب صيانة الانظمة الموروثه. النظام الموروث في كثير من الأحيان قد شهد الكثير من حالات إصلاح الخلل ' والتحدثيات والتعديلات علي طول دورة الحياة . هيكل النظام تدريجياً قد تعرض للتشوية مقارنة مع التصميم الأصلي وبالتالي أصبحت الوثائق الأصلية التي عفا عليها الزمن . كما الفجوة بين الكود والوثائق التي لم يتم تحديثها فتصبح الصيانة من الصعب ومكلفة [9] علي نحو متزايد .

النظم الموروثه التي تم دمجها مع أنظمة أخري والصيانة له تحدث عقبات ايضاً بسبب التوافق والتواصل مع الانظمة الأخري علي الرغم من أن الانظمة الوسيطة(middleware) المتطورة

للمساعدة في التكيف مع الانظمة القديمة مع البيئة الجديدة. لانه يزيد من تعقيد النظام ويتطلب بذل جهد أضافي ليتم صيانتها.

اعادة هندسة البرمجيات هو المكون الرئيسي في تطوير البرمجيات يهدف الي تعزيز القدرة لانظمة البرمجيات على التكيف والصيانة ويشير الىى عملية تصميم وتنفيذ الهياكل الجديدة لنظام البرامج الموجودة . لحل قضايا التكيف والصيانة دون أهمية لتعديل الوظائف الداخلية.

جميع الادوات المستخدمة في مجالات هندسة البرمجيات والاكثر شيوعاً تطبق النمذجة الكائنيه (object – oriented modeling) "OOM" واكثر الاطر شعبيه مثل تصميم المكونات الاساسية للJ22EE و .Net علي اساس OOM أصلاً انشئت لتزيل العقبات في البرمجة الهيكلية والنمذجة في نطاق المشكلة واعادة الاستخدام.

نموذج الكائنى المنحى يحاكي سلوك الانسان في العالم الحقيقي بالتالي يتم نمذجة المشكلة بكفاءة مثل مشكلة الانسان. هذا يفترض أن مجال المشكلة يحتوي علي عدد من الكائنات وكل كائن له سلوكيات وصفات مختلفة [10] علي سبيل المثال أفترض أن الكمبيوتر المحول هو كائن السمات يمكن أن تكون العلامة التجارية وتعريفات الجهاز والسلوكيات للكمبيوتر المحمول يمكن أن تكون يحسب ويعمل.

من اجل تحقيق مرونة أفضل والتوافق مع التكنولوجيا الناشئة وغالباً ماتحتاج النظم القديمة الي تحويله الي انظمة الكائنى المنحى تقليدياً الهندسة العكسية و نموذج الكائنى المنحى والتقنيات الرئيسية المستخدمة في اعادة هندسة الانظمة القديمة .

وقد تمت دراسات علي نطاق واسع لتكنولوجيا الكائنى المنحى موخراً من قبل المهندسين للمساعدة في اعادة بناء OOD بناء علي OOA الموجودة [11] ومع ذلك نادراً ماتم اختبار تقنية علي نظم مالية محددة غير كائنية هذه الرسالة تستكشف تطبيق تقنيات الكائنى المنحى لاعادة بناء نظام غير كائنى ونناقش نتائج التجريب .

2-1 مشكلة البحث:

من المعروف ان التغير فى هيكلة الانظمة وتعديلها وتطوير منهجيات تقيمها يتطلب تطوير لهذه الانظمة لذلك التطور المتسارع فى البرمجيات والانظمة المالية وتغير هيكلة القوات المسلحة ووضع اسس جديدة لتحسين الاداء المالى من وزارة المالية افرز مجموعة من المشاكل تركز

هذه الدراسة علي اعادة هندسة النظم القديمة غير القائمة علي الغرضي التوجه لتحسين جودة النظام والتحديات التي تواجه هذه الدراسة هي :

1/ تكنولجيا الارتقاء بها Technology upgrading

2/ هجرة الوظائف functionality migration

3/ اعادة التقاط المتطلبات requirement re-capture

4/ تحسين الهيكل الغير كائني improving the non-oo structure

تطوير التكنولوجيا يشير الي اعتماد التقنيات الاساسية المتطورة للنظام القديم من اجل توفير اطار اكثر من حيث الفهم حيث يمكن للمهندسين التواصل والفهم وصيانة النظام باعلي كفاءة ومع ذلك في الوقت الحاضر التطوير سريع من اي وقت مضي في نتائج صناعة البرمجيات في وجود فجوة كبيرة بين التقنيات ا والتكنولوجيا القديمة والجديدة الغير موجودة في وثائق النظام القديم فى كل جانب من جوانب البرمجيات بما في ذلك جمع الاحتياجات وطرق التصميم ولغات البرمجة واجراء الاختبارات وتنسيق البيانات والوثائق والتعامل مع هذه الفجوة يتطلب فهم تكنلوجيا النظام القديم والجديدعلي حدا سواء . التقنيات القديمة تستخدم من قبل النظام الموروث وتحتاج الي جهد كبير تشير هجرة الوظائف الي تنفيذ نفس الوظائف التي يقدمها النظام القديم القائم علي تكنولوجيا منصة جديدة ' وتتطلب هذه العملية دقة وفهم الوظائف القائمة ولكن نظراً الي أن الوثائق غير موجودة أو غير كافية قديمة ' القناة الرئيسية لفهم النظام القديم هو عن طريق مراجعة الكود وتشغيل النظام يدوياً .

الفشل في تلبية متطلبات معينه هي واحدة من الاسباب لاعادة تصميم النظام الموروث علي سبيل المثال تنسيق البيانات القديم يجعل النظام غير قادر علي تلبية متطلبات التواصل مع الانظمة التي تستخدم تنسيق البيانات الجديد هذا الفشل يرتفع في الغالب من المتطلبات غير الوظيفية بدلاً من المتطلبات الوظيفية لان الاول هو ضمنى.

خلفية المستخدم ذات الصلة والتغير من الحين للاخر المتطلبات الوظيفيه مستقره نسبيا لالتقاط المتطلبات غير الوظيفية بكفاءة مطلوب من المطوريين مراقبة النظام من وجهة نظر المستخدم ' التي هي اكثر صعوبة كمايبدو ايضاً لاتوجد وسيلة منهجية للمساعدة والاستدلال الا بالاساليب البحثية.

يسبب الدعم المحدود لاعادة الكود اللاكائني فمن الصعب تحقيق نفس المرونة . التي يتم توفيرها من قبل لغات الكائنى المنحنى معظمها يبتبع مواصفات الكائنى المنحنى بالرغم من اختلاف بناء الجمل فى حين تتبع اللغات غير الكائنى المنحنى تختلف نماذج التصميم وليس هنالك طريقة منهجية وضعت النظام غير الكائني لمساعدة التحليل والتصميم في معظم الحالات الاداة الرئيسيه التي يتم استخدامها في اعادة هندسة النظام غير اغرضي التوجه هو تجارب المطوريين ومراجعة الكود.

3-1 أهداف البحث

الهدف الرئيسي من هذا البحث هو

1. اعادة هندسة انظمة البرمجيات القديمة .
2. ايجاد نموذج ربط من اجل تحسين الموثوقية والمرونة والصيانة

ولتحقيق هذا الهدف يتم تأسيس اربعة اهداف .

أ. تحليل النظام القديم

تحليل النظام القديم هو اكتشاف نشاة النظام القديم والتكنولوجيا ذات الصلة' والقضايا القائمة وودورها في سياق النظام . الفهم الجيد لتلك العوامل يمكن أن تساعد في اختيار المنهجية والادوات المناسبة في عملية اعادة الهندسة.

ب. استكشاف منهجيات الهندسة الكائنية

استكشاف معظم اجراءات هندسة البرمجيات المشتركة والمنهجيات لفهم استخدامها والسيناريوهات وتعطي الدراسة شاملة RUP واطار تطوير البرمجيات المتكررة .

ج. اقتراح منهجية مناسبة

اختيار منهجيات خاصة تستخدم منهجيات المنحى الكائنى المناسبة للمشاريع الصغيرة ' والانظمة المالية وبناء خطوات تطبيق لتحديد الهياكل الفرعية في غير الكائنى المنحنى الـــ RUP و نموذج الكائنى المنحنى تتكون من المنهجيات المختلفة التي لاتكون مناسبة لجميع انواع المشاريع ينبغي أن تكون مصممة خصيصاً لتناسب حجم ومتطلبات المشاريع لتحقيق لتجربه جيده للمشروع ' ويجب تطبيق التقنيات التي تمثل مجال المشكلة

4. اجراء دراسة حالة لتطبيق منهجيات اعادة هندسة النظام القديم " الانظمة المالية لادارة الشون المالية" واختباره لاظهار النتائج .

تحليل النظام القديم وسيتم تطبيق المنهجيات للتعرف علي قضايا نظام القديم وانشاء الحل وتنفيذ عملية اعادة الهندسة لتحسين جودة كل النظام .

5.بناء نموذج ربط لربط الانظمة عبر الوحدات والادارات المختلفه.

1-4 محددات البحث

يعتقد الباحث أنه من أهم محدداتها ما يلي:

1. حداثة الموضوع.
2. قلة البحوث المتعلقة بالموضوع.

1-5 أهمية البحث

تنبع أهمية الدراسة وبشكل رئيسي من أهمية الادارة العامة للشئون المالية بوزارة الدفاع اذ به كل ميزانيات القوات المسلحة والتى تعتبر نسبة كبيرة من ميزانية الدولة فلابد من التطوير واستخدام كل ادوات التكنلوجيا الحديثه والمتطوره واستخدام ادوات ربط امنه وموثوقة.

كما ان وجود نماذج واليات لتقييم البرمجيات المالية يجعله تواكب التطورات الحديثه فى مجال البرمجيات . ومن ثم الحفاظ على المال العام وبذلك المحافظة على الاقتصاد الوطنى .

1-6 **حدود البحث**

1. تتمثل الحدود المكانية للبحث فى وحدات الادارة العامة للشئون المالية بوزارة الدفاع
2. اما الحدود الزمانية فتتمثل فى حوسبة العمل المالى فى الفترة من 2012_ 2015م.

1-7 الدراسات السابقة

1. دراسة أعدتها فداء عباس بعنوان "دراسة تحليلية مقارنة بين الطرائق البنيوية والطرائق غرضية التوجه في تحليل وتصميم النظم" 2005.

هدفت الباحثة في دراستها إلى الوقوف على الأسس والمبادئ التي تستند إليها كل من الطرائق البنيوية والطرائق غرضية التوجه في آلية تحليل النظام، مع المقارنة بين هذه الطرائق، حيث خلصت إلى أن الفرق في استخدام الطرائق غرضية التوجه المتمثلة بلغة النمذجة الموحدة UML والطرائق البنيوية لتحليل وتصميم أي نظام ما هو إلا مجرد ارتياح وخبرة لا أكثر، فالمحلل (المصمم) قد يرتاح إلى استخدام الطرائق غرضية التوجه (UML) في حين محلل (مصمم) آخر قد يرتاح إلى استخدام المنهجية البنيوية في بناء نظامه.

2. دراسة أعدتها منى محمد بعنوان "مقارنة منهجية الـ UML مع بعض المنهجيات السائدة في تصميم نظام المعلومات" 2005.

لقد كان الهدف من هذه الدراسة إجراء مقارنة بين لغة النمذجة الموحدة UML وبعض المنهجيات السائدة في تحليل النظم، حيث عالجت هذا الموضوع من خلال شرح أدوات ومفاهيم هذه المنهجيات، وإجراء المقارنات بينها وبين UML مقدمة التوصيات والنصائح والمقترحات اعتماداً على نتائج المقارنات التي توصلت إليها.

3. دراسة أعدها علي حسين أحمد بعنوان "المطابقة بين النماذج المختلفة في منهجية CASE*Method لتحليل وتصميم النظم" 2002.

تأتي أهمية هذه الدراسة من كونها تسلط الضوء على إحدى منهجيات تحليل وتصميم النظم المطبقة عملياً (CASE*Method)، ولأنها توفر الإمكانية لتحديد مدى ملاءمة كل تقنية من التقنيات المتوفرة في منهجيات تحليل النظم لتطبيق معين أو مرحلة معينة من مراحل التطبيق، الأمر الذي سيساهم في زيادة الفائدة من هذه التقنيات وتحديد خطوات متسلسلة لعملية التدقيق المتبادل وفق النموذج الذي يتم الانطلاق منه.

4. دراسة أعدتها بيداء عبد الكريم الزير بعنوان "تحليل وتصميم النظم بالطرق غرضية التوجه"، 2000.

لقد كان الهدف من هذه الدراسة هو معرفة ودراسة المنهجية الغرضية التوجه في التحليل والتصميم ابتداءً من مفهوم النموذج الغرضي التوجه وصولاً إلى التحليل والتصميم الغرضي

التوجه OOA، OOD. توصلت الباحثة في دراستها إلى تحديد مزايا ومساوئ التصميم الغرضي التوجه وما هي أوجه التشابه والاختلاف بين المنهجية البنيوية والمنهجية الغرضية التوجه.

5. دراسة أعدها الباحث F. Stefani بعنوان "العمل على تحويل تحليل MERISE باتجاه غرضي التوجه" 1998

في إطار عمل المشروع الذي كان هدفه الرئيسي بناء تصميم لإعادة هندسة برامج الأنظمة الموروثة اعتمدت طريقة مبنية على القاعدة (Rule - Based) من أجل التحويل الترتيبي نحو غرضي التوجه لتحليل MERISE. بحيث تؤدي إلى إنجاز أداة نصف أوتوماتيكية لتحويل مخطط تحليل MERISE إلى مخطط تحليل غرضي التوجه. من خلال بناء عمليات تساعد في إعادة بناء القواعد المستخرجة من حالة العالم الحقيقي التي تقدمها الشركة التي ترغب بالانتقال إلى التقنيات غرضية التوجه لكي تحسن من نوعية أدائها في عمليات الصيانة والبرمجيات.

6. دراسة أعدها كل من الباحث A. Romanczuk والباحث I. Borne بعنوان " تحليل MERISE أكثر موضوعية باستخدام قواعد التحويل" 1997 .

لقد كان الهدف من هذا المشروع هو تصميم أداة نصف مؤتمتة من مخطط تحليل MERISE تُنتِج مخطط تحليل تقنية النمذجة الكائنية OMT: Object Modelling Technique. اعتمد هذا العمل على حالة مأخوذة من الواقع الحقيقي قدمتها إحدى الشركات التي رغبت في الانتقال إلى تقنيات غرضية التوجه (Object-Oriented) لكي تحسن من نوعية عمليات الصيانة ومن الأجزاء المرنة، حيث استخرجت قواعد الترجمة والتحويل بطريقة ترتيبية مخططة من تحليل MERISE للحالة المدروسة. وقد أدت المقارنة بين كلا الطريقتين إلى تحديد مجموعة من القواعد الأوتوماتيكية وكذلك مجموعة من المفاهيم التي لا تتوافق مع التحليل لكلا البيانات.

7. دراسة قدمها Antony Bryant and Lesley Semmens (Eds) في عام 1996 بعنوان "تحسين وصول مطوري البرامج التجارية إلى الطرائق الرئيسية: تكامل منهج MERISE مع Z" .

هدفت هذه الدراسة إلى تبيان أن الطرائق التقليدية تساهم بشكل فعال في تحديد مواصفات النظام، أما الطرائق المنهجية فهي أكثر ملاءمة للأنظمة التجارية وعلى الأخص بالنسبة للمستخدم، كما تبحث هذه الدراسة إمكانية تطبيق Z في MERISE، وتوصي باعتماد طرق ربط معينة بينهما حسب طريقة التكامل التي اقترحها .Semmes et.al. والتي تبين كيف أن التشارك مع الطرائق المنهجية يمكن له أن يحسن من الوصول إلى الطرائق الرئيسية باستخدام أمثلة من الحياة الواقعية.

8. دراسة بعنوان "نمذجة نظام المعلومات للبرامج البحثية باستخدام طريقة MERISE" مثال: مسح للأمراض البيئية التي تصيب الأبقار الحلوب، أبحاث الطب البيطري عام 1994 أعد هذه الدراسة كل من الباحثين: Perochonl و Lescourret. F

في هذه الدراسة تم تقديم طريقة MERISE من أجل تصميم نظام معلومات والنمذجة المفضلة للبيانات، وذلك باستخدام مثال عن برنامج البحث الاستكشافي حول أمراض ما قبل المخاض. لقد تم تقسيم نظام المعلومات إلى نظامين رئيسين يدعى كل منهما (حقل) والذي تم تمثيله بمخطط نظري شامل، من خلال الإرشادات النمذجية الدقيقة. يسهل هذا القسم إعطاء الصلاحية للنماذج من قبل الشركاء في المشروع ويجعل بالإمكان إعادة الاستفادة منه في البرامج البحثية الأخرى.

13. الدراسة الذي قدمها كل من Jose parets, Juan Carlos Toores

بعنوان الصيانة للبرمجيات مقابل تطوير البرمجيات منهج للتطوير نظم البرمجيات

(Software maintenance versus software evaluation: an approach to software system evolution)

يهدف هذا البحث للتغير النظرة التقليدية لتطوير البرمجيات كعملية حيث يفصل فريق تطوير البرمجيات والادوات والتطبيق وصيانة النموذج التنفيذي للمعلومات وذلك من خلال تقديم وجهة نظر تطوير البرمجيات نظرة ابداعية مع التفاعلات المعقدة بين فريق تطوير البرمجيات والنظام المستخدم هذا النهج يسمح لتصور الصيانة كنوع من الاسقاط الطبيعي للعملية التطوريه وقام

بعرض ثلاثه مناهج مختلفة في مفهوم نظم المعلومات وعرض التعقيد للعلاقات بين نظم المعلومات ونظام البرمجيات وتطوير النظام ثم عرض الفوائد للنهج التطوري علي المستوي المعرفي والمنهجي والعملي .

14. الدراسة التي قدمت من قبل German Godfrey,

بعنوان الماضي والحاضر والمستقبل لتطوير البرمجيات

The past, present, and future of the software evolution

التى قدمت في عام 2008 في "SWAG" software Archit Group

التغير هو السمة الرئيسية لتطوير البرمجيات والبرمجيات والنظم يجب أن تستجيب لمتطلبات التطور والمنصات والضغوط البيئية الأخري الهدف من الدراسة مناقشة مفهوم تطور البرنامج من عدة زوايا دراسة مدي ارتباطه وأختلافه عن الصيانة ومناقشة الافكار حول تطوير البرمجيات الناشئة من تطور البرمجيات ونموذج دورة تطوير الحياة من Bennett و Rajlich تم مقارنه انواع اخري من التطور من العلوم والعلوم الانسانية والاجتماعية وندرس القوة التي تغير الشكل . واخيراً مناقشة الطبيعية المتغيرة للبرنامج بشكل عام من حيث صلته بالتطور واقتراح التحديات والتوجيهات المستقبلية لبحوث تطور البرمجيات .

15. الدراسة التي قدمت بواسطة Bandakova بعنوان نحو تطور نظام البرمجيات باستخدام مبادي التكيف

Toward Evaluation of software system using Adaptive principles

قدمت في

Applied machine intelligence and informatics, 2008, Sami 2008 6th international symposium on 21-22 Jan 2008 IEEE

الخطوة الاولي في تطوير نظام البرمجيات هو نمذجة خصائص النظام المستهدف ' تطور النظام يبدأ فعلياً في العقل البشري في شكل افكار مجددة . المبرمج بحاول التعبير عن هذه الافكار في شكل تنفيذي للبرمجيات باستخدام واحدة من ادوات النمذجة القائمة ' ولكن مع الزمن الافكار المنهجية التي تم استخدامها اثناء تطوير البرمجيات تفقد . ويكون لدينا المفهوم الوحيد هو حل المشكلة. اذكانت هنالك بعض التغيرات لنظام البرمجيات والنظام ليس قادراً علي توليد التغيرات

للانسان . هنالك سوال كيف يمكن للنظام خلق مفهوم للتغير يفهمه الانسان؟ الفكرة الاساسية أن النظام يمكن أن يخلق مثل هذا المفهوم من خلال نموذج يمكن الانسان من التكيف وعمل التغيرات مع النظام يقوم بثوابت اللغة.

16. الورقة مقدمة من Jain Weili, Xunan Wang "

بعنوان Research and Practice of agile unified process

المقدمة في المؤتمر الدولي لتكنلوجيا البرمجيات والهندسة 2010 "ICSTE

من اجل الجمع بين اساليب التطويرالرشيق "agile" مع العملية الموحدة"UP" في مجال تطوير البرمجيات وتحلل هذه الورقة بعض الاساليب الرشيقة الممتازة الممتدة من المشاريع العملية . نظم الادارة وادارة التعليم عن بعد تم تصميم " agile up" النموذج الرشيق للعمليات الموحدة التي تمتص كل اساليب التطوير الرشيق والعمليات الموحدة ويحاول تقصير دورة تطوير وتحسين جودة البرمجيات والحد من مخاطر مشاريع البرمجيات

17. الدراسة التي قدمها كل من Putra , Yuliawati ,Mur Santo

في عام 2012 في "ICACSIS"

بعنوان Unified Process on agile development them

البرمجة المتطرفه "XP" هي الطريقة المستخدمة علي نطاق واسع لتطوير البرمجيات وتستخدم هذه الطريقة لتحسين جودة البرامج ومع ذلك فأن استخدام XP يقتصر علي المنظمات الصغيرة والمتوسطة . ولذلك تم تطوير البرمجة المتطرفه الصناعيه "IXP" لتلبية احتياجات المنظمات الكبية وهي بمثابة تطور . لـــ XP ومع ذلك IXP ليست مكتملة به كل الاجراءات والادوات التي يحتاجها المطور .

جاءات العمليات الموحدة لراشوانال "RUP" لتطوير البرمجيات التي تتسم بالمرونة لاي اطار اخر يتم تطبيقه فيه RUP يوفر الخطوات والمسئوليات الواضحة لعملية التطوير للبرمجيات قام الباحثون باقتراح اطار للحصول علي طريقة مناسبة للمنظمات الكبيرة وتوفير الراحة للمطوريين من خلال الجمع بين IXD و RUP.

18. الدراسة التي قدمها Khomh و Dhaliwal و Ying وAdams في عام 2011

بعنوان هل انشاء الاصدرات اسرع يحسن النوعية ؟ دراسة حالة موزيلا وفيرفوكس

Do faster releases improve software quality an empirical case study of Mozilla Fire Fox?

في الوقت الحاضر الكثير من شركات البرمجيات تتغير من دورة الاصدار الطويلة 18 شهر الي دورة الاصدار القصيرة علي سبيل المثال جوجل Google chrome و Mozilla Firefox تقوم بعمل اصدار جديد كل 6 أسابيع فهذا الاصدار يقلل من وقت انتظار المستخدمين وتوفر فرص التسويق أفضل للشركات ولكن غير الواضح اذ ماكانت نوعية المنتج تتحسن ام لا دورات للاصدارات أقصر تؤدي الي قصر وقت الاختبار الهدف من هذه الورقة هو دراسة حالة لعملية تطوير موزيلا فايرفوكس في العام 2010 و2011 هي الفترة التي انتقلت فيه الاصدرات الي فترة أقصر لتقيم العلاقة بين الفترة للدورة وجود البرمجيات وقد لوحظ من قبل المستخدم النهائي .

1 . مع الدورة القصيرة للاصدار المستخدمين لايعناون من الاخطاء السابقة .

2. الاخطاء تحل سريع

3. يكون للمستخدمين تجربة للتعامل مع هذه الاخطاء في السابق.

1-8 التعليق على الدراسات السابقة:

يتبين من الدراسات السابقة أنها أجريت في بيئات مختلفة مع اختلاف الأنشطة التي طبقت عليها أيضاً، وتتنوع العوامل والمتغيرات التي تناولتها، وقد أظهرت تلك الدراسات ضرورة الاهتمام بالعوامل المؤثرة في كفاءة وفاعلية نظم المعلومات عند بناء أو تطوير أداء النظم للوصول إلى نتائج أفضل ، ولعل أهم ما يميز هذه الدراسة أنها تركز على اعادة هندسة البرمجيات الموروثه الغير غرضية االتوجه باستخدم منهجية التحليل الغرضى التوجه بطريقة مبسطة غير معقدة ودراسة حالة لتطبيق هذاه المنهجيه برمجيات الادارة العامة للشئون الماليه بوزارة الدفاع باعتبارها تشكل جزءاً مهماً من القطاع المالى والمحاسبى بالسودان فضلاً عن أن طبيعة أداء هذا القطاع يتميز بوجود عمليات محاسبية ومالية أكثر تعقيداً من تلك الموجودة في قطاعات أخرى، وبالتالي فإن هذه الدراسة تتميز بمجتمعها ذات الطبيعة التخصصية من ناحية، وعدم وجود دراسة تبحث في تلك العوامل في القطاع المذكور على حد علم الباحث.

1-9 منهجية البحث:

يتبع الباحث منهجاً تحليلاً استقرائياً يبني على دراسة ميدانية تطبيقية للنظم المالية المؤتمتة مستخدم طريقة اعادة هندسة البرمجيات باستخدام الكائنى المنحى، يقوم الباحث منذ بداية عام2003 وحتى الان بالعمل لدى الجهة المذكورة وبدا منذ العام2010 لجمع المعلومات المطلوبة للتمكن تقيم النظم المستخدمة منذ بداية تطوير انشطة الادارة العامة للشئون المالية بوازارة الدفاع حتى 2015.

1-10 هيكل البحث:

تم تقسيم البحث فصول كما يلى :

الفصل الاول :المقدمة

الفصل الثانى : تطوير آنظمة المعلومات

الفصل الثالث : نماذج تطوير البرمجيات ونماذج جودة البرمجيات

الفصل الرابع : المنهجية

الفصل الخامس : دراسة الحالة

الفصل السادس: نموذج الربط المقترح

الفصل السابع : الاستنتاجات والتوصيات

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الباب الثانى

تطوير البرمحيات واعادة هندسة البرمجيات

2-1 الانظمة القديمة الموروثة (Legacy system)

مصطلح النظام القديم الموروث قد أستخدم من قبل مجموعة من المبرمجين اذ لم تكن منذ عدة سنوات قضية ولكن يدل ذلك علي أن الكثير من الناس في حقل البرمجيات مقتنعون بأن يصبح البرنامج الجديد برنامج موروث بسرعة وأن هذا يسبب العديد من المشاكل في الاعمال والايقأع السريع في التطور التكنلوجي وكل الدوافع وراء البحث في هذا الموضوع بسبب زيادة الطلب للقيام بتطوير البرمجيات لتكون اكثر كفاءة لانه تصبح صيانة البرمجيات هى المرحلة الاكثر كلفه في دورة حياة البرمجيات [12] والغرض الاساسي من طرق الموضوع اعلاه هو أثبات جدوي استخدام تقنيات التحول واستخدام التجريد لتطوير النظم القديمة الي اكثر موثوقية ومرونة لاطالة حياة المنتج البرمجي .

2-1-1 دورة حياة نموذج الشلال الاصلي "water fall" هي :

1. تحليل المتطلبات Requirements analysis
2. التصميم Design
3. التنفيذ Implementation
4. الصيانة maintenance

تحليل متطلبات النظام يحدد ماينبغي القيام به وتحت قيود معينة ' التصميم يحدد كيف يجب فعل ذك 'والتطبيق والتنفيذ يعني بناء نظام البرمجيات والصيانة تقوم بالحفاظ علي اداء النظام و وظائفه علي نحو فعال وكفؤ.

لقد مرت سنوات عديدة ولكن نموذج الشلال لازال يعمل ووثق في عام 1970 بواسطة رويس [13] وهو الاكثر قبولاً علي نطاق واسع ومع ذلك لابد من التوسع في عملية الصيانة ' لتمثل أنشطة اوسع بكثير ليس للحفاظ علي وظائف مصممة اصلاً ولكن ايضاً اضافة وظائف جديدة والتعامل مع البيئة المتغيرة ومواكبة التغيرات الجذرية ومتطلبات التوسع في الصيانة.

كلمة صيانة بالمعني العام تعني تحديد الاجزاء التي يتم استبعادها او استبدال الاجزاء التي تبلي . ولكن البرمجيات لاتخضع لهذا بالمعني العام لان الانظمة ديناميكية وهي بالتالي لا يمكن ان تبلي والصيانة هنا تعني ببساطة تحديد الاخطاء في النص الأصلي للتنفيذ واصلاحها هذا يتجاهل مشكلات بيئات سريعة التغير والمتطلبات .وهنالك العديد من انظمة البرمجيات التي اجبرت على الخضوع لتحسينات جذرية .

وتشير كل هذه الاعتبارات أن كلمة صيانة ينبغي أن يحل محلها أعادة الهندسة (reengineering) وفي الواقع لقد تم تعديل معظم البرامج القديمة بشكل كبير وهذا يقال عليه مشكلة الانظمة القديمة وأن الكثير من البرامج تواجهه هذه المشكلة.

2-1-2 **تغير الاعمال** Business change

أن نجاح العديد من الاعمال بشكل حاسم يعتمد علي البرامج التي تتطور من حيث توفير الخدمات وادارة الاعمال هذا ليس فقط للشركات التي تقوم بتطوير البرمجيات ولكن الشركات الاخري ايضاً مثلاً: عند دمج مجموعة من الهياكل فى شركة ووجد ان عملية دمج البرمجيات غير ممكن علي نحو فعال.

الاعمال بحاجة الي أن تكون علي النحو متزايد اكثر مرونة واستجابة للسوق وتطوير وتسويق خدمات جديدة في الوقت المناسب للقيام بذلك . دعم انظمة البرمجيات مطلوبة أن تكون مرنة وله امكانية التعديل الدورى والتحسين لعمل البرنامج بشكل صحيح يمكن نجاح وظائف الشركة وبذلك تطوير البرمجيات والاهداف المهمة في تطوير البرمجيات هي الموثوقية والمرونة هي الي حد كبير اكثر أهمية.

2-1-3 **تطوير البرمجيات** software Evolution

قبل عشرين عاماً كانت البرمجيات تصحح كل عام في بعض الاحيان ويتم اصدار جديد ربما مرة واحدة في العام واستخدام مصطلح صيانة لتمكين البرامج لمواصلة عمل ما كان عليه قبل عشر سنوات ولكن في العشرة سنوات الاخيرة البرمجيات اصبحت تحتاج الي أصدار رئيسى مع تعديل الوظائف وتعديل المتطلبات واستخدام مصطلح اعادة الهندسة يعني اضافة وتعديل الوظائف والمتطلبات الجديدة من المستخدمين . اليوم يحتاج البرنامج الي تغير علي اساس مستمر مع التحسينات الرئيسية المطلوبة في فترة زمنية قصيرة (أيام أو اسابيع وليس شهور أو سنوات)

من اجل تلبية متطلبات العمل الجديدة وفى هذه الحالة مصطلح ارتقاء مناسب ويكون هنالك حاجة للصيانة واعادة الهندسة في كثير من الاحيان .

اعادة هندسة البرمجيات لاتزال هي التقنية الاساسية لتطوير نظم البرمجيات وعملية اعادة الهندسة اصبحت ضرورة والخطوات الرئيسية لاعادة الهندسة هي:-

1. تحديد البرامج الموجودة " اي فهم البرامج الموجودة وماذا تفعل".

2. تحديد التعديلات المطلوبة في البرامج وكيفية تنفيذ التعديلات فعلياً .

فهم البرامج وسيلة لتحديد واستخراج تصميم البرمجيات الفعلي والحالي . فأن التصميم الحالي سيكون عادة مختلف عن التصميم الأصلي . فانه لا معنى له وغير ضروري لمحاولة لاستخراج التصميم الأصلي. ووثائق التصميم الأصلي فى بعض الاحيان قد تكون متاحة ولكن نادراً مايتم الاحتفاظ به و به كل التغيرات التي قد حدثت حتي وقت التطوير وبالتالي قد لا تصف التصميم الحالي للنظام.

ومع ذلك قد تكون هنالك فرصة لاستخراج التصميم الحالي مع الاصدار الحالي مع الرمز لأن التصميم الحالي يعكس طريقة عمل النظام . هذا يشير الي أن فهم البرمجيات القائمة الرئيسية سوف تكون له حاجة جنب الي جنب مع الطرق للتعامل مع التعقيد لتوليد وجهات نظر بديلة لاستيراد المعلومات المفقودة للكشف عن الاثار الجانبية وتوليف التجريد العالي.

اتخاذ قرار بشأن تعديل البرنامج ينطوي علي معرفة ماهي الاضافة والحذف الي البرنامج وفقاً لاحتياجات العمل .

تنفيذ تعديلات علي النظام ينطوي علي توظيف أفضل التقنيات المناسبة المتاحة لضمان أن البرنامج لديه تعديل ذو جودة وكفاءة عالية يحاول هذا الباب الاجابة علي الاسئلة الاتية حول تطوير البرمجيات بمرونة وموثوقية من خلال مناقشة القضايا التالية:-

1. اذ بدانا مع البرامج القديمة والبرامج سوف يتم تعديلها بشكل كبير بما في ذلك تعليمات البرامج باستخدام وسيلة رسمية او غير رسمية رغم أن معظم المنظمات قد استخدمت نوع من الاساليب الرسمية في تطويرها علي سبيل المثال (حالات الاستخدام وبطاقات التعريف) كيف يمكن تطبيق ذلك لاسخراج تمثيل على مستوى عالى(التصميم او المواصفات) من البرنامج.
2. اذا كان التطوير ممكن في ظل قيود معينة ماهى هذه القيود؟
3. ماهو الاطار والطريقة لاستخراج تصميم من البرامج الموجودة ؟
4. ماهو الاطار لتغير او لتعديل البرامج الذي يلي احتياجات العمل الحالي وتحقيق اعلي جودة؟
5. هل هنالك ادوات تدعم هذا النهج؟
6. ماهي المقاييس التى تقيس نتيجة الكود الذي سينتج من عملية اعادة الهندسية؟

2-2 هندسة البرمجيات والتطوير oftware Engineering and Evolution

2-2-1 تطور نظام الكمبيوتر computer System Evolution

عبر التاريخ نظام الكمبيوتر لديه ثلاثة عناصر رئيسية او فرعية هي نظام الاجهزة ونظام الاتصالات وكانت هذه العناصر الثلاثة وضعت بطريقة تقترن بشكل وثيق من بداية عصر الحوسبة الحديثة علي الرغم من أن الثورة التكنلوجيا للحوسبة هي مجرد بضع عقود قديمة وعدد كبير من طرق التطوير من حيث النظم والبرمجيات يمكن تقسيم تطور نظام الكمبيوتر الي مايلي من فترات :

1. **من أوخر** 1945 **الي منتصف** 1960 **الاجهزة** "hardware"

توفر قوة الحوسبة والاتصالات والبرمجيات ليس ضمن هذه القوة ونظام البرمجيات كان مصمماً خصيصاً ومعظمها موجهة نحو تطبيق محدد وغالباً ماتنفذ من جانب واحد واي شخص ينظر الي البرمجة كحرفة وتوزيع البرمجيات محدد نسبياً[3] .

2. **من منتصف** 1960 **الي اواخر** 1970

وتطور الأجهزة احرزت تقدماً كبيراً كان ولا يزال الاتصالات كجزء من الحوسبة ولكن نظم البرمجيات تطورت تطورات هامة بما في ذلك التطوير المتعدد للبرمجيات والنظم ونظم الوقت الحقيقي والجيل الاول من انظمة قواعد البيانات والاهم من ذلك بدأ الناس انتاج البرمجيات وبدء عملية التطوير والتوزيع علي نطاق واسع في سوق متعدد التخصصات وتم تمديد هذا البرنامج باضافة البيانات وبرامج جديدة لتلبية الاحتياجات الجديدة ’ منتجات البرمجيات لابد من تصحيحها ويتم الكشف عن الاخطاء وتعديدها كلما تغيرت متطلبات المستخدم أو تكيفها علي الاجهزة الجديدة التي تكون ضمن أنشطة صيانة البرمجيات ثم تم ادراك أن هذه البرامج تواجه أزمة ثم بدأ زيادة الجهد المبذول علي برامج الصيانة لاستيعاب ذلك ولكن جعلت الطبيعه تخصيص برامج مستمر يحافظ عليه من الصعب ذلك.

3. **من منتصف** 1970 **الي أواخر** 1980

جعلت الاجهزة مزيداً من التقدم الكبير من حيث التكامل والمعالجات اصبحت الأتصالات جزء هاماً من الحوسبة وقدمت هذه بنية لتطوير وتوزيع النظم وقد واجه مطوري البرمجيات مع مطالب تشغليه علي برامج الشبكات المحلية والعالمية والنطاق الترددي العالي للاتصالات

الرقمية . نشأة الاجهزة الشخصية "pc" وشركات البرمجيات مع مبيعات تصل الي ملايين النسخ.

4. من بداية 1990 وحتى الوقت الحالى

الأجهزة ونظم الاتصالات وأصلت في التطور وتزايد تطور البرمجيات وقد ساعد ذلك ظهور تقنيات البرمجة الكائنية والنظم الخبيرة ونظام الذكاء الاصطناعي وفي الوقت نفسه ظهور الشبكة العالمية لتقديم المعلومات وفي الوقت نفسه كان هنالك انفجار في انتاج البرمجيات.

اظهرت الزيادة السريعة في إنتاج البرمجيات كثير من المشاكل، وفي الوقت الحاضر لا تزال الأزمة البرمجيات مستمر بكثافة اعلى. المشاكل لا تقتصر على البرامج التي لا تعمل بشكل صحيح وفقا للمعايير المطلوبة. بدلا من ذلك، أزمة البرنامج تشمل المشاكل المرتبطة بكيف يمكننا تطوير البرمجيات، وكيف وقد كانت المشاكل المرتبطة بأزمة البرمجيات من قبل طبيعة البرنامج نفسه.

2-2-2 خصائص نظم البرمجيات الكبيرة [15]

1/ **التعقيد** "complexity"

هذه خاصية اساسية في جميع البرمجيات الكبيرة وهذا يؤدي الي العديد من المشاكل مثل صعوبة الاتصالات وغالباً ما تحدث بين فريق كبير المطوريين والتي يمكن أن تؤدي الي عيوب المنتج وتجاوز التكاليف والتاخير في الجدول الزمني لذلك قد يكون من الصعب تصور كل حالات النظام مما يجعل من المستحيل فهم كل النظام تماماً فمن الصعب الحصول علي لمحة عامة عن النظام لذلك المحافظة علي التكامل المفاهيمي تصبح علي نحو متزايد صعبة فمن الصعب ضمان أن يتم احتساب كل الصعوبات ومنحني التعلم حاد جداً للموظفين الجدد.

2/ **المطابقة** "Conformity "

تغير العديد من انظمة الحواسيب بسبب الحاجة الي التوافق مع المؤسسات البشرية المعقدة.

3/ **التغير** "change"

استخدام اي نظام ناجح سيخضع للتغير لتعزيز قدراته أو حتي تطبيقه خارج المجال الأصلي وكذلك لتمكينه من البقاء علي قيد الحياة خارج الحياة العادية أو ليعمل علي بيئات وانظمة تشغيل مختلفة.

4/ **الاخفاء** "invisibility"

بالنسبة لأنظمة البرمجيات المعقدة لا يوجد تمثيل هندسي في متناول المصممين وبناء الاجهزة المعقدة أو الاجهزة الالكترونية المعقدة . هنالك العديد من الرسوم البيانية المتميزة ولكنها تتفاعل مع روابط النظام باعتبارات معينة " التحكم في التدفق ' تدفق البيانات والاعتماد والتسلسل الزمني " طريقة واحده لتبسط هذه المحاولة للسيطره علي التعقيد هو قطع الروابط حتي تصبح الرسوم البيانية هياكل هرمية [16] . وع ذلك وجود نموذج دقيق أو تجديد للنظام قد اصبح لا يمكن الاعتماد عليها لتحسن وتعديل النظام علي فترة من الزمن.

2-2-3 **استعراض حالات الممارسة في نموذج المتطلبات تشمل مايلي** [17] :

1. اختراع المتطلبات وليس أستخلاصها ' أيضاً المتطلبات تاخذ في الاختبار الاولويات والتفضيلات بحيث ذات الاولوية الاقل يمكن التخلي عنها.
2. معظم المواصفات تزايديه " اي الزبائن هم نادرا مايقدرون علي توفير مواصفات كاملة في اي مرحلة من مراحل المشروع" .
3. هنالك فجوة بين المطور والمستخدم.
4. تغير واجهة المستخدم باستمرار.

الاعتراف بوجود مشاكل وتحديد أسبابها هي الخطوة الاولي لايجاد الحلول نفسها يجب أن تقدم المساعدة لعملية تطوير وتحسين جودة البرمجيات والسماح للبرمجيات بمواكبة عالم الاعمال.

ليس هنالك أفضل طريقة لايجاد حل لازمة البرمجيات ومع ذلك من خلال الجمع بين اساليب شاملة واكثر قوة في البنيات الاساسية لتطبيقات البرمجيات وتقنيات أفضل لضمان الجودة والفلسفة العالية للتنسيق والسيطرة والادارة والانضباط لتطوير البرمجيات " هندسة البرمجيات".

2-3 **هندسة البرمجيات** software Engineering

أستخدام هندسة البرمجيات كمصطلح يمكن ارجاعه علي الاقل في 1968 في مؤتمر الناتو "NATO Conference" عقد في جارمش بالمانيا الغربية والمتابعة في المؤتمر الذي عقد قرب روما ' ايطاليا في عام 1969 التعريف التالي من [18] هي أنشاء واستخدام هندسة الصوت والمبادي من اجل التطوير البرمجيات اقتصادياً التي يمكن الاعتماد عليها ويعمل بكفاءة علي أجهزة حقيقة .

وقد دفع هذا جزئياً من المشاكل التي واجهتها نظم تشغيل الحاسوب IBM360,OS360 هندسة البرمجيات لديها ثلاثة عناصر

1/ طرق "method" : والتي توفر تقنيات لبناء البرمجيات بما في ذلك هياكل البيانات ' بنية البرامج ' والخورازميات والترميز والاختبار والصيانة.

2/ ادوات " tools" : الادوات التي توفر الدعم الالي للطرق .

3/ العمليات "process" : التي تمكن الاساليب والادوات من العمل معاً وتمكن من التطوير " اي تحدد التسلسل الذي سيطبق أساليب وأنجازها والضوابط التي تساعد علي ضمان الجودة وتنسيق التغير والمعالم التي تمكن مديري البرامج لتقيم التقدم فى العمل.

هنالك طرق مختلفة للجمع بين العناصر الثلاثة المذكورة اعلاه ' العائد من عناصر هندسة البرمجيات المختلفة نماذج هندسة البرمجيات ' هنالك الكثير من النماذج لهندسة البرمجيات ' ويتم اختبار النموذج الصحيح بناء علي طبيعة المشروع والتطبيق واساليب والادوات والضوابط والانجازات المطلوبة سوف يتم مناقشة ثلاثة نماذج [14]

1. دورة الحياة التقليدية classic life cycle
2. النماذج الاوليه prototyping
3. تقنيات الجيل الرابع fourth – generation techniques

ويطلق علي نموذج دورة الحياة التقليدية ايضاً نموذج الشلال ليس هنالك تكرار في العملية من البداية الي النهاية ويتخذ منهج تسلسلي بدقة لتنفيذ الانشطة التالية :

تحليل المتطلبات – التصميم – الترميز – الاختبار والصيانة

2. نموذج النمذجة الاولية : تمكن المطور من أنشاء النماذج الاولية للبرامج [19] يتم بناءها لتسمح للمشاكل والمتطلبات التي يجب أن ينظر اليه بسرعة ' النموذج يبدأ بجمع المتطلبات حيث يجلس كل من المطوريين والعملاء لتحديد كل المكونات للنظام اي معرفة كل المتطلبات والخطوط العريضة بمزيد من التعريف للنظام ' ثم يحدث تصميم سريع يمثل كل الجوانب التي تكون مرئية للمستخدم عملية التصميم السريع تؤدي الي نموذج أولي يتم تقيم النموذج الأولي المطلوب هو تلبية حاجة العملاء وفي الوقت نفسه تمكين المطور لفهم ما يجب القيام به.

3. الجيل الرابع من تقنية "4GT" نموذج يضم مجموعة واسعة من الادوات البرمجية التي لديها شي واحد مشترك وهو تمكن مطور البرنامج لتحديد بعض الخصائص للبرنامج في أعلي مستوي [19] بالاداة "Tolls" يتم توليد الشفرة تلقائياً علي أساس مواصفات المطور.

و"4GT"لهندسة البرمجيات يركز علي القدرة علي تحديد البرامج الي الجهاز في المستوي الذي هو أقرب للغة الطبيعة أو تدوين يمنح وظيف هامة ولكنه يميل لاستخدامها مرة واحدة واضحة المعالم في مجال التطبيق ' أيضاً نهج "4GT" اعادة استخدام بعض العناصر مثل الحزم الموجودة وقواعد البيانات.

دورة الحياة الكلاسكية هي أقدم نموذج والأكثر استخداماً علي نطاق واسع لهندسة البرمجيات والقالب الذي يوفره التحليل – التصميم - الترميز – والاختبار – والصيانة لها نقاط ضعف .

علي سبيل المثال نادراً ماتتبع مشاريع حيقية تدفق متسلسل لان التكرار يحدث دائماً ويخلق مشاكل هذا النموذج لانه غالباً من الصعب من البداية العميل يعطي جميع المتطلبات صراحة وهنالك فارق زمني كبير بين مرحلة التصميم النهائي والمنتج النهائي خلال هذا الوقت من المرجح أن كثير من المهام زادت أو هنالك تغير في المتطلبات .

النموذج الأولي هو فعال لهندسة البرمجيات والمفتاح لتحديد قواعد اللعبة منذ البداية ' هذا يعني العميل والمطور علي حد سواء يجب أن يوافق علي النموذج الاولي حتي يفي كل المتطلبات .

المشكلة مع هذا النموذج أن العميل يرى انه نسخة من عمل البرنامج النهائي' غير مدركين استخدامه للحصول علي جودة البرامج وان طول أجل الصيانة لم ينظر اليه " اي في كثير من الاحيان يصبح النموذج النهائي موضع التنفيذ" علي الرغم علي ما قبل المرجح أن تصبح جزء متزايد الأهمية في برنامج "4GTs" للتطوير بسبب التخفيض الكبير في الوقت تطوير البرمجيات باستخدام ادوات 4GT .

استخدام ادوات 4GTليست أسهل بكثير من استخدام لغات البرمجة ' لأن مصدر التعليمات البرمجيه التي تستنتجها هذه غير فعالة والصيانة لنظم البرمجيات الكبيرة التي يتم تطويرها بأستخدام 4GT مفتوحة السوال وتشمل المشاكل القائمة التالية:-

1. لغات محددة بشكل واضح " غير مكتملة أو تتعارض.
2. عدم كفاءة التنفيذ.
3. اختلاط المستويات " الخروج الي لغات المستوي الادني"

4. نفص الموظفين المدربيين.

5. عدم وجود دعم من المطوريين.

ونتيجة لهذه المشاكل بعض الشركات الكبيره لديها رغبة علي محمل الجد علي ازالة من 4GTوعوضاً عن ذلك الحفاظ علي ملايين من الكود المولد بلغة كوبول .

لو اخذنا نظرة عامة لهندسة البرمجيات يمكن الحصول من خلال عملية تطوير البرمجيات [14] تتضمن العملية بصفة عامة ثلاثة مراحل بغض النظر عن نموذج هندسة البرمجيات الذي تم اختياره .

تعريف التطوير والصيانة واجهت جميع مراحل التطوير وتركز المرحلة الاولي من التطوير علي تعريف ماهي المعلومات التي سيتم تجهيزها ' ماهي الوظائف المطلوبة لاداء النظام ' ماهي الواجهات المطلوبة ' ماهي قيود التصميم الموجودة وماهي معايير التحقق المطلوبة لتعريف نظام ناجح توجد ثلاثة عمليات فرعية في هذه المرحلة

1. تحليل النظام بتعريف القواعد لكل عنصر قائم وتحديد الدور الذي يؤديه .

2. تخطيط مشروع البرمجيات وتخصيص الموارد وتقدير التكاليف وتحديد المهام والجدول الزمنية ووضع خطط الجودة وتحديد المخاطر .

3. تحليل الاحتياجات وتحديد مزيد من التفاصيل ووظيفة البرنامج قبل تبدأ الحل.

وترتكز هذه المرحلة علي كيفية تطوير " أي مطور للبرامج يحاول وصف وتعريف معمارية البرنامج والبيانات المتعلقة به ' وماهي الاجراءات التفصيلية التي تطبق وكيف يتم تحويل التصميم الي لغات برمجة وكيف تتم عملية الاختبار وهنالك ثلاثة خطوات محددة في هذه المرحلة

1. تصميم البرمجيات وترجمة متطلبات البرنامج الي مجموعة من التمثيلات التي تصف بنية البيانات والهندسة المعمارية والخورازمية الداخلية.

2. الترميز واداء الترجمة من تمثيل التصميم الي اللغة التي تنتج التعليمات للتنفيذ من قبل الكمبيوتر.

3. أختبار البرمجيات والكشف عن العيوب في الوظائف والمنطق والتنفيذ.

وتركز مرحلة الصيانة علي التغير المرتبط بتصحيح الخطأ والتكيف المطلوب مع تطوير بيئة البرمجيات والتعديلات بسبب التحسينات الناجمة عن تغير العملاء للمتطلبات ' ومرحلة الصيانة تطبق الخطوات للتعريف والتطوير ولكن ذلك من خلال البرامج الموجوده.

2-4 **جودة البرمجيات** software Quality

هندسة البرمجيات تعمل لتحقيق أنتاج عالي الجودة للبرمجيات لذلك من المفيد توضيح الجودة وعوامل جودة البرمجيات وتعريف الجودة والتوافق وتحديد المتطلبات بشكل صريح المتطلبات الوظيفية ومتطلبات الاداء وتوثيق عملية التطوير ويكون التطوير حسب المعايير والخصائص[14]

2.4.1 عوامل جودة البرمجيات :

1. الصحة " Correctness "

اي مدي تحقيق البرنامج للمتطلبات واهداف ومهام العميل

2. الموثوقية Reliability

الي أي مدي يمكن أن يتوقع اداء البرنامج ووظائفه المطلوبه مع الدقة المطلوبة .

3. المرونة Flexibility

الجهد المطلوب لتعديل البرنامج التشغيلي

4. الكفاءة Efficiency

كمية الموارد والشفرة المطلوبة من قبل البرنامج لاداء الوظائف المطلوبة .

5. السلامة Integrity

مدي امكانية السيطرة علي البرامج بعد بدخول الاشخاص غير المرخص لهم بالدخول الي البرامج والبيانات .

6. سهولة الأستخدام Usability

الجهل المطلوب لمعرفة تشغيل البرامج واعداد المدخلات وتغير الناتج من البرامج.

7. الصيانة Maintainability

الجهد المطلوبه لتحديد مكان وأصلاح الخطا أو تنفيذ بعض التغيرات الاخري في البرنامج .

8. قابلية الاختبار Testability

الجهد المطلوب لاختبار البرنامج لضمأن اداء الوظيفة المطلوبة .

9. امكانية النقل Portability

الجهد المطلوب لنقل البرنامج من جهاز / أو بيئة إلى أخرى

10. اعادة الاستخدام Reusability

ان البرامج أو جزء من البرنامج يمكن اعادة استخدامه في تطبيقات اخري .

11. التشغيل البيني Interoperability

الجهد المبذول للنظام من جهاز للاخر.

ضمان جودة البرمجيات هو النشاط التي ينبغي تطبيقها في كل خطوة في عملية هندسة البرمجيات ضمان جودة البرمجيات تشمل اجراءات التطبيق الفعال للاساليب والادوات والاستعراضات الفنية الرسمية استراتيجيات وتقنيات الاختبار واجراءات تغير السيطرة واجراء ضمان الامتثال للمعايير واليات قياس واعداد التقارير .

موثوقية البرمجيات والمرونة هما اهم عوامل الجودة ' وخاصة عندما يحتاج البرنامج الي تعديل كما أن الموثوقية يمكن أن تفسر عي أنها قدرة البرنامج للحفاظ علي مستوي اداء النظام عند استخدامه تحت شروط محددة.

انتاج برمجيات يمكن الاعتماد عليها هدف بالغ الأهمية من حيث البرامج والهندسة قد يسبب البرامج التي لا يمكن الاعتماد عليه خسائر مالية كبيرة ' ترتبط ارتباطاً وثيقاً الموثقية والمرونة في الحالة التي يحتاج البرنامج فيها الي تغير بإستمرار ارتفاع المرونة سيمكن التغير القادم علي أن تنفذ بسهولة مع الحفاظ علي الموثوقية.

2-5 البرمجيات في اي مكان Software everywhere

في الوقت الحاضر البرمجيات يمكن الأطلاع عليه في اي مكان ومع ذلك الكثير من البرمجيات ما زال التطوير مستمر فيها احد الاعتبارات الهامة فى نظام البرمجيات هو بيئة التطوير بأكملها في المعني العام تتضمن بئية التطوير والاساليب الفنية ' والاجراءات الادارية ' والمعدات الحاسوبية وطريقة أستخدام الكمبيوتر والادوات الالية لدعم عملية التطوير والموظفين وتطوير البرمجيات ومعالجة العمل اليدوي ومثالية بيئة التطوير ينبغي أن يعزز انتاجية النظام للمطورين وتوفير مجموعة من الادوات التي تبسط عملية انتاجية البرمجيات ' البيئة ابد أن تتضمن المرافق سواء بالنسبة للفرد أو مجموعة المطورين وعن الادارة العامة للمشروع [22] اليوم أصبحت هندسة البرمجيات ' واضحة المعالم بأستمرار مع تطور والانضباط في انتاج

البرمجيات يختلف كثيراً الان مما كانت عليه في عام 1968 عندما تم ادخال مفهوم هندسة البرمجيات اول مرة.

حالة التقنية الاضاحية في انتاج البرمجيات في ذلك الوقت يمكن النظر اليه من خلال الدراسات والمشاكل التي نوقشت في المؤتمرين للناتو عن هندسة البرمجيات في عام 1968 – 1969 علي سبيل المثال بعض القضايا التالية ابرمت [24,25] عن مجموعة التقارير وعن الادارة العامة للمشروع [22] .

1. مشاكل الحجم
2. الترتيب والاداء
3. التقنيات والاستراتيجيات اللازمة للاستخدام .
4. كيفية تحديد أنظمة البرمجيات.
5. تخطيط المشروع والتحكم به.
6. انشاء البرمجيات غير الموثوق به.

علي الرغم من أن بعض هذه المشاكل لا تزال حتي اليوم فقد كان التقدم بها وخاصة في المجالات التالية :-

- النمذجة " modeling" نمذجة النظام ونمذجة المتطلبات .
- اعطاء الطابع الرسمي "formalization" للمواصفات والتحقق .
- علوم الحاسوب "computer science" اللغات ومفاهيم البرمجة مثل التقسيم وانواع البيانات المحددة .
- أسلوب وتصميم النماذج
- البرمجة الكائنية.
- الدعم "support"
- قواعد البيانات وادوات وبيئات تطوير البرمجيات .
- العوامل البشرية Human Factors
- مشاركة المستخدمين وادارة المشروع وواجهات .
- القياسات : Metrics

- الجودة والاعتمادية والتكاليف علي الرغم من هذا التقدم ولكن لا تزال هنالك العديد من المشاكل التي لم تحل في المجالات التالية:

1. الطرق الرسمية :formal methods

مزيد من المواصفات وتحقيق التطوير والارتقاء لمواجهة مشاكل الحياة الحقيقية الكبيرة .

2. القياسات Metrics

طرق محسنة للتقيم والتنبوء بالتكلفة وجودة البرمجيات مثل الموثوقية والصيانة وغيرها .

3. اعادة الاستخدام Reuse

أن تكون وسيلة رئيسية لاحداث حاجة ماسة للزيادات في الانتاجية واذا كانت ممارسة سيكون لدينا القدرة علي طلب البرمجيات .

4. اعادة هندسة البرمجيات Reengineering

تحسين وسائل جديدة للحد من التكاليف وزيادة الاعتمادية والمرونة

5. الادارة : management

تقنيات اكثر موثوقية ' واكثر فعالية لادارة جميع جوانب دورة الحياة.

6. ادوات الدعم Tools support

زيادة توفير الدعم الالي للبرمجيات لدعم جميع أنشطة هندسة البرمجيات سواء علي اساس الفرد وبوصفها بيئة دعم متكاملة .

7. التكنلوجيا التطبيقية Applied technologies

تطبيق تقنيات اخري مثلالذكاء الاصطناعى لتعزيز العمل لمهندسي البرمجيات.

2-6 صيانة البرمجيات Software maintenance

في الايام الاولي من الحوسبة "1950-1960 " في البرمجيات تتالق الصيانة بجزء صغير جداً من دورة حياة البرمجيات في اواخر 1960 و1970 قد تم انتاج المزيد والمزيد من البرمجيات والناس بدات ادراك أن البرامج القديمة لا تنمو بسهولة من هذه النقطة بدات صيانة البرمجيات يتم الاعتراف بها كنشاط رئيسي بحلول 1970 كان المجال يعاني مشاكل كبيرة من تراكم التطبيقات وصيانة البرمجيات تتخذ المزيد من الجهد في التطوير في بعض القطاعات.

في عام 1980 اصبح من الواضح تقيد المعمارية القديمة في التعميم الجديد [25] . كل هذه العوامل اداة علي زيادة الطلب علي تغير كبير في البرمجيات وتشغل هذه التغيرات تحديد

الاخطاء والتحسينات الي جانب حلول المشاكل التي تتطلب تغيرات في المقام الاول وتنفيذ التغيرات يولد مشاكل جديدة .

واحدة من خمسة قوانين لليمان لتطوير نظام البرمجيات مباشرة يتناول تحليل البرمجيات وهو ينص علي أن البرنامج الذي يتم أستخدامه في بيئة العالم الحقيقي يجب تغيره أو يصبح أقل واقل فائدة في تلك البيئة [26] وفقاً لذلك يجب تطوير اليات للتقيم والسيطرة واجراء التغيرات ويعرف صيانة البرمجيات بأنه تعديل منتج البرمجيات بعد تسليمها لتحسين الاخطاء وتحسين الاداء وغيرها من الصفات أو للتكيف المنتج في البيئة المتغيره [27] وصيانة البرمجيات مطلوبه لتلبية انواع التغيرات الرئيسية وبالتالي أنشطة الصيانة يمكن تقسيمها الي هذه الفئات المقابلة [28] :

1. **الصيانة التصحيحية corrective maintenance**

قد يكون هنالك خطا في البرنامج بحيث لا يتفق سلوكها مع مواصفاتها من الخطاء قد يؤدي الي التعارض مع المواصفات أو انها قد تثبت أن المواصفات غير مكتملة أو ربما غير متناسقة حتي يتسني للمستخدم المحافظة علي المواصفات المفتوحة شمل الصيانة التصحيحة ازالة هذه العيوب . حتي ولو كان نظام البرنامج الى حد ما خالي من الاخطاء . اتغيرات البيئية التي سوف تعمل عليها غالباً ما تكون عرضة للتغير علي سبيل المثال الترقية من جهاز الي اخر وتحريك النظام من المركزية الي جهاز مركزى فتوصف بالصيانة التكيفية .

2. **صيانة التكيف adaptive maintenance**

علي سبيل المثال الشركة المصنعة قد تعرض اصدارات جديدة من نظام التشغيل وازالة الدعم من البرامج الموجودة وقد أسند البرنامج لبيئة جديدة والي أجهزة مختلفة.

3. **الصيانة التحسينية perfective maintenance**

هذا يتم نتيجة لتغير لمتطلبات المستخدم فعلي سبيل المثال قد يحتاج جناح الرواتب لتعديل البرنامج لادخال الضرائب ' نظام مراقبة الكهرباء في الوقت الحقيقي قد يحتاج الي تطوير لتلبية معايير السلامة الجديدة.

4. **الصيانة الوقائية preventive maintenance**

توقع مشاكل في المستقبل وجعل الصيانة اللاحقة سهلة [29,30] مثال ذلك جزء واحد من مجموعة كبيرة يتطلب التصحيح المستمر علي مدي فترة من الزمن ' فمن المعقول اعادة هذا الجزء باستخدام هندسة البرمجيات الحديثة.

التكلفة الكبيرة المرتبطة بصيانة البرمجيات هي نتيجة لحقيقة أن البرامج أثبت من الصعب الحفاظ عليها تميل النظم مبكراً ليكون الغرض منه غير منتظم وغير واضح . وثائق النظام غالباً ما تكون غير مكتملة أو خارج مدة صلاحيتها مع الاساليب الحالية غالباً مايكون من الصعب اعادة اختبار أو التحقق من النظام بعد اجراء التغير .

سوف تتطور برامج ناجحة لا محاله ولكن عملية التطوير يؤدي الي تدهور البنية وزيادة التعقيد[32,33,34] الان وضح أن برامج الصيانة هي الاكثر تكلفة في مرحلة من مراحل دورة حياة البرمجيات لمعظم المشاريع في عام 1970 من 30% الي 40% من ثم استخدام ميزانيات عالية علي صيانة البرمجيات من 40% الي 60% في 1980. اليوم ميزانية الصيانة تصل من 70% الي 60% برامج الصيانة لها دورة في دورة حياة البرمجيات علي مدي السنوات السابقة اقترحت عدة نماذج مهمة للبرمجيات النموذج المذكور هنا يمكن أن الصيانة أن تحدث بسبب تغير يحتاج للمستخدم الي تصحيح الاخطاء أو تغير في البيئة علي الرغم من هذه الانواع مختلفة علي المستوي التفصيلي علي مستوي اعلي يمكن وصفها بعملية من ثلاثة مراحل متكررة.

1. طلب التحكم Request Control:

يتم جمع المعلومات عن الطلب ويتم تحليل التغير باستخدام تحليل الأثر لتقسيم التكاليف والاستحقاقات ويتم تحديد الاولية لكل طلب.

2. تغير التحكم Change Control

يتم اخذ الطلب من القائمة ذو الاعلي اولية وتحلل المشكلة (اذ وجدت) يتم تحليل الترميز (التصميم والتعريف اذ وجد) يتم تصميم التعديل وتوثيقه واختبار المنتج المعدل وتاكيد الجودة ثم تنفيذه .

3. التحكم في الاصدار :Release Control

يتم تحديد الاصدار الجديد ' يتم اختبار التقنية ثم يتم توزيع الاصدار لاختيار القبول من قبل العميل . حالياً تجري هذه الخطوات الثلاثة دائماً من شفرة المصدر ومعلومات التصميم وحتي الوثائق غير كافية في كثير من الاحيان لا وجود لها . لذلك تعتمد صيانة البرمجيات من شفرة المصدر لفهم وظائف وسلوك النظام من خلال الشفرة مهى جزء حيوياً من صيانة البرامج ومهمه [33]

قد ركزت معظم الجهود البحثيه علي صيانة البرمجيات والاساليب والتقنيات والادوات التي تدعم عملية الصيانة عندما يتم تنفيذ أنشطة الصيانة خارج السمة السالبة للجميع . اذا اخذنا صيانة البرمجيات بعين الاعتبار الصيانة هي المفتاح والهدف الذي يواجه بعين الاعتبار في هندسة البرمجيات ' الصيانة سهلة اذ كان البرامج مفهوماً ويمكن تصحيحها وتكيفها وتعزيز ذلك [14] تتأثر صيانة البرامج بكثير من العوامل من الصعب تحديد كل العوامل ومع ذلك بذلت كثير من الجهود لمعالجة هذه المشكلة من زوايا مختلفة والعوامل التي وضعها **Kopetz** [34] عرف عدد من العوامل المتعلقة بتطوير البيئة : توافر الموظفين المؤهلين للبرمجيات وفهم بنية النظام وسهولة معالجة النظام ' واستخدام البرامج القياسية ' واستخدام انظمة التشغيل القياسية تعطي هيكل ووثائق البرنامج قياسية وتوافر حالات الاختبار الذي بني في التصحيح المرفق وتوافر جهاز الحاسوب المناسب لاجراء الصيانة [35]وتوفر مقاييس الصيانة عن طريق قياس الجهد الذي انفق اثناء الصيانة النفقات التالية :

1. وقت الاعتراف بالمشكلة Problem recognition time.
2. الوقت الاداري Administrative time.
3. وقت جمع ادوات الصيانة Maintenance tools collection time.
4. وقت تحليل المشكلة Problem analysis time.
5. وقت تغير المواصفات Change specification time.
6. وقت نشاط التصحيح Active correction time.
7. وقت الاختبار المحلي Local testing time.
8. وقت الاختبار العام Global testing time.
9. وقت استعراض الصيانة Maintenance review time.
10. مجموعة زمن الاصلاح Total recovery time.

[35] يقيس الصيانة من حيث نفقات التطوير الاصلية بنفقات الصيانة ويقوم باخراج نسبة العوامل التي تؤثر علي هذه النفقات وهي:-

1. التجزئة : العملية تقيس مدي تقسيم النظام الي أنظمة فرعية مستغلة.
2. المرونة : العملية تقيس مدي استقلال نظام البرمجيات من اى تطبيق محدد.
3. التعقيد: العملية تقيس تجميع نظام البرمجيات وتوزيع المكونات الي مجموعات [35] .

ويمكن تقسيم سمات البرمجيات الي نوعين داخلية وخارجية الصفات الداخلية هي خاصية من البرنامج نفسه " علي سبيل المثال تعقيد وحجم هياكل البيانات والاقتران والتماسك والجودة والاعتمادية ' والصفات الخارجية هي خاصية البيئة علي سبيل المثال توافر ادوات التصحيح والمهارة والتدريب وادارة المستودع ' ربما العامل الاكثر أهمية الذي يؤثر علي الصيانة والتخطيط للصيانة ' اذ نظرنا الي البرنامج وعناصر البرامج التي ستخضع حتماً للتغير فأن فرص هذا البرنامج في الصيانة من المرجح انه سيزيد بشكل كبير [14] . ومع ذلك الصيانة تعتمد علي البرمجيات نفسه ومشكلة اخري في الصيانة هي تنفيذ التغيرات التي لم تضع في التصور حتي تصميم البرنامج ولم يتم التخطيط له فتتعامل معه الصيانة وهي السمة الاساسية للبرمجيات في كل مرحلة من مراحل هندسة البرمجيات والصيانة ايضاً يجب النظر اليها خلال مرحلة المتطلبات ومجالات تعزيز المستقبل والمراجعة المحتملة.

وفي هندسة البرمجيات نوقشت واجهات النظام التي ينظر اليها في مرحلة التصميم ' تصميم البيانات وتصميم الهندسة المعمارية وتصميم الاجراءات ' ينبغي تقيمها قبل الوصول الي شفرة البرامج.

أيضاً ينبغي اجراء انشطة الصيانة بطريقة متانيه لانه عملية تعديل البرنامج هو خطير بمعني أن الاخطاء والسلوكيات غير المرغوب فيها او غيرها من الاثار الجانبية [14] قد تحدث نتيجة لتعديل البرمجيات وتشمل الاثار الجانبية للوثائق عندما يتم تغير الشفرة بحيث يكون له اثار جانبية جزئية اخري تحدث هذه الاثار الجانبية عند تشفير تصميم البيانات ربما لم تعد صالحة او عند عمل تغيرات علي الشفرة ولا تنعكس علي الوثائق التصميم أو ادلة المستخدمين.

2-6-1 تطوير البرمجيات

يمكن القول بأن من اليوم الذي يذهب فيه نظام برمجي للعمل لاداء الخدمات و الوظائف واجراء العمليات في البيئة المحددة له سيكون عرضه للتغير وعلاوة علي ذلك فأن نظام البرمجيات عند التسليم سوف يحتوي بعض العيوب الكامنة التي لم تكتشف خلال الاختبار هنالك عدة عوامل تؤدي الي أن نظام البرمجيات حتماً ستتطور في المجال والبيئة و الوظائف من اجل البقاء لفترة طويلة [38,39] .

2-7 اعادة هندسة البرمجيات للتطويرSoftware Reengineering For Evaluation

تطور البرمجيات هو عملية مستمرة لاعادة هندسة البرامج 'اعادة الهندسة تعني تغير واحد في الدورة ولكن التطور يمكن أن يستمر الي الأبد . وبعبارة أخري الي حد كبير تطوير البرمجيات تكرار عملية اعادة الهندسة.

اي نظام حوسبة بما في ذلك الأجهزة والبرمجيات ستنمو حتماً من حيث الحجم و الوظائف ويسبب هذا التعقيد احتمال اخطار خفيفة وعلاوة علي ذلك فان بعض هذه الأخطاء قد يسبب خسارة كارثية من المال والوقت او حتي حياة الانسان ' هنالك انظمة كبيرة ومعقدة بحيث يسجل لفرد يقوم ببناء وصيانة جميع جوانب تصميم النظام والهدف الرئيسي لهندسة البرمجيات هو تمكين المطوريين لبناء الانظمة التي تعمل بشكل موثوق علي الرغم من هذا التعقيد [36-39] طريقة واحدة لتحقيق هذا الهدف واستخدام الطرق الرسمية واللغات استناداً علي الرياضيات والتقنيات وادوات للتحديد والتحقق من انظمة البرمجيات استخدام الطرق الرسمية لا يضمن دقة ولكن في وسعهم ذيادة كبيرة لفهم النظام من خلال الكشف عن التناقضات'والغموض وعدم الاكتمال [40] .

مزيج من الهندسة العكسية والهندسية الي الامام تقنيات اعادة هندسة البرمجيات هو الحل العملي لمشكلة تطور النظام المحوسب الموجود ادارة التغير الديناميكية لانظمة البرمجيات قد تم تنفيذها الي حد كبير عن طريق أستخدام تقنيات مخصصة التي عادة ماتكون مكلفة نوعاً ما وهنالك اثنين من المزايا :

1- استخدام الطرق الرسمية كأساس لبرامج اعادة الهندسية يمكن أن تساعد مهندس البرمجيات للحصول علي وصف دقيق للنظام الذي يجري اعادة هندسته بالتالى زيادة كبيرة في جودة النظام الجديد.

2- الاتمتة وهي واحدة من الاهداف الرئيسيةلاعادة الهندسة من خلال تطبيق الطرق الرسمية formal methods وهذه المزايا تعطي بشكل طبيعي المرونة والاعتماد لاعادة تصميم البرمجيات لانه مستقبل أعادة الهندسة لايزال بادوات آليه تدعمه والموثوقية تعزز بالضبط والدقة التي تقدمها الطرق الرسمية.

2-7-1 دورة أعادة هندسة البرمجيات Software reengineering cycle

لفهم أفضل لعملية أعادة هندسة البرمجيات ندرس اولاً الرسم البياني لدورة اعادة الهندسة [الشكل -] الذي عرفه [41] Bachman والذي يتميز بالهندسة الامامية والعكسية الهندسة العكسية تبدأ من الجزء السفلي الايسر من الشكل مع تعريف التطبيقات القائمة ويجعل التطبيقات علي مستوي عالي من التجريد في الجزء العلوي من الشكل تصميم الكائنات التي تكونت من خطوات الهندسة العكسية وتعزز وتحقق من التصميم العكسي الكائنات التي يمكن أستخدامها في عملية الهندسة الامامية في اسفل الشكل -ــ يصبح تطبيق النظام الجديد بتطبيق النظام الموجود يذهب الي تصميم هذا النموذج العديد من أنظمة البرمجيات يخضع عادة للمراحل التالية:

تعريف المواصفات التصميم التنفيذ مواصفات التصميم ! Specification

Specification ! Design ! Implementation (! Design)هذا يمثل العملية قبل تنفيذ البرنامج وقد حددت المواصفات اولاً ثم أشتقاق تصميم معين ثم تنفيذ المواصفات لبعض من الوقت ' بعدها البرنامج يحتاج الي تغير وتصميم أو مواصفات والتي تكون مختلفة عن الموجوده والتي تم الحصول عليها في الهندسة العكسية . وتعبر المواصفات علي مايفعله البرنامج والتصميم كيف يطبق هذا البرنامج حسب المواصفات الموضوعه وكيف يؤدي هذه المهمة ' وبالتالي فأن العملية المذكورة اعلاه يمكن تكون ممثله علي النحو التالي:

ماذا ؟ ماذا وكيف ؟ كيف ؟ بماذا وكيف ؟ ماذا والمواصفات والتصميم والتنفيذ عادة يترتب علي مستويات مختلفة من التجديد للانتقال من مرحلة الي اخري.

وفي مستويات التجريد عادة ما تكون المواصفات الاكثر تجريداً من تنتفيذها وبالتالي يمكن مرة آخري أن يكون تمثيل العملية المذكورة اعلا مايلي:

مجرد؟ اقل تجريداً؟ المملوسه [الفيزيائة] ؟ اكثر تجريداً ' مجرد هذا يشير الي أن مستوي التجريد من البرنامج هو مهم وميزه عند كل من الهندسة العكسية والامامية لذلك التجريد مهم لكل منهم.

2-7-2 تصنيف اعادة هندسة البرمجيات Taxonomy of software reengineering

في هذا القسم المصطلحات الاساسية التالية نوفر نطاق واضح في مجال اعادة هندسة البرمجيات

1- **الهندسة الامامية Forward engineering**

هي عملية تقليدية للانتقال من المستوي الاعلي تجريد وهنا التنفيذ مستقل عن التصميم الفعلي للنظام.

2- **الهندسة العكسية** Reverse engineering

هي عملية تحليل النظام للاتي:

1. التعرف علي مكونات النظام وترابطها.
2. أنشاء تمثيل للنظام بشكل اخر او علي مستوي من التجريد

3- **اعادة التوثيق** Redocumentation

هو أنشاء أو تعديل للنظام الموجود علي سبيل المثال تدفق البيانات التوثيق وهياكل البيانات والتحكم والتدفق اي تعديل التمثيل و واجهات بديله وهو أبسط شكل من اشكال الهندسة العكسية . ويمكن أن يكون الشكل الطبيعي اعادة الهيكلة.

4- **استعادة التصميم أو التصميم العكسي** Design recovery or reverse design

هي مجموعة فرعية من الهندسة العكسية في مجال الدقة والمعلومات الخارجية وتخصم أو تضم الي المنطق الضبابي لملاحظات النظام الموضوع لتحديد تجريدات ذات مغزي علي مستوي أعلي . ويتم الحصول عليها مباشرة عن طريق فحص النظام نفسه ' أستعادة التصميم يعبرعن تجريدات التصميم بمزيج من الموجود و وثائق التصميم أذ وجد والخبرة الشخصية والمعرفة العامة حول المشكلة وتطبيقات المجال.

5- **فهم البرامج أو استيعاب البرامج Program understanding or program comprehension**

هو المصطلح المتعلق بالهندسة العكسية فهم البرنامج يعني دائماً أن الفهم يبدا بشفرة المصدر حيث أن الهندسة العكسية يمكن ان تبدأ في شكل ثنائي وقابل للتنفيذ من النظام او الوصف رفيع المستوي في تصميم البرمجيات . ويشغل فهم العلوم المعرفية العقلية والبشرية .

عمليات فهم البرنامج يمكن أن يتم بطريقة مخصصة وليس علي التمثيل الخارجي فقط بينما الهندسة العكسية في نهج منظم لتطوير التمثيل الخارجي لفهم النظام لمقارنته مع التصميم المحسن وكل ذلك يبدأ علي مستوي شفرة المصدر.

6- **اعادة الهيكلة** Restructuring

هي التحول من احد الاشكال لتمثيل اخر علي نفس المستوي التجريدى النسبي مع المحافظة علي السلوك الخارجي الذي يخضع له النظام (اي الوظائف والدلالات).

7- **اعادة الهندسة** Reengineering

هي فحص وتعديل نظام يخضع لاعادة ذلك في شكل جديد وتنفيذه لاحقاً في النموذج الجديد' عملية هندسة النظم المحوسبة تتضمن ثلاثة خطوات رئيسية:

1. الهندسة العكسية
2. اعادة الهيكلة الوظيفية
3. الهندسة الامامية

8- **المواصفات العكسية** Reverse specification

هو نوع من الهندسة العكسية ' حيث تستخرج المواصفات من التعليمات البرمجية للمصدر أو مواصفات التصميم والمواصفات في السياق تعني وصفاً مجرداً لبرنامج ما في الهندسة الامامية للمواصفات نعني ما الذي يقوم به البرنامج ومع ذلك هذه المعلومات ليست المدرجة في شفرة المصدر . فقط في حالات نادرة يمكن استردادها من التعليمات البرمجية لشفرة المصدر أو من الاشخاص الذين شاركو في عملية الهندسة الامامية الاصلية.

9- **اعادة الترميز** Recode

ينطوي علي تغير خواص شفرة المصدر ' لغة الترجمة واعادة هيكلة رسائل التحكم في شفرة المصدر ' وتشمل التغيرات الاخري الممكنة المطابقة لمعايير الترميز وتحسين وتسجيل قراءة شفرة المصدر واعادة تسمية عناصر البرنامج.

10- **اعادة التصميم** Redesign

ينطوي علي تغير خصائص التصميم والتغيرات الممكنة وتشمل التغيرات اعادة هيكلة معمارية التصميم وتغير نموذج بيانات النظام كما ورد في هياكل البيانات او وتحسين الخوازمية.

11- **اعادة التعريف** Respecify

ينطوي علي تغير خصائص الشروط ' يمكن نوع التغير يشير الي تغير فقط في شكل متطلبات موجودة اي اخذ المتطلبات الرسمية التي اعرب عنها للغة وتغير هذه اللغة هذا النوع من التغير

يمكن ايضاً أن يشير الى تغيره متطلبات النظام مثل اضافة متطلبات جديدة أو تغير القيود الموجوده.

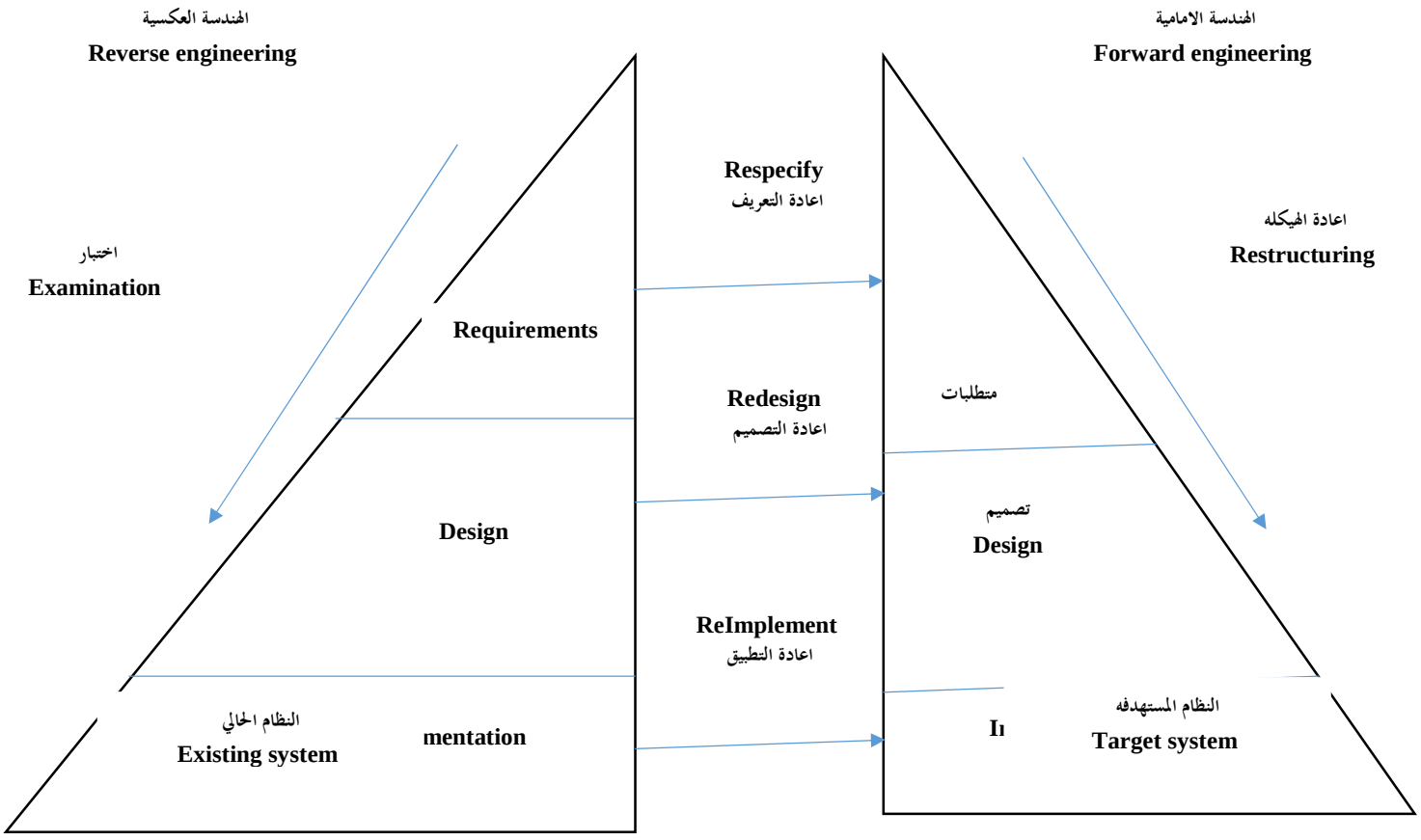
يعرض الشكل 2-1 نموذج عام للهندسة العكسية ويعرض الشكل 2-2 نموذج عام لاعادة الهيكل من حيث التعاريف المذكورة اعلاه .

هيكل الوظائف

Functional restructuring

(Alteration)

"البدائل "



الشكل 2-1 النموذج العام للهندسة العكسية

level المستوي	Reverse engineering الهندسة العكسية	Forward engineering الهندسة الامامية
Requirements المتطلبات		
Specification التعريف	ANALYST تحليل	BUSINESS الاعمال
Implementation التطبيق	ANALYST تحليل	DATA تحليل البيانات SYSTEM النظام
Operation العمليات		

الشكل 2-2 النموذج العام لاعادة الهندسة

2-7-3 الهندسة العكسية Reverse engineering

الهندسة العكسية تنطوي علي تحديد أو استعادة متطلبات برنامج أو مواصفات التصميم التي يمكن أن تساعد في فهم وتعديل البرنامج. والهدف الرئيسي هو اكتشاف السمات الأساسية للنظام بما في ذلك متطلبات البرمجيات والمواصفات والتصميم والتنفيذ وبعبارة أخري لاسترداد وتسجيل المعلومات رفيعة المستوي عن النظام بما في ذلك التالي :

1. بنية النظام ومكوناتها والعلاقات بينهما.
2. الوظائف والعمليات لكل مكون.
3. السلوك المتحرك للنظام أو كيف يتم تحويل المدخلات الي مخرجات.
4. المنطق لها (عملية التصميم التي تقرر بين عدد من البدائل في كل خطوة للتصميم)
5. بناء وحدات الاختبار والتوثيق .

هنالك عدة أغراض لأجراء الهندسة العكسية المدرجة في [8] انها يمكن فصلها عن قضايا الجودة " علي سبيل المثال لتبسيط البرمجيات وتحسين نوعية البرامج وازالة الاخطار" وقضايا الأدارة " علي سبيل المثال رفع مستوي البرمجة وتمكين وتحسين ادارة برامج الصيانة " والمسائل التقنية " علي سبيل المثال السماح بتغيرات كبيرة في البرنامج التي يتم تنفيذها لاكتشاف وتسجيل تصميم النظام وتمثل نموذج العمل الاساسي الذي تبني عليه البرمجيات".

تعتبر الهندسة العكسية هي النشاط الذي يفحص ويفهم كائنات النظام وتسجيل النتائج للاختبار والفهم من ناحية أخري الهندسة العكسية هي المفتاح لبقية عملية إعادة الهيكلة . لانها تمكننا من التمثيل المجرد للنظام القائم "من شفرة المصدر" التي تستخدم في اعادة الهيكلة اللالحقة أو حتي معاودة تطبيق النظام. لأن تقنيات وأساليب الهندسة العكسية غير ناضجة يجب يجب النظر في الاحتياطات الستة التالية عند اجراء الهندسة العكسية:

1. قد تكون الشفرة معينة وليس عامة حتي تعكس تصميم كل النظام .
2. قد يكون الشفره به أخطاء وأنه ليس من الواضح اذكان من المفيد عكس الهندسة لتحسين الاخطاء في التعليمات البرمجية .
3. الهندسة العكسية نفسها قد تتعرض للاخطاء واعادة التصديق عليها سيكون في خطة المشروع.

4. الهندسة العكسية سوف تكون مكلفة والمرجح غير واضح دائماً بالتالي سوف يكون هنالك حاجة الي تحليل التكاليف والفوائد.

5. لا توجد معايير او طرق معياريه للهندسة العكسية .

6.لم تعد هنالك اي تدابير راسخة لاعادة الهندسة . واحد اهداف الهندسة العكسية هو برنامج لاستخراج تصميم والمواصفات من شفرة البرنامج هنالك سببان لذلك:

الاول: هو انه من اجل تحقيق مكاسب انتاجيه كبيره ويجب أن يكون التغير علي مستوي عالي من التجريد والمرونة " اي مستوي التصميم او مستوي المواصفات" وهذا ينطبق علي مايلي لستة أسباب [8]

1- تمثيل نظام علي مستويات اعلي من التجريد هو اكثر احكاماً من علي مستويات منخفضة وبالتالي فأن النظام هو أسهل للفهم ككل.

2- الكائنات التي تمثل النظام في مستويات عالية من التجريد " علي سبيل المثال نماذج ومتطلبات المواصفات ' هي الهياكل التي تنتج انظمة الصيانة وعلاوة علي ذلك فهي اقرب الي مجال التطبيق ' ويتم التعبير عن العديد من المتغيرات من حيث مجال التطبيق.

3- وثائق صيانة النظام يمكن أن محددة بوضوح .

4- التعديل يمكن السيطرة عليه بشكل أفضل مما يؤدي الي تدني اقل في الهيكلة .

5- تقنيات هندسة البرمجيات الحديثة المتاحة لمهندس البرمجيات يؤدي الي جودة عالية في اعادة الهندسة في هذه المرحلة .

6- المستوي العالي من التجريد للكائنات هي المكونات المناسبة التي يمكن التعبير عنها في خطة الاختبار.

السبب الثاني هو الحاجة لمشاريع اعادة الهندسة لان المراجع والوثائق ذات الصلة ليست موجودة بشكل كامل ' وقد يكون هنالك بعض الوثائق المتاحة ولكن لا يتم تنفيذ البرنامج بأستمرار مع الوثائق و الوثائق الأصلية قد كتب به كل التحديث وبالتالي تكون غير صالحة للاستعمال في البرامج الحديثة لاعادة الهندسة.

هذا يعني أن استخراج تصميم البرامج أو المواصفات من شفرة البرنامج القديم هو خطوة حيوية خصوصاً عندما يكون البرنامج هو فقط الوثائق المتاحة أو هو المصدر الوحيد الذي يعتمد عليه.

الغرض من هذا النوع من الهندسة العكسية هو:

1. اعادة تطبيق النظام.

2. المساعدة في فهم البرنامج.

3. لايمكننا ببساطة اعادة التطبيق والبرامج مباشرة سبب الاستمارات الكبيرة في البرامج الموجودة

2-7-4 الوضع الحالي للطرق الرسمية في اعادة الهندسة

الحالة الراهنة للطرق الرسمة في اعادة الهيكلة في هنالك جدل حول أستخدام وأهمية الطرق الرسمية تقدم حل كامل لتطوير النظام ومدرسة اخري تزعم أن الطرق الرسمية أستخدامه قليل " بسبب التكلفة التي قد تنطوي عليها" هنالك وجهة نظر ثالثه هي أستخدام الطرق الرسمية وطرق الفهم.

اي كانت المدرسة الفكرية التي تنسب اليه أنه من المهم ان ندرك أن تعقيد النظم في البناء ينمو بأستمرار ووجود منهجية منضبطه ومنتظمة ودقيقة امر ضروري لتحقيق مستوي معقول من الاعتماد والثقة في هذه الانظمة. وأستجابة لذلك وجد النشاط البحثي المكثف مما ادي الي أنتاج تقنيات تطوير رسمية وادوات التحقق المرتبط بها التي تم تطبيقها بنجاح في الهندسة الامامية هذه النظم علي سبيل المثال عمليات الجبر والمنطق الزمني وتم استخدامها مع درجة من النجاح في مجال الهندسة العكسية كما تم وضع الطرق الرسمية الي الامام كوسيلة لتحقيق مايلي:

1. تحديد الfrom والتحقق من النظام القائم لاسيما تلك التي تعمل بالفعل في تطبيقات السلامة الحرجة.

2. تقديم وظائف جديدة

3. الاستفادة من التحسن في تقنيات تصميم النظم.

نحاول مراجعة فئة كبيرة من الطرق الرسمية التي هي مفتاح لاعادة هندسة النظم المحوسبة [14-18] سوف تقوم ايضاً بمناقشة بعض الفوائد والقيود ومع ذلك فمن الضروري اولاً وضع بعض الأسس والمصطلحات والنظره الحالية للممارسة.

يتم أستخدام الطرق الرسمية كمصطلح للاشاره الي الاساليب مع أسس سليمة في الرياضيات وينبغي التميز بين هذه الاساليب الهيكلية التي تعرف في [19] . الطرق الرسمية تجعل وظائف النظام تكون محددة بدقة في حين تسمح بشكل منهجي ومنتظم ودقيق لمواصفات هيكل النظام ومع ذلك في الأونة الاخيرة كانت هنالك أنشطة بحثية كبيرة للقيام بمايلي:

1. دمج الطرق الرسمية والبنية مع اسلوب منظم معروف [20,21] .

2. توسيع بعض الطرق الرسمية للمتطلبات الغير وظيفية مثل الوقت والموثوقية [22,27] .

نحن نري أن الطرق الرسمية يجب أن تكون من بعض المكونات للنموذج الدلالي وهي لغة المواصفات وادخال حساب بالتفاضل والتكامل والمبادي التوجيهية للتطوير والادوات الداعمة ونظام التحقق.

1/ النموذج الدلالي: هي بنية الرياضيات والمنطق السليم يضمن جميع الشروط والقواعد المستخدمة بمزيد من الدقة ينبغي ان يعكس النموذج الدلالي النموذج الحسابي للتطبيق المقصود.

2/ مواصفات اللغة: هي مجموعة الرموز التي تستخدم لوصف السلوك المقصود للنظام هذه اللغة يجب أن يكون لديها دلالات مناسبة داخل النموذج.

3/ نظام للتحقق وحقل قواعد حسابية سليمة والتي تسمح بالتحقق من خصائص وحقل المواصفات .

4/ تطوير المبادي التوجيهية هي الخطوات التي تبين أستخدام الاسلوب.

5/ وتشمل ادوات دعم مثل دليل مساعد لبناء الجمل ومدقق للكتابة والرسوم المتحركة والنماذج الاولية.

والطرق الرسمية يمكن تطبيقها بطرقتين مختلفتين:

1. لانتاج المواصفات التي به الاسس التقليدية لتطوير النظام .

2. لانتاج المواصفات الرسمية علي النحو الوارد اعلاه لاستخدامها.

2-8 **التلخيص** :

نظراً للتطور السريع في الحاسوب والاجهزة والبرمجيات ومطالب وتكاليف التغيرات في البرنامج في زيادة مستمره تشمل تغيرات البرمجيات الا جزء كبيراً من تكاليف دورة الحياة لان هنالك تغيرات علي البرنامج باستمرار وصيانة البرمجيات لم تعد معبرة بمافيه الكفاية لوصف مثل هذه التغيرات وتطور البرمجيات اصبح الان ضرورة.

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الباب الثالث

نماذج عمليات البرمجيات

3-1 مقدمة:

لا ينكر أحد أهمية الحاسوب في حياتنا اليومية، وفي هذه الفترة لاسيما أنه أقتحم العديد من مجالات الحياة إن لم تكن كلها فقلما نجد منتج إلا وقد أصبح الحاسوب أحد أجزاء السيطرة في صناعته وبالذات في الدول المتقدمة صناعيا وتكنولوجيا .ونظرًا لأن الاعتماد على أجهزة الحاسوب قد أصبح كبيرًا خاصة في الآونة الأخيرة مما أدى ذلك إلى زيادة عدد الساعات التي يجلس فيها الكثيرون أمام أجهزة الحاسوب لإنجاز الأعمال المختلفة في المكاتب والشركات أو للترفية والاستمتاع في المنازل بغية الفائدة .وما ذلك إلا نتيجة للتقدم التكنولوجي في صناعة الحاسوب وأيضًا في صناعة البرمجيات الميسرة فقد أصبح الحاسوب هام جدًا في جميع نواحي الحياة وبالخصوص في المجال المعلوماتي ومجال قواعد البيانات حيث أصبح من غير الممكن حصر الشركات المصنعة للبرمجيات التي تسهل الأعمال المكتبية والإدارية في الشركات الخاصة والعامة والبنوك والمنظمات الحكومية والعلمية وذلك من خلال تسهيل العمليات التي تجرى لأي بيانات متداولة بكثرة في تلك المنظمات.وبتقدم علم الحاسوب أصبح من السهل إجراء أي من العمليات المعقدة والطويلة والمتكررة بسرعة كبيرة جدًا ودقة عالية حيث أن البرمجيات المتعلقة بالمجال المعلوماتي أحكمت السيطرة على المعلومات في أي منظمة كانت من خلال حمايتها وضمان أمنها وسريتها .فأصبحت البرمجيات العنصر الجوهري في تطور النظم والمنتجات المعتمدة على الحاسوب. ولقد تطورت البرمجيات خلال العقود الأربعة الماضية، من كونها مشكلة خاصة أو أداة تحليل معلومات إلى صناعة بحد ذاتها .لكن ثقافة البرمجة وتاريخها في البداية أوجدت مجموعة من المشاكل التي بقيت حتى اليوم ، أصبحت البرمجيات عاملا لا معوقًا لتطوير النظم المعتمدة على الحاسوب .وتتكون البرمجيات من برامج ومعطيات ووثائق ، يحوي كل من هذه العناصر تشكيلة أنشئت لتكون جزء من إجرائية هندسة البرمجيات .كما أن الهدف من هندسة البرمجيات هو توفير إطار عمل لبناء برمجيات ذات جودة عالية [3] .

3-2 عمليات البرمجيات Software Processes

هى مجموعة من الأنشطة المترابطة المتماسكة المطلوبة لتطوير وإنتاج النظم البرمجية.والأنشطة العامة هى: توصيف المتطلبات، التصميم، التنفيذ، الإختبار، التحقق، الصيانة، إرتقاء النظم البرمجية. وتمثل هذه الأنشطة فى نموذج عمليات البرمجيات. هي

مجموعة من الأنشطة المرتبة المترابطة (Consistency) والقيود (Constraints) والموارد (Resources) التى تهدف لتوصيف (Specifying) وتصميم (Designing) وتنفيذ (Implementing) واختبار (Testing) وصيانة وترقية (Maintenance and Evolution) منتج برمجي يفي بجميع متطلبات العميل, وغالباً ما يقوم بهذه الأنشطة مهندسوا البرمجيات.

ويمكن وصف عمليات البرمجيات بوضع هيكل عام لها (Process Common Framework) ، وذلك من خلال وضع عدد من النشاطات فى أطار محدد (Framework Activities) والتي يمكن تطبيقها على جميع المشاريع البرمجية بغض النظر عن حجمها أو تعقيدها ، وعدد من مجموعات المهام (Task Set) اللازمة لكل نشاط هيكلي ، هذا بالإضافة إلى مجموعة من نشاطات هندسة البرمجيات والتي يطلق عليها " نشاطات المظلة "، وهي عبارة عن مجموعة من الأنشطة مستقلة عن أي نشاط برمجي هيكلي وتُجرى طوال عملية بناء المنتج البرمجي (مثل : متابعة المشروع البرمجي ومراقبته ، ضمان جودة المنتج البرمجي ، إدارة تكوين البرنامج ، إدارة المخاطر ، وخلافه) .

وتشير تلك العمليات (**Process** Software) ــ في الوقت نفسه ــ إلى مراحل تطوير البرمجيات. وبمفهوم علم هندسة البرمجيات (**Software Engineering**) يُطلق على تلك العمليات أيضاً - اصطلاح عام وهو دورة حياة البرمجيات (**Software Life Cycle**) . وعلم هندسة البرمجيات هو علم يهدف إلى إنتاج برمجيات خالية من الأخطاء وذات جودة عالية في وقت محددة وبميزانية محددة ، وبطريقة اقتصادية بحيث يفي بجميع متطلبات الجهة المستفيدة. وبغض النظر عن مجال التطبيق أو حجمه أو درجة تعقيده ، وبدون الدخول في تفصيلات النماذج (**Models**) التي تصف عمليات المراحل يمكن تقسيم مراحل تطوير البرمجيات إلى المراحل الرئيسية التالية :

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

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

4. مرحلة صيانة وترقية البرنامج (**maintenance Phase**) .

وتُعد نتائج كل مرحلة من هذه المراحل هى معطيات للمرحلة التي تليها ، فمثلاً نتائج مرحلة تحليل المشكلة هي معطيات لمرحلة التصميم ، ونتائج مرحلة التصميم هي معطيات لمرحلة التنفيذ ، وهكذا وسوف نتناول الآن هذه المراحل بإيجاز وبدون الدخول في التفاصيل الخاصة بكيفية إنجاز كل مرحلة.

3-3 نماذج عمليات البرمجيات

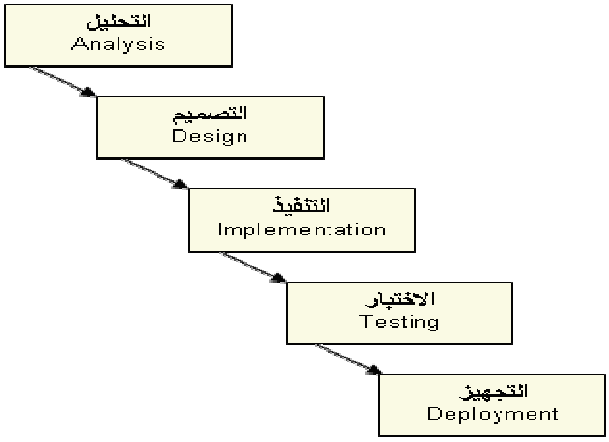
نموذج العملية البرمجية Software process model هو تمثيل مجرد لوصف العملية من منظور معين معين[1] ومن النماذج العامة لعمليات البرمجيات:

1. Waterfall Model
2. Iteration Model
3. V-shaped Model
4. Spiral Model
5. Extreme Programming (XP) Model

وقد تم اختيار هذه النماذج ودمجها لما فيها من مميزات تتماشى مع تطوير البرمجيات دورة حياة النظم وعلى النحو الآتي:

3-3-1 نموذج الشلال [4] [5]

تم اختيار هذا النموذج كونه يعد من النماذج الأقدم والأبسط حيث تسير دورة الحياة بشكل تدريجي وواضح لذا فإنه يسهل على المطور توضيح كيفية سير العمل بالمشروع للعميل والمراحل المتبقية للعمل، الشكل(3-1)، وقد كان هذا النموذج أساس عمل كثير من المؤسسات لفترة طويلة مثل وزارة الدفاع الأمريكية واستنبط منه العديد من النماذج الأكثر ويعتبر هذا النموذج الأكثر استخدامًا في هندسة البرمجيات، ولهذا النموذج مميزات وعيوب على النحو الآتي[6]



الشكل 3-1: نموذج الشلال

مميزات النموذج:

1. سهولة فهمه وتطبيقه.
2. مستخدم بشكل واسع ومعروف.
3. معالمه واضحة ومحددة : يعرف التحليل قبل التصميم والتصميم قبل التنفيذ.
4. يعمل هذا النموذج مع النظم الكبيرة.

عيوب النموذج :

1. مثالي، لا يطابق هذا النموذج الواقع بشكل جيد أحياناً.
2. لا يعكس هذا النموذج الطبيعة التكرارية للتطوير الاستكشافي.
3. غير واقعي لتوقع متطلبات دقيقة في بداية المشروع.
4. يتطلب فترة طويلة أثناء تطوير النظم.
5. لا يدعم هذا النموذج إدارة المخاطر.
6. ارتفاع كلفة الصيانة التي قد تنتج بسبب تغييرات في متطلبات النظام.
7. التكلفة المرتفعة في تطوير النظم الصغيرة.

3-3-2 :**النموذج التكرارى Iteration Model [7]** :

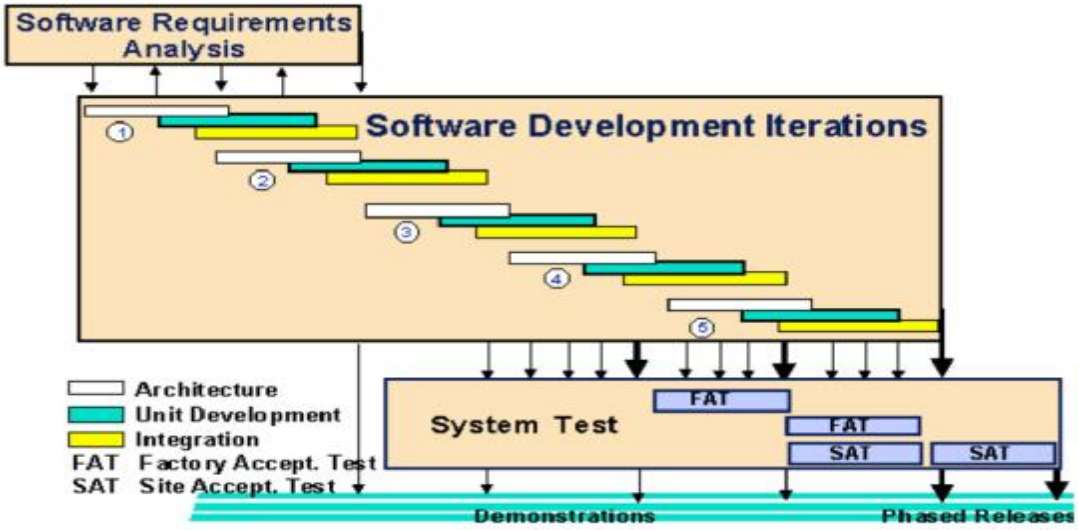
غالبًا ما تستخرج متطلبات النظام من خلال سير المشروع لهذا تتكرر العملية ويعاد العمل على المراحل المبكرة من المشروع خاصة في النظم الكبيرة، وقد تم اختيار هذا النموذج لإمكانية العودة لأي مرحلة سابقة في عملية التطوير وإعادتها عدة مرات، الشكل (3-2)

مميزات النموذج :

1. يعطي فرصة أكبر لتفاعل المستخدم مع المشروع.
2. يركز على مهارات التواصل والتنسيق أثناء تطوير المشروع.
3. قدرة النموذج على تحسين المتطلبات من خلال استيعاب المتطلبات المتغيرة أثناء التطوير.

عيوب النموذج:

1. يأخذ وقت أطول في تطوير المشروع.
2. ربما يقود إلى بعض التشويش بعد كل مرحلة بسبب عدم اكتمال المشروع بصورة نهائية، لذلك يجب تطوير آلية للتحكم في الطلبات الجوهرية للمشروع.

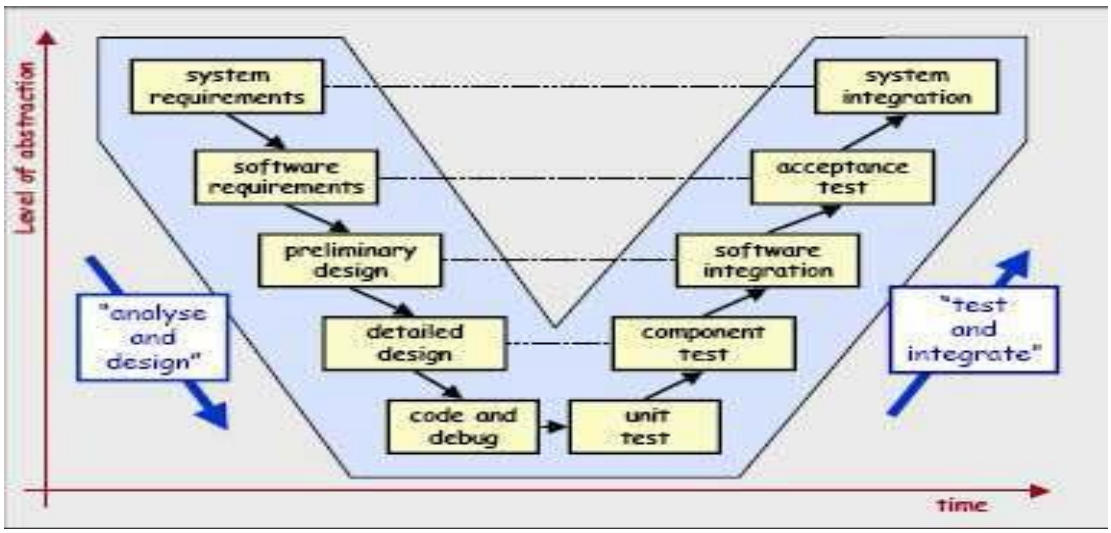


الشكل 3-2: النموذج التكرارى

3-3-3 V-shaped Model [8]

من حيث تسلسل العمليات وعدم البدء بالعملية التالية قبل الانتهاء من Waterfall يعتبر هذا النموذج شبيه بنموذج العملية الحالية، غير أن هذا النموذج متقدم من ناحية اختبار الأنشطة للمراحل المختلفة، وقد تم اختيار هذا النموذج لما فيه من مميزات في عملية الاختبار وتقسيم مرحلة التصميم إلى مرحلتين هما مرحلة التصميم العلي ا(Test) حيث (Low-Level Design Phase) LLDPومرحلة التصميم الدنيا-High Level (Design Phase)HLDPتركز

المرحلة الأولى على معمارية النظام وتصميمه بشكل عام والأخرى على تصميم مكونات النظام بشكل تفصيلي، أنظر الشكل(3-3) .



الشكل 3-3: يوضح نموذج V-shaped Model

ولهذا النموذج العديد من المميزات والعيوب كما يلي [9]

مميزات النموذج

1. سهل وبسيط الاستخدام.
2. كل مرحلة لها محدداتها الخاصة.
3. نتيجة لاختبار الخطة في .3 waterfall فرصة نجاح تطوير المشروع باستخدام هذا النموذج أكبر من نموذج

مرحلة مبكرة أثناء دورة حياة النظام.

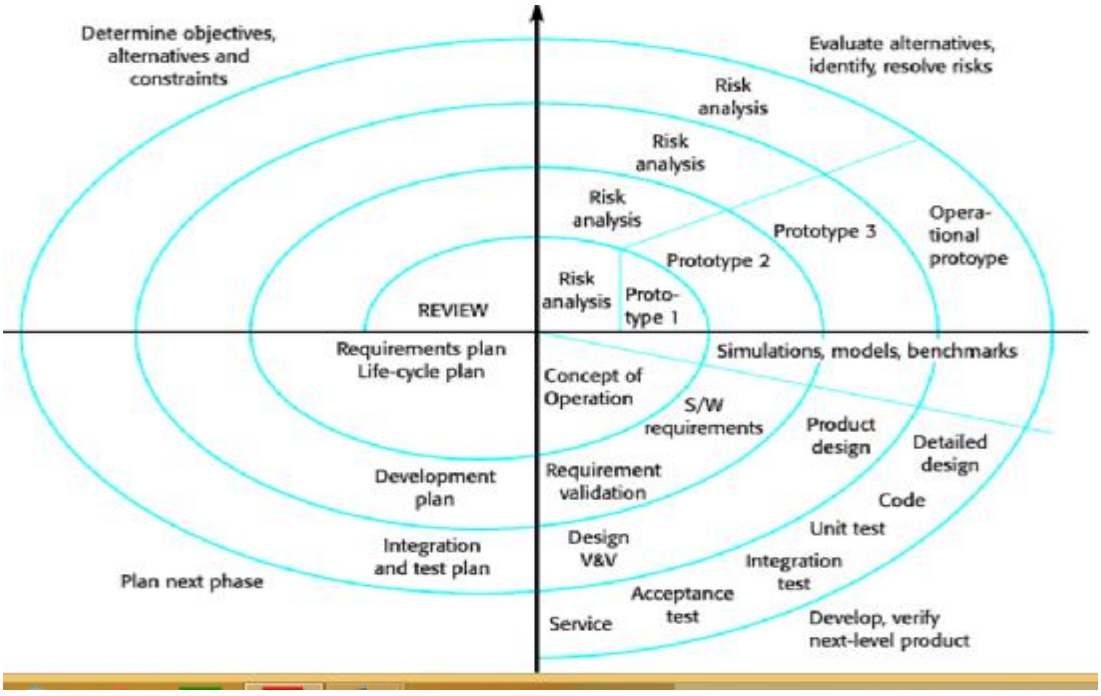
4. يعمل هذا النموذج جيدًا مع المشاريع الصغيرة التي تكون فيها المتطلبات واضحة.

عيوب النموذج

1. غير مرن بشكل كافي مثل نموذج waterfall.
2. التعديل ضمن هذا النموذج صعب ومكلف.
3. البرمجيات المطورة باستخدام هذا النموذج تنتج في وقت متأخر في مرحلة التطبيق.
4. لا يزود النموذج بطريقة واضحة لحل المشاكل أثناء مرحلة الاختبار.

3-3-4: النموذج الحلزونى Spiral Model [5]

النموذج الحلزوني الذي اقترحه "Boehm" هو نموذج تطويري لعمليات البرمجيات، ويتم فيه دمج فعاليات التطوير مع إدارة المخاطر من أجل التحكم بها وتقليلها .ويعتبر النموذج الحلزوني طريقة واقعية لتطوير نظم وبرمجيات واسعةالنطاق .وتم اختيار هذا النموذج لتركيزه على إدارة المخاطر وعلاقتها بالمراحل المختلفة لعمليات التطوير،أنظر الشكل (3-4) .



الشكل 3-4: نموذجSpiral Model

وللنموذج الحلزوني العديد من الميزات والعيوب كما يلي[10]

مميزات النموذج

1. يعطي هذا النموذج أهمية بالغة لتحليل المخاطر.
2. يعتبر نموذج جيد للمشاريع الكبيرة والحرجة.
3. البرمجيات تنتج في مرحلة مبكرة من دورة حياة النظام.
4. يستطيع فريق العمل أن يشتغل على كامل الدورة الحياتية (تحليل، تصميم، توليف، اختبار) بدلا من صرف سنوات على نشاط واحد.
5. يمكننا الحصول على ملاحظات وتقييم الزبون مبكرا و بصورة منتظمة، ورصد الصعوبات المحتملة قبل التمادي بعيدا في عمليات التطوير.

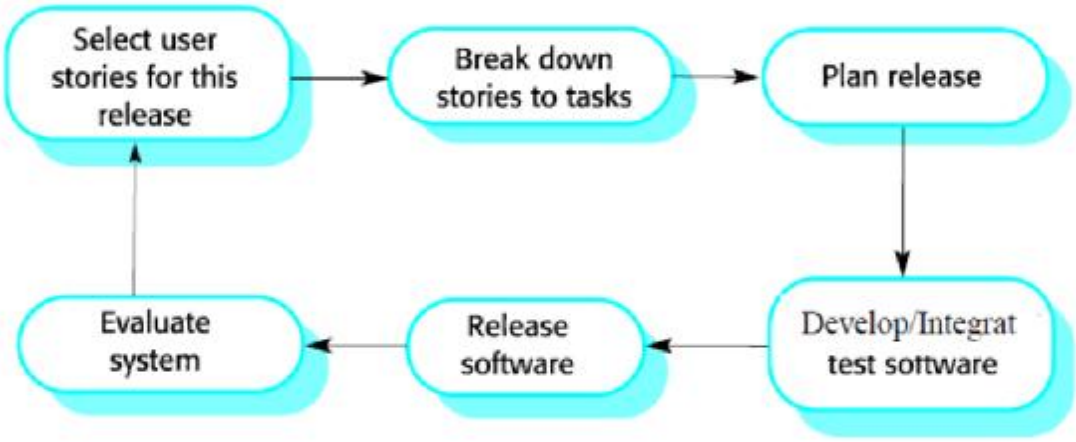
6. يمكننا التصدّي لنقاط المخاطرة مقدما، بالأخص التكرارات ذات المجازفة العالية (مثلا: التكرار الذي يتطلب تنفيذ بعض التقنيات الجديدة غير المجرّبة) يمكن تطويرها أولا.
7. يمكن اكتشاف مدى حجم و تعقيد العمل مبكرا.
8. الإصدار المنتظم للبرنامج يعزّز من الثقة.
9. الوضع الحالي للمشروع (مثل: مقدار ما تم انجازه) يمكن تحديده بدقة أكبر.

عيوب النموذج

1. يمكن أن يعتبر هذا النموذج مكلف من ناحية تطوير النظم.
2. عملية تحليل المخاطر ضمن هذا النموذج تحتاج إلى خبرة عالية.
3. نجاح المشروع يعتمد بالأساس على مرحلة تحليل المخاطر.
4. لا يستخدم مع المشاريع الصغيرة.
5. عادة ما تقترن هذه العملية بما يعرف بالتنشئة السريعة و التي تعتبر من قبل كثيرين مجرد عمل هواة (hacker's charter).
6. العملية أكثر صعوبة عند إدارتها. ففي النموذج الانحداري يمكن الاستعانة بالتقنيات التقليدية لإدارة المشروعات مثل مخططات غانط (Gantt Charts)، لكن العمليات اللولبية تتطلب أساليب مختلفة.

3-3-5: Extreme Programming (XP) Model [6]

يستخدم هذا النموذج في تطوير النظم الصغيرة التي تعتمد على كتابة البرامج بشكل مباشر دون الخوض في عمليات التحليل والتصميم المعقدة، وتم اختيار هذا النموذج لملائمته مع تطوير النظم الصغيرة وغير المعقدة، الشكل رقم(3-5)



الشكل 3-5: نموذج Extreme Programming (XP)

ولهذا النموذج العديد من المميزات والعيوب كما يلي:

مميزات النموذج

1. نموذج تطوير مناسب للمشاريع الصغيرة والمتوسطة.
2. يعتمد على النموذج التكراري.
3. يعمل على إنتاج نظام متكامل بشكل سريع.
4. يعتمد على جودة المتطلبات.

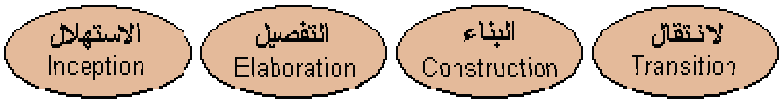
عيوب النموذج

1. صعوبة استخدام هذا النموذج مع المشاريع الكبيرة التي تتطلب توثيق.
2. يتطلب خبرة ومهارة عالية في التطبيق.
3. قد يكون مكلف أحياناً، بسبب اعتماده على خبرات برمجية عالية.
4. مرحلة الاختبار في هذا النموذج تعتبر صعبة وتعتمد على مهارة عالية.

3-4 العمل التكراري التزايدي (Iterative, Incremental Framework)

إطار العمل التكراري التزايدي (Iterative, Incremental Framework) هو امتداد منطقي للنموذج اللولبي، لكنه أكثر تقنينا و صرامة. ينقسم إطار العمل إلى أربعة أطوار رئيسية: (Inception)، التفصيل (Elaboration)، البناء (Construction) والانتقال (Transition). يتم انجاز هذه الأطوار على التوالي، لكن يجب أن لا نخلط بين هذه الأطوار و المراحل في

الدورة الحياتية للنموذج الانحداري. في هذا القسم سوف نشرح هذه الأطوار و نستعرض النشاطات التي يتم أداؤها في كل طور.



الشكل 3-6: الأطوار الأربعة لإطار العمل التكراري التزايدي

1. مرحلة البدءاو الاستهلال Inception Phase

يتعلق طور الاستهلال بوضع نطاق المشروع و تحديد التصوّر العام له. بالنسبة للمشاريع الصغيرة يمكن لهذا الطور أن يكون مجرد دردشة بسيطة على فنجان قهوة، يعقبها اتفاق على البدء في المشروع. في المشاريع الكبيرة يتطلب الأمر المزيد من التحرّي. المخرجات (deliverables) المحتملة من هذا الطور هي:

- وثيقة التصوّر.
- استكشاف مبدئي لاحتياجات الزبون.
- التحديد الابتدائي لمفردات (glossary) المشروع.
- دراسة جدوى (تتضمن محددات النجاح، التنبؤات المالية، تقديرات العائد على الاستثمار، الخ).
- التحديد المبدئي لنقاط المخاطرة.
- خطة المشروع.

2. التفصيل

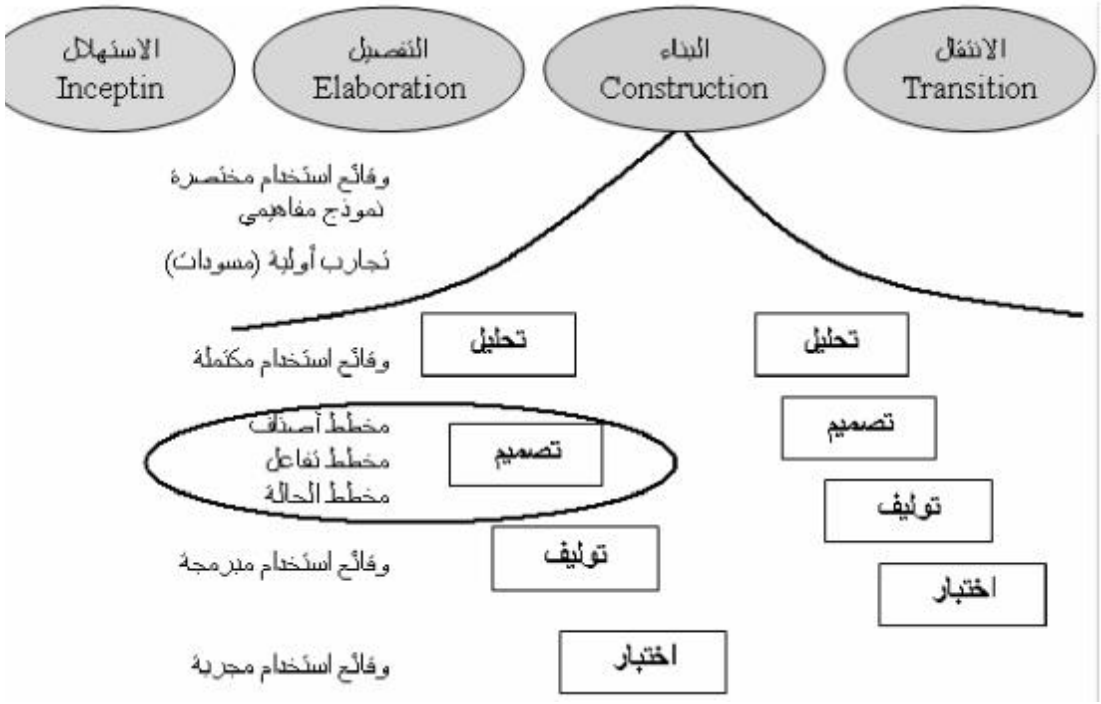
الغرض من التفصيل هو تحليل المشكلة، و المضي خطوة ابعد في إعداد خطة المشروع، و استبعاد المناطق الأكثر مخاطرة فيه. مع نهاية طور التفصيل؛ نتأمل حصولنا على فهم عام لكامل المشروع، و لكن ليس بالضرورة فهما متعمقا (لاحقا سيتم التصدي له بصورة أجزاء صغيرة يسهل مناولتها).

نموذجان من UML يكون لهما قيمة كبيرة في هذه المرحلة. نموذج حالة الاستخدام Use Case سيساعدنا في متطلبات المستفيد (أي الزبون)، و مخطط الصنفية Class

Diagram و الذي يمكن أيضا استخدامه لاستكشاف المفاهيم العامة التي يتصورها المستفيد. المزيد حول هذا الموضوع قريبا.

3. البناء

في طور البناء، نقوم ببناء المنتج. هذا الطور لن يتحقق بأسلوب خطي Linear ؛ بل يتم بناؤه بنفس أسلوب النموذج اللولبي، من خلال سلسلة من التكرارات. كل تكرار هو نفسه الأسلوب القديم: نموذج انحداري 1 بسيط. و من خلال الحرص على أن يكون كل تكرار أقصر ما يمكن، نأمل أن نتجنب المشاكل المزعجة التي ترافق الانحداريات.



الشكل 3-7: طور البناء و يحتوي على سلسلة من الانحدارات المصغرة.

مع نهاية أكبر عدد من التكرارات سوف نطمح للحصول على منظومة تعمل (مبدئيا بالطبع، ستكون منظومة محدودة جدا في المراحل المبكرة). هذه التكرارات تسمّى تزايدات Increments، ومن هنا أتت تسمية إطار العمل هذا.

4. **التحوّل (الانتقال) Transition**

1. الطور النهائي يتعلق بنقل المنتج النهائي إلى الزبائن. النشاطات المعتادة في هذا الطور تتضمن:
2. الإصدارات المبدئية لأغراض الاختيار من قبل بيئة المستخدم.
3. الاختبارات في الموقع، أو تشغيل المنتج بالتوازي مع النظام القديم الذي سيستبدل.
4. تجهيز البيانات (مثل تحويل قواعد البيانات و صبّها في قوالبها الجديدة، توريد البيانات ، الخ..)
5. تدريب المستخدمين الجدد.

3-5 العملية الموحّدة من أي بي ام The Rational Unified Process (the RUP)

هي إحدى أكثر الأمثلة شهرة للدورة الحياتية التكرارية قيد الاستعمال حاليا. تم تطوير هذه المنهجية من قبل نفس "الأصدقاء الثلاثة" الذين قاموا بتطوير UML ، و لذلك فان RUP متكاملة جدا مع UML.

بصورة أساسية، يقدّر القائمون على هذه المنهجية بأن كل مشروع يختلف عن الآخر، و ذواحتياجات مختلفة، مثلا، بعض المشاريع لا تتطلب إلا طورا قصيرا للاستهلال، بينما المشاريع ذات العلاقة بأمور عسكرية فإن طور الاستهلال فيها قد يمتد لسنوات.

حتى هذه النقطة، تعد RUP مرنة و تسمح بإعادة تكييف كل طور في العملية. أيضا تحدد RUP وبكل عناية قواعد لكل فرد في المشروع و بحسب الاحتياج.

التكرار الواحد يجب عادة ان يمتد من اسبوعين الى شهرين، أية زيادة على شهرين سوف تؤدي الى زيادة في التعقيد و الوصول الى النقطة التي لامناص منها" : الانفجار الكبير "، دوّامة ضم الأجزاء الى بعضها البعض، حيث العديد من المكونات البرمجية ستحتاج الى ان تنتظم و تلتحم لأول مرة.

اذا كبر المشروع و زاد تعقيده فلا يعني هذا أن تكون التكرارات أطول - لأن هذا سوف يزيد من مستوى التعقيد الذي سيكون على المطوّرين التعامل معه في المرّة الواحدة. بدلا من ذلك المشروع الأكبر يجب أن يخصص له تكرارات أكثر.

فيما يلي بعض العوامل التي يجب أن توثر في طول مدّة التكرار:

1. دورات التطوير المبكرة قد تحتاج لأن تكون أطول. هذا يعطي المطوّرين فرصة لأداء اعمال استكشافية على تقنيات جديدة أو غير مختبرة ، أو لتحديد البنية التحتية للمشروع.
2. الأفراد حديثو الخبرة.
3. فرق العمل المتعددة والتي تعمل على التوازي.
4. فرق العمل المتوزعة (في أكثر من موقع).

ايضا ، أريد ان اضم الى هذه القائمة المشروع ذو المراسمية العالية الذي سيحتاج عادة الى تكرارات أطول. و نقصد بالمشروع ذو المراسمية العالية ذلك الذي يتطلّب اصدار وتوفير الكثير من الوثائق الخاصة بالمشروع الى الزبون، أو ربما مشروع يجب ان يلبّي العديد من المتطلبات القانونية . مثال جيد على هذا المشاريع ذات العلاقة بالأمور العسكرية، ففي هذه الحالة فان أعمال التوثيق سوف تزيد من طول فترة التكرار - و لكن كمية التطوير البرمجي الذي يتم التصدي له في هذا التكرار يجب أن يكون في حدوده الدنيا لتجنّب عدّونا الرئيسي و هو عبء التعقيد.

3-5-1 **القيد الزمني**Time Boxing

الأسلوب الأمثل لادارة عملية تكرارية تزايدية هو هو فرض قيد زمني. بهذا الاسلوب الحازم يتم تحديد فترة زمنية ثابتة يجب خلالها اتمام تكرارية معينة.

اذا لم تكتمل التكرارية مع نهاية القيد الزمني، فالتكرارية يتم انهاؤها على أية حال. النشاط الأهم في التقييد الزمني هو المراجعة في نهاية التكرارية. المراجعة يجب أن تبحث في أسباب التأخير، و أن تعيد جدولة الأعمال غير المنتهية و تضمينها في التكرارات القادمة. احدى النصائح لكيفية تطبيق القيد الزمني، أن يكون المطورون هم المسؤولون) أو على الأقل الكلمة العليا (عن تحديد ما هي المتطلبات التي يتم تغطيتها في كل تكرار ، باعتبارهم الذين سوف يلتزمون بهذه الآجال.

الالتزام بالقيد الزمني يعد صعبا، فهو يتطلّب حسّا عاليا بالانضباطية خلال كامل المشروع. من المغري جدا التخلّي عن المراجعة و تخطّي القيد الزمني اذا حان أجل التكرار و كانت نسبة

اكتمال"99%" . حالما يرضخ المشروع لمثل هذا الاغراء و يتم تجاهل مراجعة واحدة ، فان المفهوم بكامله يبدأ في التداعي. اذا تم تجاهل عدة مراجعات ، فان التخطيط للتكرارات القادمة سوف تكون مائعة و تبدأ الفوضى في أخذ مكانها.

بعض المدراء يفترضون ان التقييد الزمني يعيق مرونة الارجاء، هذا غير صحيح، فاذا لم تكتمل التكرارية وقت انتهاء أجل القيد الزمني فإن العمل غير المكتمل يجب نقله الى التكرارات التالية، و يتم اعادة جدولة خطط التتكرارات - يمكن ان يتضمن هذا ارجاء تاريخ التسليم أو اضافة تكرارات أكثر. عموما للتقييد الزمني فوائد هي:

1. الهيكلية الصارمة تفرض عملية التخطيط و معاودتها. فلا يتم التخلّي عن الخطط اذا بدأ المشروع في التمطط.
2. إذا تم فرض التقييد الزمني، تتضائل فرص ان يغوص المشروع في الفوضى اذا ما ظهرت مشاكل، حيث ان هناك دائما مراجعة رسمية منتظمة تلوح في الأفق.
3. إذا ما تسرّب الخوف و بدأ المطوّرون في التخبّط بصورة عشوائية، سيتم وقف هذا التخبّط حالما تتم المراجعة.

بصورة اساسية، يسمح القيد الزمني للمشروع بكامله أن يتريث مرة بعد الأخرى ليحزم أمره من جديد. انه لا يعرقل امكانيات الارجاء، و يحتاج الى ادارة مشروعات قوية كي يعمل بصورة صحيحة.

التوقيتات النمطية للمشروع كم يجب ان يستغرق كل طور من الأطوار الأربعة؟ هذا يتباين من مشروع لآخر، و لكن كمؤشّر عام 10% للإستهلال، 30% للتفصيل، 50% للبناء و 10% للإنتقال

3-6 : مفاهيم المنهج الغرضي التوجه في هندسة البرمجيات

تجسِّد التقانات الغرضية رؤيةً طبيعية للعالم. إذ تصنَّف الأغراض ضمن صفوف وهرميات صفوف. يحوي كل صف مجموعة من الواصفات التي تصِفه ومجموعة من العمليات التي تعرّف سلوكه. وتقوم الأغراض (والصفوف التي تشتق منها هذه الأغراض) بكبسلة المعطيات والإجرائية معاً. إن عمليات المعالجة هي جزء من الغرض، وتبدأ هذه المعالجة بإرسال رسالة إلى الغرض. يكوّن تعريف الصف أساساً لإعادة الاستخدامية reusability في مستويات النمذجة والتصميم والتنجيز implementation.

تؤدي التقانات الغرضية التوجه إلى إعادة الاستخدام reuse، وإعادة الاستخدام (للمكونات البرمجية) تؤدي إلى تطوير برمجي أسرع، وإلى برامج ذات جودة أعلى. إن البرمجيات الغرضية التوجه أسهل صيانةً لأن بنيتها منفصلة الأجزاء بطبيعتها.

هناك ثلاثة مفاهيم هامة تميّز بين المنهج الغرضي التوجه والمنهج التقليدي في هندسة البرمجيات. فالكبسلة encapsulation تحزم كلاً من المعطيات والعمليات التي تتعامل مع هذه المعطيات في غرض واحد مسمّى. والوراثة inheritance تمكّن من توريث الواصفات والعمليات في صف لجميع الصفوف الفرعية والأغراض المستنسخة instances من هذه الصفوف. أما تعدّدية الأشكال polymorphism، فتسمح لعدد من العمليات المختلفة بأن يكون لها الاسم ذاته، وهذا يُخَفِّض عدد أسطر الرماز اللازم لتنجيز النظام، ويُسهّل التعديلات حين الحاجة إليها.

تعتمد إدارة المشاريع البرمجية في حالة المشاريع الغرضية التوجه على المبادئ الأساسية نفسها في الإدارة، ولكن التقنية يجب أن تُكيَّف بحيث تدار المشاريع الغرضية التوجه بطريقة مناسبة. إن إطار العمل في حالة المشاريع الغرضية التوجه ليس نموذجاً خطياً تتابعياً. فالنماذج الخطية التتابعية، تفترض أنّ المتطلبات محدّدةٌ في بداية المشروع، وأن الفعاليّات الهندسيّة تتقدم بطريقة تتابعية خطية. على حين أن هندسة البرمجيات الغرضية التوجه يجب أن تطبِّق نموذجاً يُشجّع التطوير التكراري. ذلك أن البرمجيات الغرضية التوجه تتطور من خلال عدد من الدورات.

هناك «نموذج عوديّ/متوازٍ» recursive/parallel لتطوير البرمجيات الغرضية التوجه. يعمل هذا النموذج وفق الطريقة التالية:

1. القيام بتحليل كافٍ لعزل الصفوف الرئيسة في المسألة والارتباطات الرئيسية؛
2. استخراج الأغراض القابلة لإعادة الاستخدام من مكتبةٍ بغية بناء نموذج أولي تقريبي؛
3. إجراء بعض الاختبارات لكشف الأخطاء الموجودة في النموذج الأولي
4. تسجيل ردود فعل الزبون على النموذج الأولي
5. تعديل النموذج التحليلي بناءً على ما استُنتِج من النموذج الأولي، ومن ردود فعل الزبون
6. تفصيل التصميم لملاءمته مع التعديلات
7. القيام بهندسة بعض الأغراض الخاصة (غير المتوفرة في المكتبة)

يستمر هذا المنهج إلى أن يتحول النموذج الأولي إلى تطبيقٍ إنتاجي. ويتطلب كل تكرار في الإجرائية العودية/المتوازية تخطيطاً وهندسة (تحليلاً وتصميماً واستخراجاً للصفوف، ونمذجة أولية، واختباراً) إضافة إلى فعّاليات التقويم.

3-6-1 التحليل الغرضي التوجه

التحليل الغرضي التوجه هو الفعّالية التقنية الأولى في هندسة البرمجيات الغرضية التوجه. والغاية منه هي تعريف جميع الصفوف (والعلاقات والسلوك المرتبط بها) التي ترتبط بالمسألة المراد حلها.

إن منهج التحليل الغرضي التوجه مختلف اختلافاً جذرياً عن المنهجيات الموجَّهة بالإجرائية كالتحليل البنيوي (منهج تقليدي). وأدى رواج التقانات الغرضية إلى إنتاج العشرات من مناهج التحليل الغرضية التوجه.

على الرغم من اختلاف طرائق التحليل الغرضي التوجه وتباين المصطلحات المستخدمة فيها، فإنها تتضمن الخطوات العمومية التالية:

ـــ الحصول على متطلبات الزبون للنظام.

ــ تعيين حالات الاستخدام use cases (أي المَشاهد التي تصف كيفية استخدام النظام).

ــ استخراج الصفوف والأغراض انطلاقاً من المتطلبات.

ــ تعيين واصفات كل غرض في النظام وعملياته.

ــ تعريف البنى والهرميات التي تنظّم الصفوف.

ــ بناء نموذج لعلاقات الأغراض.

ــ بناء نموذج لسلوك الأغراض.

ــ مراجعة نموذج التحليل مقارنةً بحالات الاستخدام.

3-6-2 : التصميم الغرضي التوجه

الشكل 3-8: هرم التصميم غرضى التوجه

يترجم التصميمُ الغرضي التوجه نموذجَ التحليل الغرضي التوجه للعالم الحقيقي إلى نموذج قابل للتحقيق برمجياً. تُوصف إجرائية التصميم الغرضي التوجه كهرَم مكوّن من أربع طبقات الشكل(3-6). تركّز الطبقة القاعدية في هذا الهرم على تصميم النظم الفرعية التي تحقق الوظائف الرئيسية للنظام. وتحدِّد طبقة الصفوف بنيان الأغراض وهرمية الصفوف اللازمة لتحقيق النظام. وتبيِّن طبقة الرسائل كيفية تحقيق المشاركة بين الأغراض. وأخيراً تعيِّن طبقةُ المسؤوليات الواصفاتِ والعمليات التي تميّز كل صف.

خلال مرحلة تصميم النظم الفرعية، تراعى أربعة مكوّنات: نطاق المسألة (متطلبات الزبون)، التفاعل مع الأشخاص، إدارة المهمات tasks، إدارة المعطيات.

تركّز إجرائية تصميم الأغراض على وصف بنى المعطيات التي تحقق واصفات الصف، والخوارزميات التي تحقق العمليات، والرسائل التي تتيح المشاركات والعلاقات بين الأغراض.

تنتقل البرمجة الغرضية التوجه بالنموذج التصميمي باتجاه التنفيذ. وتُستعمل لغة غرضية التوجه لترجمة الصفوف والواصفات والعمليات والرسائل إلى برنامج قابل للتنفيذ.

هناك طرائق مختلفة للتصميم الغرضي التوجه، بيد أنها جميعاً تتوافق مع هرم التصميم المذكور.

3-6-3 الاختبارات الغرضية التوجه

إن الهدف من الاختبارات الغرضية التوجه يطابق هدف الاختبارات في البرمجيات التقليدية، وهو إيجاد أكبر عدد من الأخطاء بأقل قدر من الجهد. بيد أن الاستراتيجية والخطط في الاختبارات الغرضية التوجه تختلف على نحوٍ ملحوظ.

تبدأ استراتيجية اختبار البرمجيات الغرضية التوجه بالاختبارات الضيقة وتعمل متوسّعةً باتجاه الاختبارات العريضة. وباستخدام المصطلحات المألوفة في اختبارات هندسة البرمجيات[ر]، نبدأ باختبار الوحدة ثم نسير باتجاه اختبار التكامل ثم نبلغ غايتنا باختبار إقرار الصلاحية واختبار النظام.

بعد أن تُنجَز البرمجة الغرضية التوجه، يطبَّق اختبار الوحدة لاختبار الصفوف كلٍ على حدة، ذلك أن نقطة تركيز الاختبار في الحالة الغرضية التوجه هي الصف وليس الوحدة البرمجية (المجتزأ) كما في حالة البرمجيات التقليدية. يَستخدم اختبار الصف العديد من الطرائق، مثل اختبار التجزئة (وهي طريقة تخفِّض عدد حالات الاختبار اللازمة لتجريب صف ما). وتَعتمد كل طريقة على تصميم متتاليات اختبار للتأكّد أن العمليات المكبسلة في الصف قد جُرِّبت. تُفحص حالة الصف (الممثّلة بقيم واصفاتها) لاكتشاف الأخطاء.

يمكن القيام باختبارات التكامل باستخدام استراتيجية معتمدة على النيسب thread. يُكامِل الاختبارُ المعتمِد على النيسب مجموعةَ الصفوف التي تتشارك في الاستجابة لدخل وحيد أو لحدث وحيد. يُكامَل كل نيسب ويختبر إفرادياً.

إن اختبارات إقرار الصلاحية في النظم الغرضية التوجه هي من النوع الموجه بالصندوق الأسود، ويمكن أن تحقق بتطبيق نفس طرائق الصندوق الأسود المدروسة في البرمجيات التقليدية. بيد أن حالات الاختبار هي الموجّه الرئيسي لاختبارات إقرار الصلاحية.

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

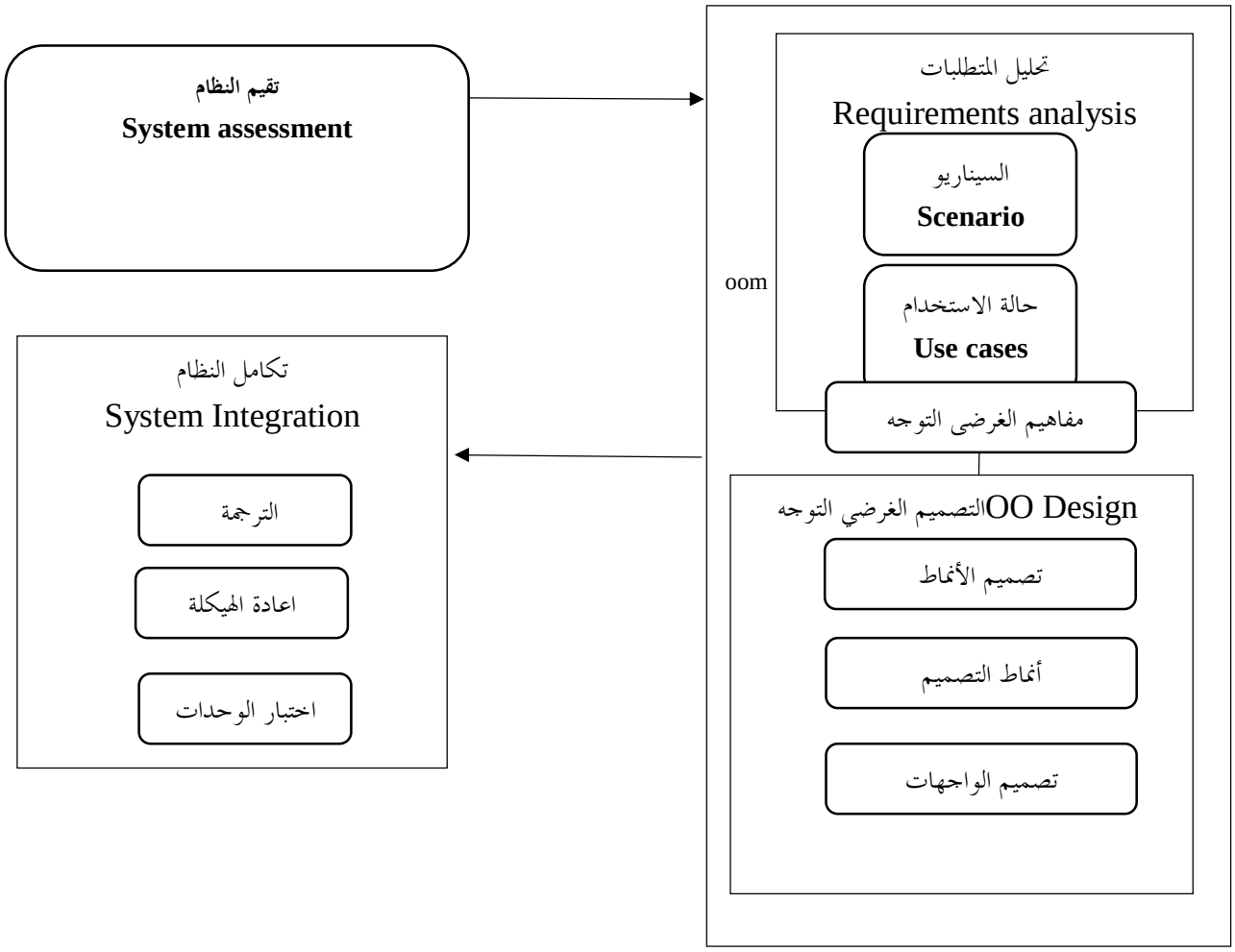
© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الباب الرابع

المنهجيه

المنهجية

المنهجية المطبقة في هذا البحث تتبع الخطوات الموضحه في الشكل (4-1) في القسم 4-1 يتم عرض اساليب تقيم النظام القديم وسيتم شرح تحليل المتطلبات في القسم 4-2 كلا من القسم 4-3 و 4-4 يركزعلي كيف يتم التقاط المتطلبات وتحليل النظام وتحديد الممثلين والسيناريوهات القسم 4-5 يقدم طرق تحديد الكائنات من المتطلبات التى تم التقاطها القسم 4-6 يركز على التقاط اهداف التصميم من خلال المتطلبات والتطبيق من خلال تحديد الكائنات وتنظيمها التى تم تعريفها 4-7 يغطي اعادة تنظيم النظام غير الكائنى المنحنى .



الشكل (4-1): المنهجية المطبقة فى البحث

4-1 تقيم جودة النظام

الدافع لاعادة الهندسية هو حل القضايا للنظام القائم ولكن هذه القضايا قد تكون مختلفة من نظام الي آ خر ، قد يكون هنالك انظمة به عيوب في الهيكل وواجهات المستخدم الفقيرة والبعض الاخر لا زال يستخدم لغات قديمة تتطلب القضايا المختلفة اساليب واستراتيجيات تحديد مختلفة ، لذا فمن المهم معرفة القضايا وفهمه تماما قبل الشروع في وضع اي استراتيجية وعملية تقييم نظام البرمجيات وتحديد القضايا يطلق عليها تقييم جودة النظام للحصول تقيم جودة شاملة للنظام يجب فحص الفهم للنظام (understandability) و الاكتمال (completeness) والايجاز (conciseness) والصيانة (maintainability) وقابلية الاستخدام والموثوقية (usability and reliability).

الفهم (understandability): يشير الي الجهود التى اتخذت لفهم هذه النظام بما في ذلك التعليمات البرمجية وكود المصدر والوثائق .

الكمال (completeness): هو هل كل المكونات الضرورية متاحة ؟ هل لا تفشل اي عملية لعدم وجود موارد او برامج ؟ هل جميع المسارات المحتملة خلال التعليمات البرمجية استأثرت بما في ذلك

معالجة الاخطاء [1]

الايجاز (conciseness): هل كل الكود يمكن الوصل اليه ؟ هل هنالك كود زايد عن الحاجة ؟ هل افرع القرارات معقدة للغاية ؟

الصيانة (maintainability) : هل كل وحدة هي فريدة من نوعها لها وظيفة محددة؟

هل التكييف مع البرامج الجديدة يتم بسهوله في هياكل البيانات ؟هل هنالك من الاجزاء البرمجية المماثلة ؟

القابلية للاستخدام (usability): مدي السهولة في استخدام التطبيق والاستفادة منه . هل واجهة المستخدم الرسومية استخدمت البديهيات؟ هل خط النص مناسب؟ هل تعليمات المستخدم متاحة ؟ هل دليل المستخدم موجود ؟

الموثوقية (reliability): هل هنالك خطر محتمل للحلحقات اللانهائية ؟ التاكد من صلاحية ادخال البيانات ضد نطاق معقول ؟ منع القسمة على الصفر ؟ هل جميع الاستثناءات وضعت فى الاعتبار ؟ هل النتائج صحيحة ؟ هل كل البيانات يتم عمل نسخ احتياطي له ؟

2-4 **استخراج المتطلبات** Requirements Elicitation

متطلبات النظام هي المميزات التى يجب ان يقدمها النظام او المقايس التى يتم التوصيل من خلالها على ان هذا النظام هو مقبول للعميل . وله نشاطين :

تحديد المتطلبات : التى تولد تعريف للمتطلبات التى تكون مفهومة للمستخدم النهائي .

تحليل المتطلبات : ايجاد العلاقات المختلفه بين المتطلبات مما ينشاء النماذج التى تمثل الفهم الصحيح للنظام من وجهة نظر المطوريين ويمكن ترجمتها وتنفيذها بدقة من قبل المطوريين.

في نمذجة الكائنى المنحى انشاء مواصفات المتطلبات هي وحدة من اكثر المراحل صعوبة حيث يتطلب التعاون بين كل المطورين والمستخدمين الذين لديهم عادة خلفيات مختلفة تماما المستخدمين عادة لديهم فكرة جيدة لفهم مجال المشكلة والتوقعات الفعلية لها يجب النظام القيام به ولكن يعرفون القليل جدا عن تطوير البرمجيات من ناحية اخري المطورين هو خبراء فى مجال البرمجيات ولكن لديهم معرفة محددة في بيئة المستخدم .

الفرق فى الخلفيات عن النظام بين المستخدم وفريق التطوير كبير فمن الصعب فى المراحل الاولى للنظام من التعاون التام بينهم لانهم يتكلمون لغات مختلفه.

لعمل توصيل لازالة هذا الفرق بين المستخدم والمطوريين فى المراحل الاولى توخذ المتطلبات وتقوم النشاطات التاليه[2]

- تحديد الجهات الفاعلة Identifying Actors

وهى الوحدات الخارجية التى من شأنها ان تتعامل مع هذا النظام

- تحديد السيناريوهات :Identifying Scenarios

أمثلة وصفيه تبين كيفية استخدام النظام

- تعريف حالات الاستخدام : تلخيص وتصنيف السيناريوهات .
- تحديد أوجه التعاون : العلاقات والتفاعلات بين حالات الاستخدام .
- تحديد المتطلبات الغير وظيفية.

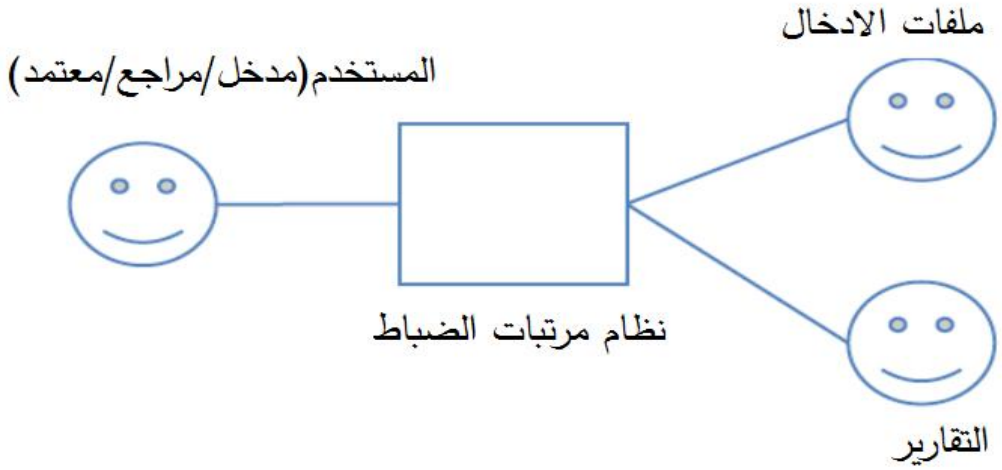
جدول (4-1) القضايا في نظام مرتبات الضباط : متطلبات مرتبات الضباط الرئسى

المستخدم/الزبون	وظائف النظام
المدخل	1.ادخال البيانات الاساسية للضباط
المراجع	2.ادخال بيانات المرتبات
المعتمد	3.اعداد المرتبات الشهرية
المدير	4.اعداد البديل النقدى.
الزبون	5.اعداد الفروقات.
	6.استخراج التقارير المختلفه

4-3 **تعريف الفاعلين** Identifying Actors

الفاعل هو الدور الذى يتفاعل مع النظام والتى يمكن ان تمثل المستخدم النهائي او جهاز او نظام آخر . وتحديد الجهات الفاعلة يوفر االرويه التى تمكن المهندس من تشكيل وجهة النظر الخارجية في نظام قيد التطور وهنالك نظام يمكن ان يكون له اكثر من ممثل بما في ذلك المستخدم النهائي ونظام التواصل . الجهات الفاعلة تتفاعل مع النظام بطرق مختلفة على سبيل المثال يصف الشكل (4ـــ2) الجهات الفاعلة لنظام مرتبات الضباط هنالك ثلاثة جهات فاعلة : الممثل الاول هو المستخدم يستخدم النظام للقيام بعمليات الحالية الممثل الثاني هو ملف البيانات الذي يقدم مساعدة البيانات .

هذه التفاعالات المختلفة تعكس وظائف النظام المختلفة مثلاً التفاعل بين المستخدم والنظام يشكل المتطلبات التفاعلية ووجهة المستخدم البديهة وهى المسئولة من ادخال بيانات المستخدم والعرض للنتائج ويتطلب من النظام المساعدة وهو ما يفسر نظام مرتبات الضباط وادلة المستخدم لذلك من المهم اظهار كل الجهات الفاعلة لمساعدة المطورين لفهم النظام من وجهات نظر مختلفة لتجنب وظائف معينة يجري تجاهلها .



الشكل (4-2): الفاعلين فى نظام مرتبات الضباط .

4-4 **تعريف السيناريو** (Identify Scenarios)

يوفر السيناريو"وصف سردى لما يفعله الناس والخبرة باستخدام تطبيقات الحاسوب " [6] وهى غير محددة بطرق رسميه وصف العالم الحقيقى القائم يعرف بعض سمات النظام.

يتكون السيناريو من سلسلة من السلوكيات للوصول لغرض محدد والتى يمكن ان تكون فى شكل وصف نصى [2] .

يصف الجدول (4-2) سيناريواضافة ضابط جديد من نظام مرتبات الضباط فى هذه الحالة يقوم المستخدم المدخل باضافة بيانات الضابط الجديد هذه حالة محددة جدا من المهم ان نلاحظ يمكن ان يكون المستخدم مراجع او معتمد وسيكون هنالك اختلاف بسيط فى تسلسل الاحداث من المهم ان نفهم ان هذا السيناريو لا يغطى جميع الاحتمالات الممكنه التى يمكن استخلاصها من الشروط الاوليه اذا قام المستخدم بتغير رد الفعل ربما يتغير السيناريو .

جدول(4-2): سيناريو يوضح ادخال بيانات ضابط جديد فى نظام مرتبات الضباط

السيناريو	اضافة ضابط جديد
الحالة	ادخال البيانات الاساسية/بيانات المرتب/حفظ النتيجة
الفاعل	المستخدم هو 1.المدخل/2.المراجع/3.المعتمد
تسلسل الاحداث	1.يقوم المستخدم1 بفتح نظام المرتبات . 2.يختار المستخدم اضافة ضابط جديد . 3.المستخدم يدخل البيانات الاساسية وبيانات المرتب 4.المسخدم يقوم بحفظ البيانات 5.تحفظ البيانات ويعطى المستخدم رسالة بذلك

4-4-1 **تعريف حالات الاستخدام** (Identifying use case)

كما ذكر السيناريو هو الوضع الممكن او حالة من حالات اللاستخدام او بعبارة اخري ، حالة الاستخدام (use case) هي تصميم لجميع السيناريوهات المختلفة تتمحور حول قضية معينة وهنالك حالة استخدام مشابهة للسيناريوا بمعني يبدأ من فاعل ومجموعة من تدفق الاحداث ولكن حالة الاستخدام لا تتحمل معلومات مخصصة مثل اسم المحاسب او نوع المستخدم ولكن الاهتمام المشترك من سيناريوهات متشابهة مثل الخطوات المشتركة لتنفيذ عملية اعداد المرتبات

يصورالجدول (4-3) حالة استخدام اضافة ضابط جديد مقارنة مع الجدول (4-2) فمن السهل نلاحظ ان الاول هو الاكثر تفصيلا.

من الصعب كتابة كل حالات الاستخدام التى تكون صحيحة ومفعلة من البداية ، المطورين ينبقي ان يكتبوا حالات الاستخدام رفيع المستوي ليكون فكرة عامة عن وظائف النظام ومن ثم الصقل خطوة خطوة كما يتم بناء فهم لكل من المستخدم والمطور وعادة ما يستخدم الرسم البياني استخدام

جدول(4-3):حالة الاستخدام لادخال ضابط جديد فى نظام مرتبات الضباط

اسم حالة الاستخدام	اضافة ضابط جديد
الفاعلين مع النظام	المستخدم هو 1.المدخل/2.المراجع/3.المعتمد ملفات الادخال/ملفات البيانات الاساسية ملفات المرتبات الشهرية/العلاوات/الاستقطاعات
تسلسل الاحداث	1.يقوم المستخدم1 بفتح نظام المرتبات . 2.يختار المستخدم اضافة ضابط جديد . 3.المستخدم يدخل البيانات الاساسية وبيانات المرتب 4.المسخدم يقوم بحفظ البيانات 5.تحفظ البيانات ويعطى المستخدم رسالة بذلك
شروط الادخال	1.اكمال كل البيانات المطلوبة 2.يتم الحفظ حسب الصلاحيات
شروط الخروج	1.اكمال كل البيانات بطريقة صحيحه او اعطاء رسالة بعدم الاكمال والخروج
متطلبات الجودة	1.يكون النظام متاح حتى اكتمال كل العملية 2.دقة ادخال البيانات

القضية تحت هذا الوضع يعرب عن انشطة البرنامج من خلال الرموز الرسومية التى هي غالبا ما تكون مفهومة حتى دون اي خلفية لعلوم الحاسوب يتعلق الامر بين الجهات الفاعلة وحالات الاستخدام التفاصيل لا تفسر حالات الاستخدام .

الشكل (4-3): يوضح مخطط حالة الاستخدام لنظام المرتبات

على الرغم من انه يبدو ان مخطط حالة الاستخدام من السهل رسمه لا يحتوى على معلومات وتفاعل اكثر من اللازم وقوتة تكمن في الحفاظ على التوازن العام والكشف عن التكرار عند انشاء العديد من حالات الاستخدام

4-4-2 التقاط المتطلبات غير الوظيفية

المتطلبات الغير وظيفية تعطى مجموعة واسعة جداً من الميزات بما فى ذلك الميزات غير المعرفة في المتطلبات الوظيفية . مثل نمط واجهة المستخدم والاستخدام ، ودليل المستخدم والوثائق وزمن الاستجابة ومع ذلك المتطلبات غير الوظيفة من الصعب التقاطه خلال التطوير لان المتطلبات غير الوظيفة له خلفية مع المستخدم ذات الصلة بما في ذلك مستوي التعليم والعمر والثقافة هذه المعلومات ليست واضحة او سهلة لا يمكن تجاهلة في مرحلة التطوير ولكن من المحتمل جداً ان تنشأ من بيئة ورد فعل المستخدم ومناقشة المنتج والخبرات السابقة للمطورين .

رغم عدم وجود طريقة تقييم منهجي توجد بعض المصادر الارشادية يعتقد ان تكون مفيدة للمتطلبات غير الوظيفة [3] قيود النظام واهداف المنظمة والقيم والتحليل التنافسى للميزات ، قيود المنظمة للتطوير واهداف المنظمة العامة للتطوير .

جدول(4-4): المتطلبات غير الوظيفية لنظام المرتبات.[2]

Example Question مثال للسوال	Category التقسيم
ما هو اسلوب واجهة المستخدم المتوقع من المستخدم ما هي الوثائق التى ينبقي توفيرها للمستخدم	Usability سهولة الاستخدام
هل النظام غالباً ما يتعطل ؟	Reliability الموثقية
كيف يتم الاداء لفترة طويلة ؟	performance الاداء
هل هنالك قيود بيئة للتنفيذ ؟	implementation التطبيق
هل هنالك اي شرط مسبق لهذا النص ؟	interface الواجهة
كيف سوف يتم تقسيم النظام ؟	Packaging

بين مختلف المتطلبات غير الوظيفة تلك التى ترتبط ارتباطاً وثيقاً بسهولة الاستخدام تستحق المزيد من الاهتمام على سبيل المثال تجربة المستخدم باعتبارها مهمة من قياس قابلية الاستخدام ويلعب دوراً حاسماً عندما يقيم المستخدم ويختار نظام البرمجيات في معظم الحالات يتم على الارجح اختيار نظام سهل الاستعمال علي نظام اخر غير سهل الاستخدام .

4-4-3 سهولة الاستخدام usability

سهولة الاستخدام هو شرط غير وظيفي ولكن هو سمة اساسية لا برنامج لكي يكون ناجحاً . ويعتقد أن الهدف من هندسة البرمجيات يجب صنع البرمجيات كونها مفيده للمستخدم ويجب أن يتعين علي سهولة الاستخدام أن يكون "هدف عام يضم كل من وظائف النظام وقضايا واجهة المستخدم .

أنشاء معيار لقياس سهولة الاستخدام ' ومع ذلك ليست سهلة السبب هو أنه من دون ربط البرنامج مع سياق تطبيقه ' القياس غير كافي ولكن الاستدلال التي تبرم من الانشطة النامية يمكن أن تكون بمثابة المبادي التوجيهية لبناء قابلية الاستخدام الجيد وفيمايلي بعض منها في الجدول (4-5)

جدول (4-5) مبادي توجيهية لسهولة الاستخدام

الوضوح البصري visual clarity	ينبغي أن يكون تصميم الواجهة واضح ومرئية بشكل جيد ويجب أن تكون الرسائل مقروءة ومفهومة
الانساق consistency	ينبغي أن يكون ظهور النظام والاستخدامات ثابته
حدسى intuitive	ينبغي أن يكون ظهور النظام والاستخدامات مناسبة لخلفية المستخدم .
رد فعل المعلومات information feed back	ينبغي أن تكون ردود الفعل واضحة ومفهومة ومفيده ' قدمت شرح لاجراءات المستخدم ووضع الاجراءات للمستخدم والاجراءت التي يتعين القيام به بعد ذلك
ملامة الوظائف Appropriate Functionality	يجب أن يكون المستخدم راضياً عن الاحتياجات ومتطلباته
المرونة Flexibility	تصميم النظام يمكن من الوصل الي الوظائف يجب أن يكون بسيط وفعال
الوقاية من الخطا والتصحيح Error pretention and correction	ينبغي للنظان منع الخطا من الحدوث وقنوات وتوفير طرق تصحيح الخطاء للمستخدم
التوجيه والدعم Guidance support	توفير كل التعميمات وفصلها

4-5 **التحليل الغرضي التوجه**(OO Analysis)

في مرحلة استنباط المتطلبات يتم انشاء وصف لمجال المشكلة وبناء التفاهم الاولي في تحليل المنحني الكائني . نوع من الوصف وسوف يتم بناءه لوصف مجال التطبيق وسمي نموذج التحليل المنحني الكائني مقارنة مع نموذج المتطلبات الذي يعطي مظهر خارج النظام .

النموذج يولد الرؤوية الخارجية للنظام بتعريف الكائنات من حالات الاستخدام وسوف يتم عمل ثلاثة خطوات لتوليد نموذج التحليل .

1. تحديد عناصر الكيان و (entity object) و كائنات الحدود (boundary object) وكائنات التحكم (control object) .
2. التفاعلات النموذجية بين الكائنات التى تم تحديدها .
3. حدد الكائنات عن طريق التجريد والتلخيص .

خلال المرحلة الاولي التعيين بين حالات الاستخدام والكائنات التى هي بمثابة تصميم المنحني الكائني موقته وسيتم تعديلها وصقلها الخطوه الثانية التركيز على التفاعلات بين الكائنات من خلال فهم تدفق الاحداث الخطوة الثالثة التحسين على تحديد الكائنات على مبادئ الغرضي التوجه .

4-5-1 تحديد العناصرو الحدود والتحكم للكائنات

(Identify Entity, Boundary and Control Objects)

في هذه الخطوة سوف يتم التفتيش في حالات الاستخدام لاستخراج ثلاثة انواع من الكائنات .

عناصر الكائنات الثابتة التى سوف تشارك فى النظام يمكن ان تكون خصائص النظام ، حالة النظام ، مدخلات المستخدم والمخرجات مثلاً في نظام الايداع توفير خدمة ايداع للمستخدم وتشمل العناصر الكائنات ذات الصلة معلومات المستخدم وسجلات الادخار اعادة رسم الايداع وهو يغطي ايضا بعض الحالات خلال الايداع بجب ان تتبع بعض الحالات لتقدم الحقيقة مثل بداية العملية تحت الاجراء فشل العملية واتمام العملية بنجاح .

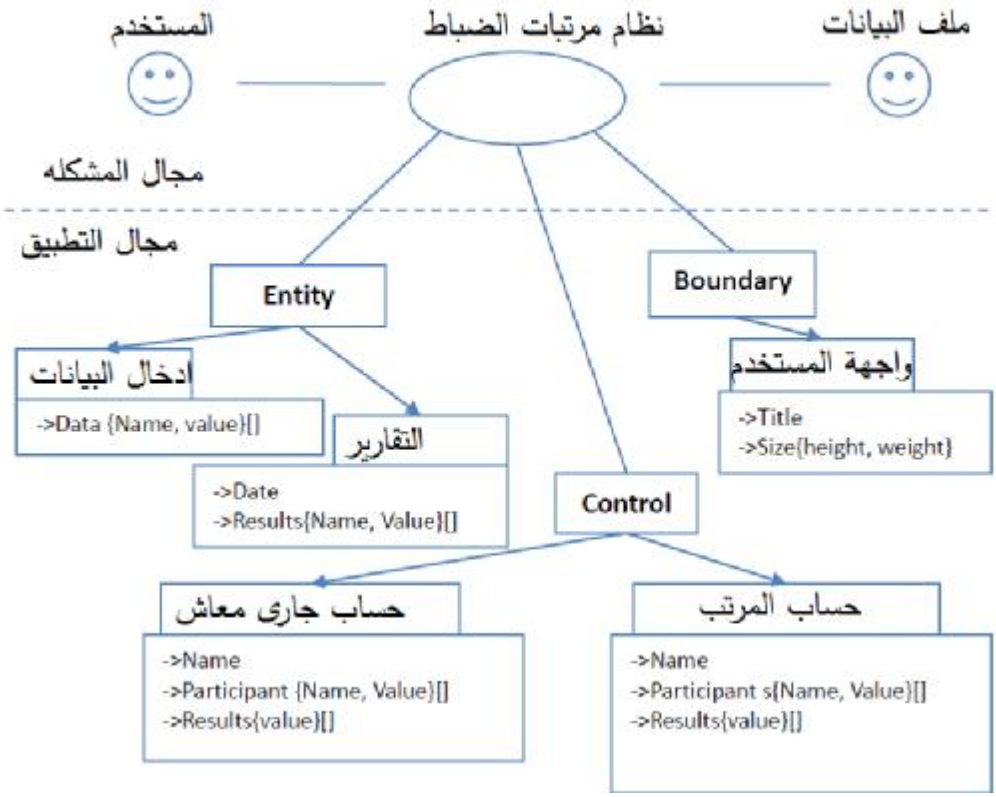
كائنات الحدود boundary object :

هي المسئولة عن التفاعلات بين الفاعلين وحالات الاستخدام اذ ياخذ مدخلات الفاعلين التى تمرر من خلال الكائنات المتعلقة بعملية الادخال وتعود الاستجابة اذ لزم الامر من قبل الجهات الفاعلة مرة اخري بالنظري نفس المثال خدمة الايداع فان كائن الحدود يمكن ان يكون لوحة التحكم ATM الشاشات التى التى تستعمل باللمس والتى توفر وظائف للمستخدم بادخال الاموال وابلاغ المستخدم بوصف المعاملات .

كائنات التحكم (control object) .

كما يدل ذلك من اسمها يفهم المنطق وراء حالات الاستخدام وينسق التعاون بين الكائنات وكائن الحدود مثلا عندما يكون الزبون يريد عمليه الايداع والنظام يتيح الاجراءات التالية: حساب المال المدخل من قبل الذبون وتاكيد المبلغ مع المستخدم وتلقي تاكيد المستخدم وتحديث معلومات ايداع العميل وتسجيل احداث ايداع العميل و نهاية المعاملة. المنطق هو التحكم في نقل الخطوات الي اخري عن طريق السيطرة على عملية الايداع .

دعونا نري مثالا كاملا لكيفية تحديد هذه الكائنات لحالة الاستخدام من برنامج مرتبات الضباط النموذج في الشكل مخطط حالة الاستخدام من نظام مرتبات الضباط يصور الشكل (4ـــ4) نتيجة لتحديد الكائنات من حالات الاستخدام فى نظام مرتبات الضباط اولا يحتاج المستخدم الي ادخال البيانات يدويا او يقوم النطام بتحميل البيانات من ملف البيانات مما يعني بأن يكون هنالك كائن حدود للتعامل مع التفاعلات ذات الصلة بين المستخدم ونظام مرتبات الضباط يمكن تحديد مثل هذا الكائن للحدود هذه باعتباره واجهة المستخدم بعد ادخال البيانات الي النظام سوف يبقي في النظام ويصبح من البيانات الثابتة وفقاً لذلك يجب انشاء كائن الكيان الي يمثل مجموعة بيانات المدخلات لادارة البيانات المدخلة وتوفير واجهة ليتم تغيرها في المستقبل لتنفيذ التقارير وحدة التحكم التى تتحكم باجراء الحسابات المختلفة من عدة مصادروالتاكد من صحته ومن الضروري ان ينشأ تنسيق الحسابات المختلفة والسيطرة عليها ، وخلال هذه العملية هنالك سمات مبدئية يمكن ان توضع مبدئيا علي كل كائن وسوف يتم مراجعتها في وقت لاحق هذا في التخصيص والتعميم .



الشكل (4-4): تعريف الكائنات من حالة استخدام مرتبات الضباط

4-5-2 نموذج التعاون(Modeling Collaboration)

الكائنات تتفاعل مع بعضها البعض بحيث ان وظيف معينة يمكن ان تحقق وتسمي هذه التفاعلات متعاونة . نمذجة التعاون هو عملية فهم واجهات وسلوكيات الكائنات على اساس ذلك لبناء نموذج التعاون على اساس حالات الاستخدام هنالك ثلاثة خطوات [4]

- فهم اشتراك الكائنات يتم تحديدها في احد احداث حالات الاستخدام .

Understand the involvement of identified objects in events of use cases.

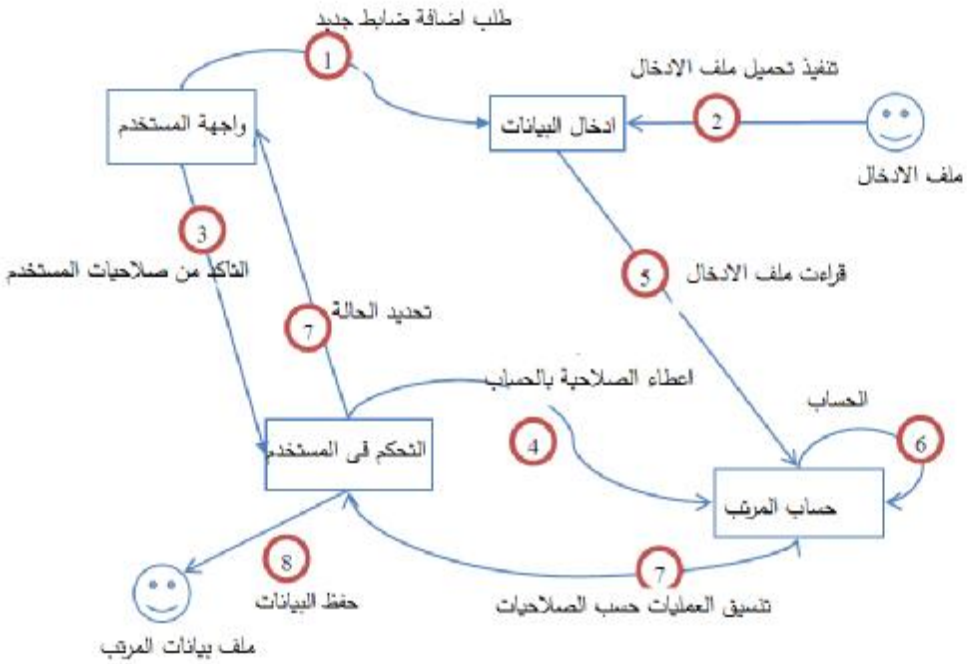
- تغيير احداث التعاون Interpret events to collaborations: لاحظ ان حدث واحد قد يكون قادر على انقسام الي عدة اوجة للتعامل .
- انشاء مخطط التعاون لدمج الكائنات والتعاون .

جدول(4-6):تحويل تسلسل الاحداث الى التعاون

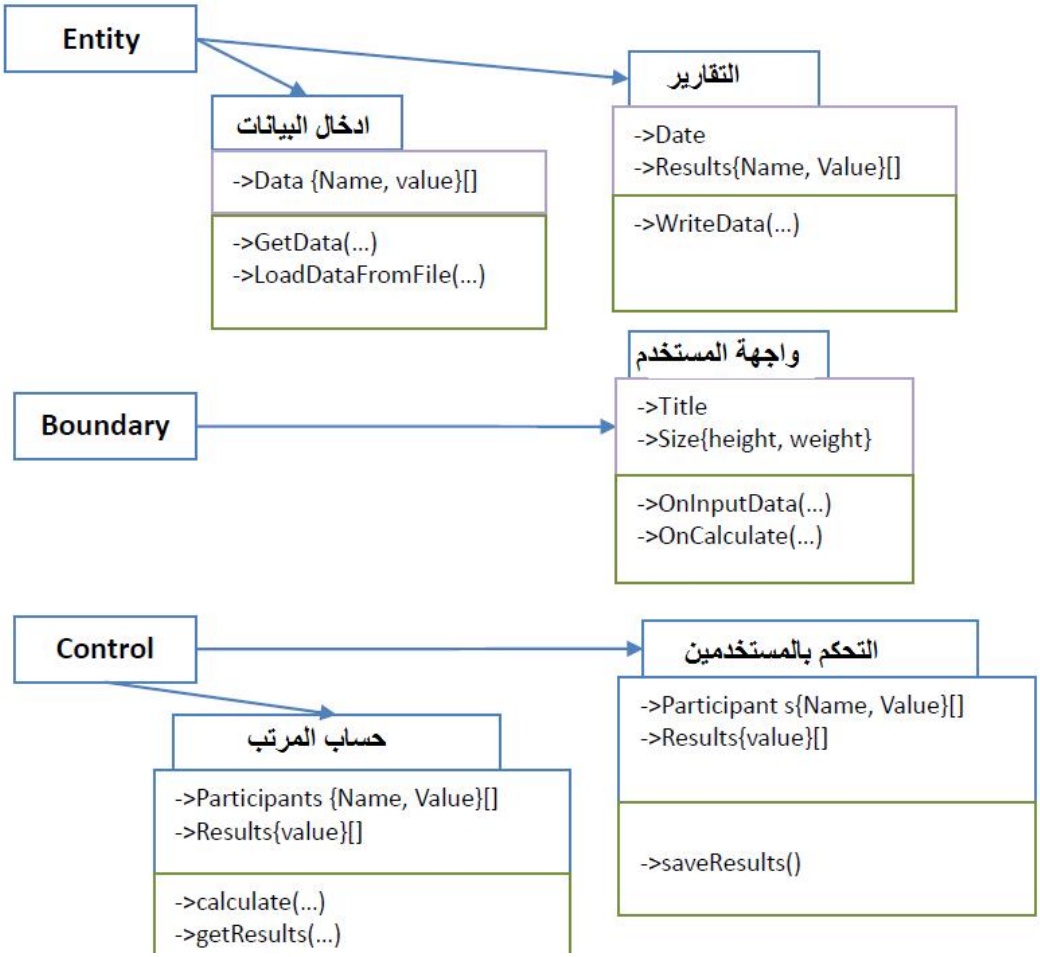
تسلسل الاحداث	التعاون	الكائنات
1.يقوم المستخدم1 بفتح نظام المرتبات	يتم تحميل ملف ادخال البيانات	واجهة المستخدم ملف الادخال
2.يختار المستخدم اضافة ضابط جديد يحمل ملف ادخال البيانات	تنفيذ الملف ملف المستخدمين	واجهة المستخدم التحكم فى المستخدمين
3.المستخدم يدخل البيانات الاساسية وبيانات المرتب	تنفيذ ملف الادخال حساب المرتب	واجهة المستخدم ملف الادخال حساب المرتب
4.المسخدم يقوم بحفظ البيانات	تنفيذ ملف الادخال	واجهة المستخدم ملف الادخال حساب المرتب
5.تحفظ البيانات ويعطى المستخدم رسالة بذلك	يتم التاكد من الصحة ثم الحفظ	واجهة المستخدم ملف الادخال

علي سبيل المثال اذا تم تطبيق الخطوات الاولي في حالة الاستخدام من الجدول(4-3) سوف تكون النتائج كما تدلل محتويات الجدول (4-6) المحور الاول والاحداث نسخ مباشر من الجدول (4-4) المحور الثاني تحتوي على التعاون ، كل مجموعة من التعاون هي حددت من الحدث في نفس الصف ليتكون المحور الثالث من الكائنات كل مجموعة من الكائنات تشارك وتتعاون في نفس الصف واستنادا الي التحليل الواراد اعلاه الرسم التخطيطي للتعاون التي اظهرت في الشكل(4-5) بتحوير المعلومات الواراده في الجدول (4-6) ويكن اكثر وضوحا وفهما وتطبيق الارقام 1ــ8 للاشارة لسلسلة من التفاعلات .

مخطط التعاون يعني ماهي الواجهات/ الاساليب التي ينبغي تعريفه لارتباط الكائنات . علي سبيل المثال يرتبط ادخال البيانات بثلاثة شراكات (1،2،5)



الشكل (4-5) :التعاون فى نظام مرتبات الضباط



الشكل (4-6) :التعاون فى نظام مرتبات الضباط

4-5-3 التعميم والتخصيص Generalization and specialization

التعميم يعني استخدام صفات مشتركة من بعض الكائنات المتجانسه واعادة تعريف تلك الميزات في فئة جديدة بعد هذه العملية الملامح التي تركت في الكائن تصبح اكثر تخصصا وهو ما يسمي بالتخصيص .

4-6 التصميم الغرضي التوجه(OO Design)

هو تحويل نموذج التحليل الكائنى الي التصميم الصحيح وهو التطبيق بالنسبة للمطورين ، يوفر تحليل المنحني الكائن عدد من المشاهدات الخارجية للنظام ، ويساعد علي التواصل بين العميل والمهندسين ، تصمم المنحني الكائن علي العكس من ذلك يوفر المعلومات الداخلية للنظام والتي

تتكون عادة من اهداف التصميم والهندسة المعمارية للنظام . لا ترتبط اهداف التصميم بالمتطلبات الوظيفية والتي هى بالفعل تصمم حسب التصميم المفاهيمى في المنحني الكائن ولكن المتطلبات الغير وظيفية التي لا تغطي حتي الان نحتاج الي ان تحقق في تصميم النظام . بنية النظام هى محددة ومفهومة وايجاد حل لتنفيذها يمكن ان تسند الي المطوريين تنفيذها . الانشطة في تصميم النظام تشمل : تصميم النظم الفرعية ، تحلل النظم الى نظم فرعية ، تصمم الكائنات وتطبيق تدوين اللغة المتعلقة بالبرمجية لتحديد الخصائص والسلوكيات

4-6-1 **التقاط اهداف التصميم**(Capture Design Goals)

وتحدد اهداف التصميم الكيفية التي ينبغي ان تبني البنية الداخلية ، ولها تاثير مباشر على الاستراتيجات التى سيتم اختيارة لتطبيقه على تصميم النظام اهداف التصميم يمكن استنتاجه من المتطلبات غير الوظيفيه على سبيل المثال هنالك تطبيق الالة الحاسبة التى تسمح للمستخدم لتنفيذ الحسابات العديدة في في الاطوار الاولي المستخدم فقط يحتاج الي اوامر الويندوز وفي الوقت نفسه يشير المستخدم بانه في المستقبل تكون تلك الاله ضمن البرنامج .

واجهة المستخدم الرسومية يمكن تشغيلها علي نظام ويندوس واندرويد هذه المتطلبات تشير الي انه يجب على الاختيار الجيد لواجهة المستخدم وتكون محدد لجعل وظائف الحالية يمكن الوصول اليه وتطبيقة من طرق ثالث يمكن تكوين المتطلبات .

- **التمديد** Extendibility

تحديد متطلبات قابلة للتمديد حتى ان لم يكن هنالك مثل هذه المتطلبات الان .

- **المرونة** : **تحديد واجهات كمنتجات كائنات مستقبلة** .

ومع ذلك هذا لا يعني ان جميع المتطلبات غير الوظيفة يمكن ان تدمج في اهداف التصميم . المتطلبات فقط الوظيفة التى تتعلق بالمنحني الكائني ينبغي النظر فيها علي سبيل المثال بعض المتطلبات غير المطلوبة مثل الاداء ووقت المعالجة في وحدة المعالجة المركزية لاضافة اثنين من الاعداد الصحيحة لاينبغي ان تغيير هدف التصميم .

4-6-2 **معمارية النظام** System Architecture

يجب ان تلبي بنية النظام اهداف التصميم التى لديها معاني مزدوجة ،تصميم النظم الفرعية وتصميم الكائنات.

4-6-2-1 تصميم النظام الفرعي Subsystem Design

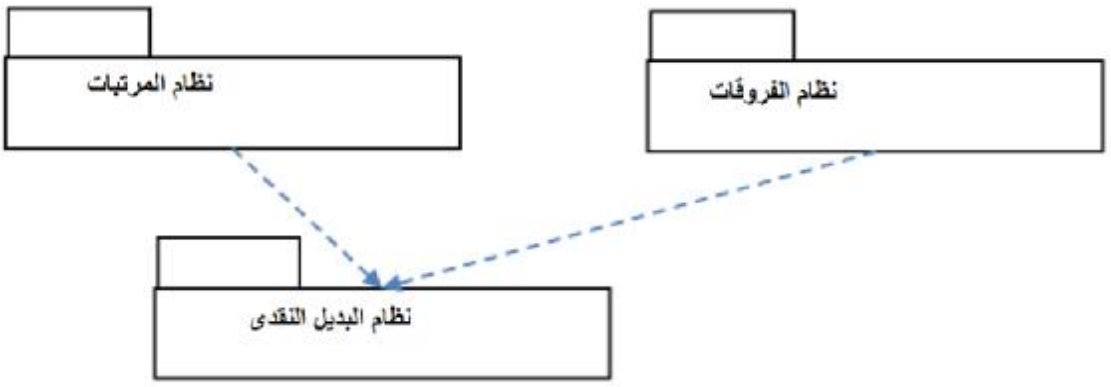
الانظمة الفرعية هي الوحدات الوظيفية التى تمثل النظام باكملة انتهاك النظام عدة انظمة فرعية يمكن ان تقلل من تعقيد تاسيس حل وتنفيذ لان النظم الفرعية مستقلة نسبيا عن بعضها البعض بحيث ان يحل طريق الحل في عدة انظمة في نفس الوقت لجعل التاكد من النظم الفرعية تصميم مكتمل ومستقل بحيث لا يفقد اي تطبيق او يكرر قدمت اثنين من العوامل لقياس جودة التصميم الفرعي :

1/ مستوي الاقتران level of coupling

2/ مستوي التماسك level of cohesion

الاقتران يشير الي التبعيه بين النظم الفرعيه ، مستوي الاقتران نسبه العدد من التبعيات اكبر عدد من التبعيات يدل علي مستوي عالي من الاقتران والمستوي العالي من الاقتران يعني أن تغير النظام الفرعي واحد يمكن ان يؤدي ال تغيرات في العديد من النظم الفرعية الاخري والتي ليست من المتوقع في مجال تطوير البرمجيات .

التماسك (cohesion) يشير الي التبعية بين الكائنات داخل نفس النظام الفرعي المستوي يقاس نفس (النظام) المستوى الفرعي يقاس نفس قياس الاقترت في هذه الحالة ولكن على مستوي عالي ، وهو ما يعني ان الاجسام داخل النظام نفسه هى منطقيا ترتبط مع بعضها البعض .



الشكل (4-7) :الانظمة الفرعية فى نظام مرتبات الضباط .

4-6-2-2 **تصميم الكائن** Object Design

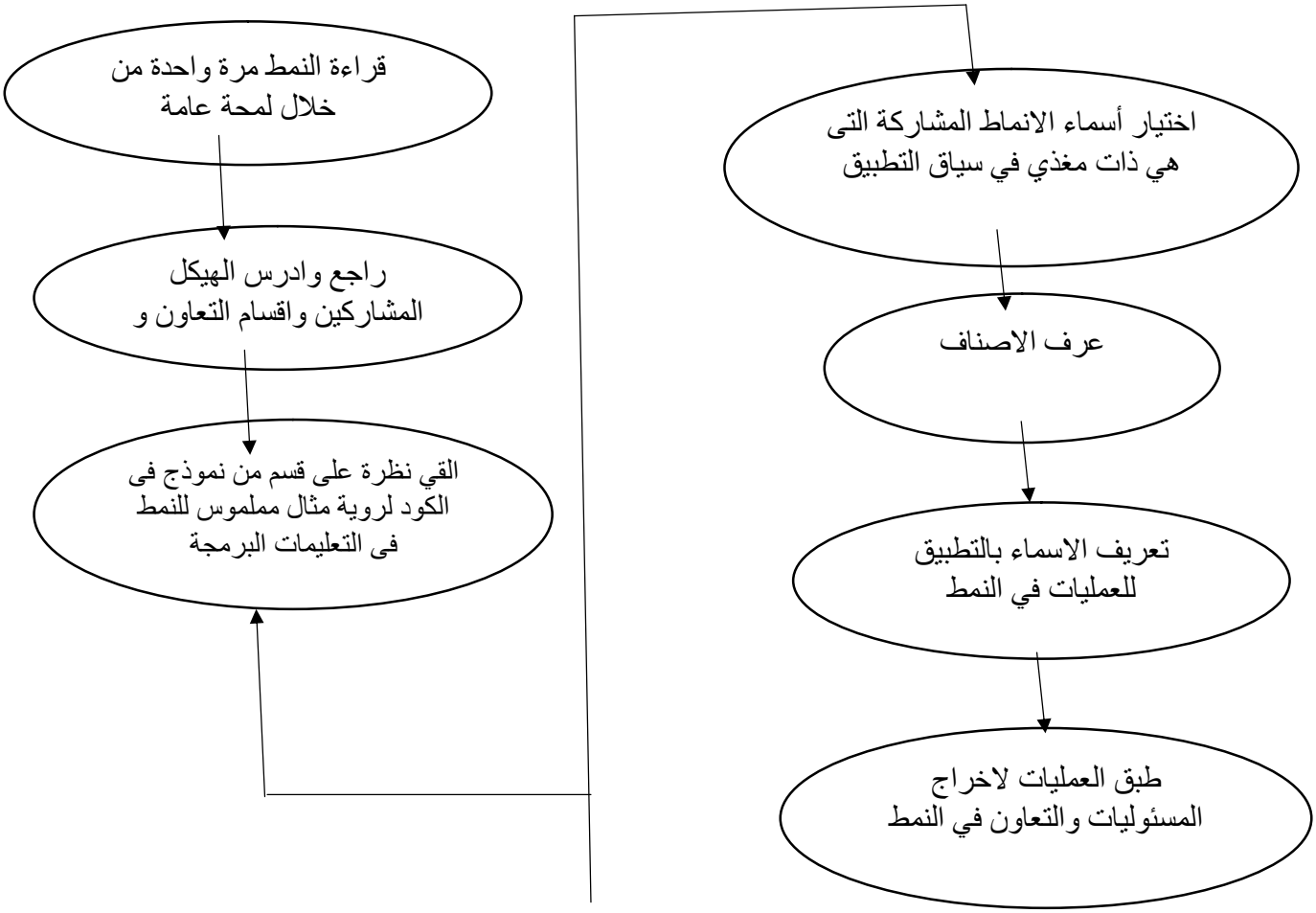
الغرض من تصميم الكائن هو للتحقق من الفجوة بين الكائنات التى كانت محددة في تحليل التصميم الكائني وتصميم النظم الفرعية وتحديد كائنات جديدة لملا الكشف عن الفجوة كما تدرس الكائنات الموجودة وتهذب اذ لزم الامر مهم التحفيز وقياسات تصميم الكائن هي اهداف التصميم ننظر مرة اخري الي الاله الحاسبة على سبيل المثال اذا كان هنالك اي خاصية .المرونة غير مطلوبة شراء التصميم التالي هو مقبول حيث جميع الوظائف الحسابية في جزء لا يتجزء في واجهة المستخدم .

ومع ذلك يجب النظر بمنصة المرونة وهذا التصميم يمكن استبداله على النحوالتالي الذي يضيف واجهة المستخدم للتخلص من واجهة المستخدم من الحل 1 الي الحل 2 يجب ان يتم الاجراء ت التالية:

1. اضافة كائن جديد واجهة محول
2. استبدال التفاعل بين المستخدم والحسابات مع تفاعلات جديدة
3. تعديل سلوكيات واجهة المستخدم لحساب الوظائف .

4-6-2-3 **انماط التصميم** Design Patterns

لتحقيق المرونة الموثوقة بين النظم الفرعية وتصميم الكائنات تطبيق انماط التصميم هي الحلول لتصميم النظام الذي يتم استخلاصة من خبرة التطوير ، وثمة نمط التصميم يلائم السياقات ومجال المشكلة لتحديد انماط التصميم المناسب يتيح هذا البحث المبادئ التوجيهية في الشكل (4-8) [5].



الشكل (4-8) الاجراءات لتطبيق نمط التصميم

من المهم ملاحظة ان الانماط هي حل عام حتى انها قد لا تناسب تماما في بعض السياق وتحتاج الي تهذيب لتكون مناسبة للوضع الخاص كما فى الشكل (4-8) الاجراءات لتطبيق نمط التصميم

4-6-3 تصميم واجهة المستخدم

الهدف هو تصميم واجهات فعالة لنظم البرمجيات ويجب ان يتمركز تصميم الواجهة على المستخدم وان الواجهة متماسكة ومنطقية وقادرة على توفير المساعدة للمستخدم للتغلب على الأخطاء.

تقود مستخدمى النظام عبر واجهات هذا النظام بدلا من الوظيفة,ويمكن أن يسبب تصميم واجهة ضحلة قليلة الامكانيات قيام المستخدم بارتكاب أخطاء فاجعة ,وقد يكون سبب تصميم هذه الواجهة الفقيرة سببا فى عدم الاستخدام النهائى للعديد من البرمجيات.

4-6-3-1 واجهة المستخدم الرسومية

يتفاعل معظم مستخدمى نظم الاعمال مع هذه النظم من خلال واجهات رسومية على الرغم من انه فى بعض الأحيان قد تكون هنالك واجهات تقليدية قديمة موروثة معتمدة على النص .

4-6-3-2 التصميم المتمركز على المستخدم

هو منهجية تصميم واجهة المستخدم حيث تكون حاجات المستخدم الحاكم الاعلى وحيث يتورط المستخدم فى عملية التصميم ويتعلق تصميم الواجهة دائما بتطوير واجهات نماذج أولية .

4-6-3-3 مبادى تصميم واجهات المستخدم

يجب أن يأخذ فى الاعتبار احتياجات وخبرة وامكانات مستخدمى النظام ,يجب ان يهتم مصمم النظام بطبيعة الاشخاص وحدودهم العقلية ويجب عليه ادراك ان الاشخاص معرضون لارتكاب الاخطاء سهوا او بدون قصد وهنالك عدة مبادى تحكم تصميم الواجهة ولو انها قد تكون غير قابلة للتطبيق على كل التصميمات.

4-6-3-4 تفاعل المستخدم والنظام

يجب تقرير نوع المشكلة التى تنجم عن تصميم النظم التفاعلية مثل:

1.كيفية وجوب تزويد المعلومات من المستخدم .

2. كيفية وجوب تزويد المعلومات من المستخدم الى نظام الحاسوب .

3. كيفية وجوب تقديم المعلومات من نظام الحاسوب الى المستخدم .

4-6-3-5 دعم المستخدم

يغطى ارشاد المستخدم كل امكانات النظام لدعم المستخدين بما فى ذلك التعليمات الفورية ورسائل الخطا ودليل الاستخدام. يجب ان يتكامل نظام ارشاد المستخدم مع واجهة المستخدم لمساعدة المستخدمين عندما يحتاجون معلومات عن النظام او عندما يقومون بارتكاب بعض الاخطاءيجب ان يكون نظام التعليمات والرسائل متكاملا ومجمعا قدر الامكان .

4-6-3-6 رسائل الخطا

تصميم رسائل الخطأ من الموضوعات ذات الاهمية الحرجة فرسالى الخطا الفقيرة قد تجعل المستخدم يرفض النظام بدلا من قبوله ويجب ان تكون الرسالة محددة وواضحة وبناءة.

4-6-1 توثيق المستخدم

يجب توفير معلومات ورقية مع النظام ويجب تصميم المستندات لمدى واسع من المستخدين من مستوى عديمى الخبرة الى الخبراء وتكون الوثائق وفقا للانواع الاتية :

1. وصف وظيفى :وصف موجز لما يمكن ان يفعله النظام .
2. دليل استخدام تقديمى : مقدمة للنظام .
3. مرجع دليل استخدام النظام : وصف كل امكانات النظام بالتفصيل .
4. دليل تثبيت النظام وصف كيفية تثبيت النظام .

جدول(4-7): مبادي تصميم واجهة المستخدم

الالفة للمستخدم User familiarity	ينبغي أن يستند على المصطلحات المفهومة للمستخدم وليس مصطلحات الحاسوب
الاتساق consistency	ينبغي أن يكون ظهور النظام والاستخدامات متناسقة القوائم والارقام
تقليل المفاجاة Minimal surprise	ان يعمل بطريقة معروفة تمكن المستخدم من التنبو عند المقارنة .
الاسترداد Recoverability	يجب توفير نوع من المرونه عند خطاء المستخدم يمكن من الرجوع وتصحيح الخطا
توجيه المستخدم User guidance	يجب أن يتم توفير بعض التوجيهات المستخدم مثل أنظمة المساعدة، الكتيبات، الخ
التنوع للمستخدمين User diversity	ينبغي الدعم المرافق لتفاعل انواع مختلفة من المستخدمين. على سبيل المثال، بعض المستخدمين قد تشهد صعوبات وهكذا ينبغي أن يكون النص أكبر متاح

4-7 الانظمة لنظم الموزعة

معظم الأنظمة الحالية تعتبر انظمه موزعه والأنظمة الموزعة هي عباره عن انظمه تعمل على العديد من الأجهزة مرتبطة مع بعض بشبكه.,وتدعم النظم الموزعة مشاركة الموارد ,الانفتاح والالتقاء والتزامن والتعاون والنمو وسماحية الخطأ والشفافية.

4-7-1 دوافع توزيع الأنظمة ومزاياها

1. **الموثوقية**Reliability

كان مفهوم الموثوقية متخصصا بعالم التصنيع فعندما نقول تجهيزات ذات وثوقيه عالية يعني انها لا تتعطل, ومن ثم انتقل هذا المصطلح الى البرمجيات وظهر لنا سؤال طالما حيرنا وهو متى نعتبر النظام موثوقا؟ يختلف مفهوم الموثوقية حسب نسبة الخطأ المقبولة , لذا كان الوصول الى الموثوقية امرا صعبا بالطرق التقليدية ولرفع الموثوقية نسبة 20% يجب رفع الكلفة 50% من هنا ظهرت فكرة رفع الموثوقية من خلال الأنظمة الموزعة فبدلا من رفع وثوقيه كل جهاز يتم استخدام اجهزة عده وفي حالة تعطل واحد يعمل الاخر.

2. **التشارك بالمصادر** Sharing of resources

كان هناك ضرورة كبيرة للتشارك بالمصادر لأنها كانت مكلفة فظهرت فكرة توزيعها .

3. **تجميع القدرة الحاسوبية** Aggregate computing power

الحصول على قدره هائلة والاستفادة منها واستغلالها وذلك عن طريق توزيع العمل على اجهزه .

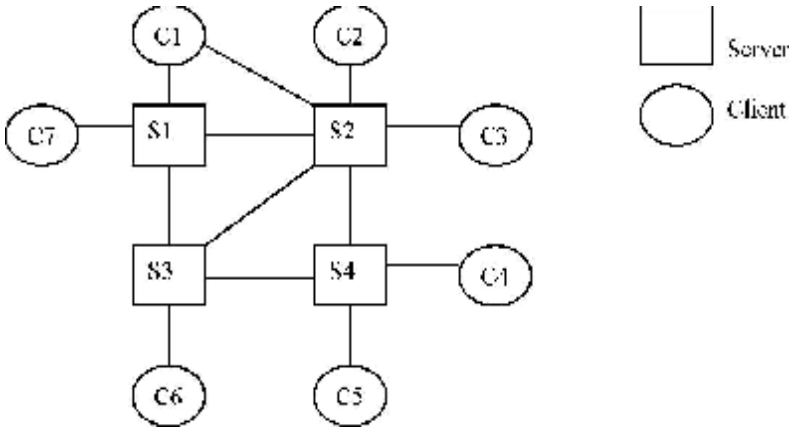
4. **الانفتاح وقابلية التوسع Openness / Scalability**

تعتبر الأنظمة الحاسوبية انظمه قابلة للتوسع ومفتوحه حيث يمكننا في لحظه تطويرها والتكامل معها .

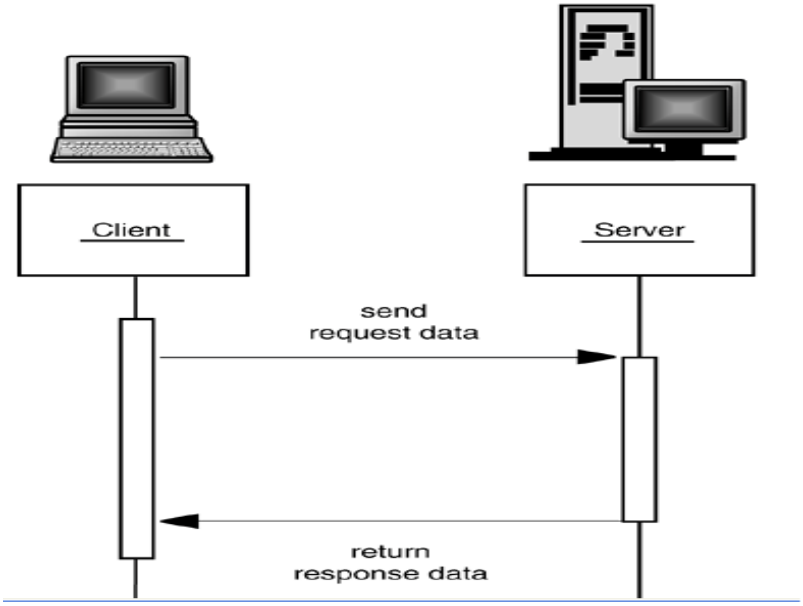
4-7-2 معماريات الخادم العميل

تتم نمذجة التطبيق كمجموعة من الخدمات التى توفرها أجهزة الخدم وكمجموعة من العملاء التى تستخدم هذه الخدمات.

يعرف العملاء الخدم لكن ليس من الضرورة ان يتعرف الخدم على العملاء وكل من العملاء والخدم عبارة عن عمليات معالجة منطقية وليس شرطا ان يكون تخطيط المعالجات للعمليات بنسبة1:1 .



الشكل (4-9) مخطط يبين برمجة معمارية المزود والزبون



الشكل (4-10) **يوضح حوار الزبون / المزود**

4-7-3 **طبقات التطبيق** application layer

عند تصميم نظام الزبون / المزود يتم تقسيم النظام تطبيقيا الى ثلاث طبقات توضح كما يلي

تقسم التطبيقات الى ثلاث طبقات :

1-طبقة العرض

وفيها يتم عرض البيانات وتوفر واجهة للمستخدم النهائي

2-طبقة المعالجة

وفيها يتم القيام بعمليات المعالجة واجراء العمليات الحسابية

3-طبقة ادارة قواعد البيانات

وفيها يتم ادارة قاعدة البيانات و تقديم البيانات المطلوبة من قبل الزبون طبقات التطبيق.. لماذا؟

نحن نعرف ان الأنظمة الموزعة من احد مشكلاتها الأمنية ,ان من حلول تعزيز الأمني في الأنظمة هو تقسيم النظام الى عدة طبقات بحيث يتم وضع الاجزاء المهمة في الطبقات الدنيا للنظام وبهذا نستطيع وضع قوانين ومنهجيات للوصول المخول لمثل هذه الاجزاء ومن هنا جاءت معمارية الطبقات الثلاث للتطبيق.

وعلى اساس طبقات التطبيق تنوعت معماريات العميل الزبون الى نوعين هما

1. Two-tier C/S architecture
2. Three-tier C/S architecture

في Two-tier architecture يتم توزيع الطبقات بنسبة2:1 وهذا يعني انه على المزود او الزبون ان ياخذ طبقتين ويأخذ الاخر الطبقة الاخرى وقد انبثق عنها معماريتين حسب

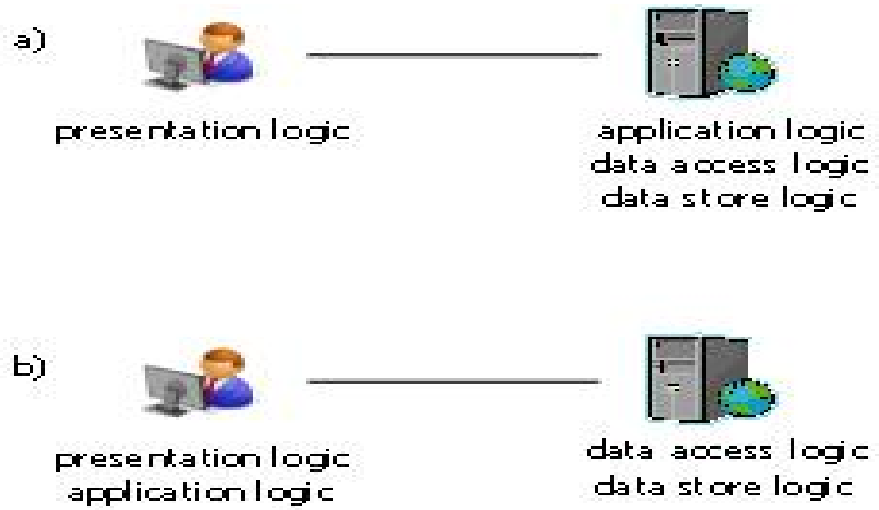
عدد الطبقات التي على الزبون هما

1. معمارية العميل النحيل THIN CLINT architecture
2. معمارية العميل البدين FAT CLINT architecture

4-7-4 العميل النحيل والعميل البدين

نموذج العميل النحيل يتم فيه تنفيذ كل معالجة التطبيق وادارة البيانات على الخادم ويكون العميل مجرد مسئول فقط عن تشغيل برمجيات التقديم أو العرض, يستخدم نموذج العميل النحيل عند تحويل النظم القديمه الموروثه الى معماريات الخادم العميل وتعمل النظم الموروثة كخادم من جانبها بواجهة رسومية تنفذ على العميل والعيب الكبير هو انها تضع عبء المعالجة على كل من الخادم والشبكة .

اما فى نموذج العميل البدين يكون الخادم مسئول فقط عن ادارة البيانات وتنفذ البرمجيات الموجودة على العميل التطبيقات والتفاعلات مع مستخد النظام,فى نموذج العميل البدين يقوض العميل للقيام بمهام معالجة اكثر حيث تنفذ معالجة التطبيق محليا ويناسب اكثر نظم الخادم العميل الجديد حيث تطور قدرات لكنها اكثر تعقيدا من النموذج النحيل خاصة فى الادارة يتبنى هذا البحث هذه المنهجيات لانه يقوم على اعادة هندسة الانظمة القديمة الموروثة .

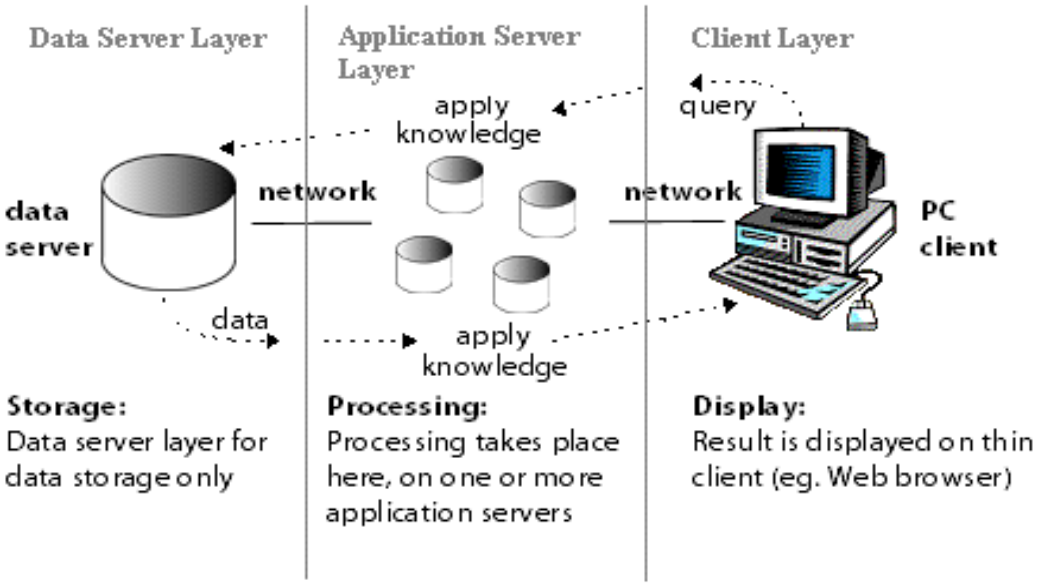


الشكل (4-11) **يوضح معمارية العميل البدين والنحيل.**

4-7-5 معمارية الثلاثة طبقات Three-tier C/S architecture

جاءت هذه المعمارية بحلول لكل مشاكل معمارية Two-tier C/S architecture

واحد امثلة هذه المعمارية هي شبكة الانترنت والشكل التالي يوضح هذه المعمارية



الشكل (4-12) معمارية الثلاثة طبقات.

8-4 اعادة تنظيم النظام غير الغرضي التوجه Non – OO System Reorganization

في اعادة هندسة النظام الموروث قد لا يكون المنحنى الكائنى ينطبق على جميع النظم الفرعية علي سبيل المثال النظام الفرعي المكون بلغة غير كائنة مثل C لا يدعم ميزات المنحنى الكائنى ولكن ميزات المنحنى الكائنى يوجد تقليد بعض الملامح مثل التغليف التى تمتدي في بعض الاحيان لتعزيز التنظيم والفهم .

والسؤال هو كيفية ادخال الكبسلة الي لنظام غيري كائني ؟ الفرق واضح بين النظام الكائني والذي لا يدعم البرمجة الكائنة لتحقيق نفس المتطلبات النظام غير الكائني يطبق في شكل وظائف / وانطمة فرعية ولكن نظام المنحني الكائني ينفذ مجموعة من الاصناف والاصناف تنظيم بمجموعة من الخصائص والسلوك . ويبدوانها مختلفة تماماً ولكن لديه سمات مشتركة وهذا هو احد العناصر الاساسية في كل المنهجيات هي الوظيفة في غير البرمجة الكائنية النظام عبارة عن مجموعة الوظائف وهي مجموعة من الدوال والمتغيرات في تصميم المنحني الكائني بتكوين من تفاعلات الاصناف ، كل الاصناف تقوم بكبسلة لكل الدوال والمتغيرات بالرغم من اختلاف تنفيذ الدوال إذ هنالك بعض الوظائف التى يتم تنفيذها بشكل جيد لتحقيق الصفه الاساسية الموجودة في الانظمة غير الكائنية يمكن تنظيم هذه الوظائف في صنف . ايضا اذ كان هنالك وظائف تعمل على مجموعة من المتغيرات في التصميم غير الكائني فانه يعني انه متماسك وربما ينتهي الي نفس الفئة في تصميم المنحني الكائني بطبيعة الحال فانه لا يعني ان الاصناف يمكن بناءوها من قبل ببساطة ووضح الوظائف معا ويمكن تطبيق بعض مبادي المنحني الكائني للنظام غير الكائني الموجود . من غير قصد يمكن ان يتم الكشف عنها وتستخدم للمساعدة في تصميم المنحني الكائني .

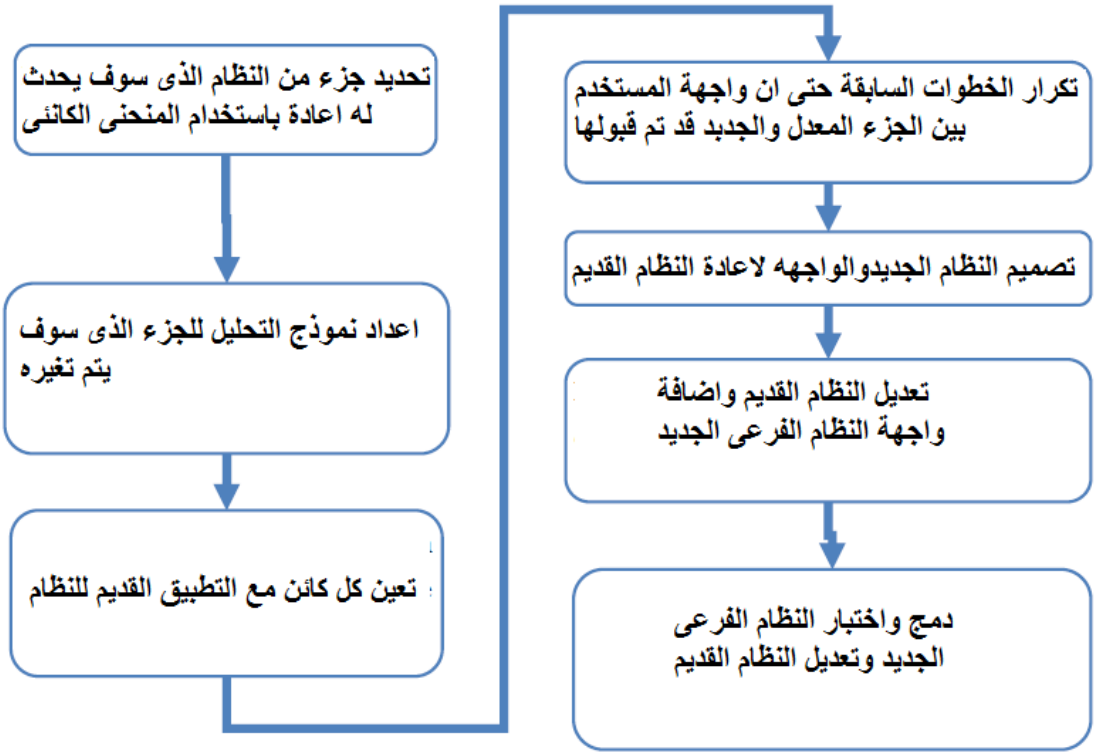
وعلاوة على ذلك حتي عيوب النظام القائم يمكن استخدامها فى تصميم النظام الجديد على سبيل المثال عيب واحد مشترك هو الازدواجية والرمز يتكرر يمكن الحل من خلال الصيانة والمرونة يحدث ذلك على مستوي الكود لا يتم منعك من تصميم النظام على مستوي عالي فاستولت على كود الازدواجية القائمة معرفة رموز الاذدواجية القائمة يمكن ان توفر المهندسين ذوي الخبرة في اجزاء ما يمكن ان يسبب الاذدواجية عندما تنفذ نفس المنطق في النظام الجديد وبالمثل

اقتران ضعيف بين الوظائف يمكن ان يحدث فى تصميم المنحني الكائني ايضا البحث في هذه الحالة يوفر نظرة شافية لنطاق المشكلة ويساعد في تصميم هيكل انيق .

والسؤال التالي هو كيفية استخراج هذه المعلومات ؟

يكون ذلك باتباع الخطوات التى يوضحه الشكل الشكل (4-13) وفقا للخوارزمية فى الشكل الشكل (4-1) فنجد انه الخطوة الاولى هى تحديد الجزء من النظام الذى سوف يحدث له اعادة هندسة باستخدام تقنيات المنحنى الكائنى ثم بعد ذلك تبدا الخوارزمية فى الشكل (4-1) بعمل اول خطوة وهى تقيم النظام المحدد لاعادة الهندسة ثم اعداد نموذج التحليل للجزء الذى سوف يتم اعادته

وهذا يتم وفقا للخطوة الثانيه فى الشكل (4-1) وهى التحليل الغرضي التوجه وبعد تحديد الكائنات يتم تعين كل كائن فى النظام الجديد مع التطبيق فى النظام القديم ثم تصميم واجهة المستخدم فى الخطوة الثالثة فى الشكل الشكل (4-1) وهى التصميم الغرضي التوجه والاستمرار فى تكرار العمليات حتى الوصول لواجهة المستخدم الجديده المقبوله ومن ثم تعدبل النظام القديم ودمج واجهة المستخدم الجديده ومن ثم الخطوة الاخيرة فى الشكل (4-1) وهى دمج واختبار النظام الفرعى الجديد وتعديل النظام القديم



الشكل (4-13) اعادة هندسة النظام غير الغرضي التوجه

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الباب الخامس

دراسة حالة

5-1 مقدمة

سوف يتم انشاء اطار لتطبيق منهجية الغرضى التوجه فى اعادة هندسة برمجيات النظام القديم حسب المنهجية فى الشكل (4-1) وهى المنهجية المطبقة فى البحث ولكن هل سوف تطبق لكل النظام فى مرة واحدة ؟ والاجابة هى لا سوف تطبق حسب المنهجية فى الشكل (4-13) اعادة هندسة النظام غير الغرضي التوجه التى تبدا بتحديد جزء من النظام الذى سوف يحدث له اعادة هندسة ثم اعداد نموذج التحليل للجزء المراد اعادة الهندسة له والتصميم الغرضى التوجه وتصميم الشاشات ثم التحقق واختبار النظام وذلك باستخدام المنهجية فى الشكل(4-1) ونتحقق فيه من كل المطلوبات ولكن فى البداية سوف يتم الغاء نظرة عامة على كل النظام ثم اختيار نظام مرتبات الضباط لتطبيق المنهجية .

5-2 دراسة حالة (case study)

دراسة الحالة التى اجريت طوال هذه الاطروحة على مشروع اعادة هندسة النظام المالى بالادارة العامة لادارة للشئون المالية بوزارة الدفاع بتطبيق منهجيات الغرضي التوجه التى نوقشت في الفصل الرابع سوف يتم اجراء بعض التعديلات على المنهجية لتناسب هذا المشروع.

5-3 مقدمة عن النظام

لقد بدأت الادارة العامة للشئون المالية بوزارة الدفاع فى مواكبة هذا التطور فى عام 1997 م وذلك بحوسبة مجموعة كبيرة من انظمة العمل المالى واعداد البنيات التحتيه له والكوادر الموهلة لذلك العمل .

ففى العام 1997 م كانت البداية بحوسبة الفصل الثانى وحققت التجربة نجاحا مقدرا وكبيرا فى تنفيذ الاهداف المخطط لها والمتمثلة فى سرعة الاجراءات وتامين المعلومات وسلامتها واستخراج البيانات المطلوبة بسهولة ويسر .

وكانت التجربة الثانية فى العام 1999 م بانشاء مكتب الرصد والمتابعة لادخال بيانات الفصل الاول مع سجل الافراد وحقق البرامج اهدافه فى ضبط المرتبات ومحاصرة المخالفات المالية وحصر القوة .

المرحلة الثالثة كانت مرتبات الضباط ثم بعد برامج الميزانية ثم برامج المعاشات ثم بعد ذلك تم حوسبت معظم العمل المالى واخره هو الدفع الالترونى لمرتبات ضباط الصف والجنود .

5-3-1 انظمة الادارة العامة للشئون المالية

تتكون الادارة المالية على عدد كبير من الانظمة وهى :

1.نظام الميزانية

وهو وهو يقوم باعداد الميزانية والصرف على الفصل الاول والثانى وهو يربط الادارة المالية بوزارة المالية ويوزع على كل من القوات البرية/القوات الجويه/القوات البحريه/القوات المشتركة ووزارة الدفاع. ويحتوى على عدد كبير من الانظمة الفرعية .

2.نظام المحاسبة

وهو وهو يقوم باعداد الحسابات والصرف على الفصل الاول والثانى وهو يربط ببرنامج الموازنة ويوزع على كل من القوات البرية/القوات الجويه/القوات البحريه/القوات المشتركة ووزارة الدفاع. ويحتوى على عدد كبير من الانظمة الفرعية .

3.نظام الرصد والمتابعه .

هذا النظام يقوم بعملية متابعة الصرف للفصل الاول لضباط الصف والجنود لمراقبة التعديل والحركة الدوريه للزيادة والنقصان وهو من الانظمة القديمة وسوف يتم الاستغناء عنه عند اكتمال مرتبات الصف والجنود لكل الوحدات .

4.نظام مرتبات الضباط

وهو وهو يقوم باعداد المرتبات والفروقات والبديل النقدى لكل الضباط بالقوات المسلحة وهو يربط ببرنامج البنوك الخارجيه ويوزع على كل من القوات البرية/القوات الجويه/القوات البحريه/القوات المشتركة ووزارة الدفاع. ويحتوى على عدد كبير من الانظمة الفرعية .

5.نظام مرتبات الصف والجنود

6.نظام السجلات .

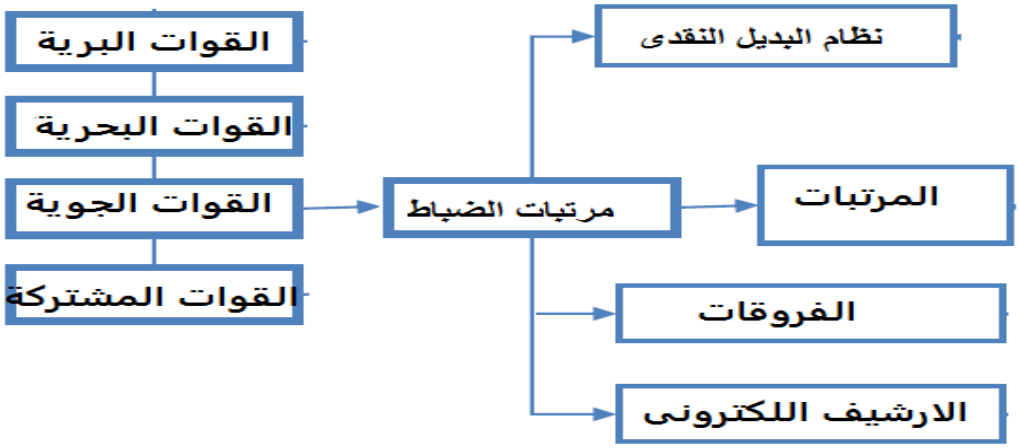
7.نظام المعاشات .

8.نظام المرجعة .

المنهجية المتبعة تعتمد على اختيار نظام فرعى لاعادة هندسته ووفقا لذلك سوف نختار فى البداية نظام مرتبات الضباط وذلك لاكتمال هذا النظام وتوزيعه على القوات المختلفه وهو يقوم بارسال المرتبات الى البنوك المختلفه داخل ولايات السودان المختلفه عن طريق بنك السودان المركزى

5-3-1-1 انظمة مرتبات الضباط

يقوم النظام باعداد وتحسيب المرتبات للضباط والبديل النقدى واى فروقات للعلاوات ويتكون النظام من اربعة مكونات رئيسيه :مرتبات الضباط والبديل النقدى والفروقات والارشيف الالكترونى ويوزع على كل من القوات البريه والبحريه والجويه والقوات المشتركة مع اختلاف قى بعض اسماء العلاوات والوحدات والفئات لكل نوع من القوات الشكل(5-1) يوضح الوظائف الرئسيه والفرعيه لنظام المرتبات.



الشكل (5-1) انظمة مرتبات الضباط

5-3-2 نظام مرتبات الضباط

يقوم نظام المرتبات باعداد المرتبات فى البداية عند دخول ضابط جديد يقوم بادخال البيانات الاساسية وبيانات المرتب بواسطة المدخل ثم تراجع بواسطة المراجع وتعتمد بواسطة المعتمد عند الاعتماد يتم انزال بيانات الضابط كبيانات اساسية معتمدة والمرتبات الشهرية ويتم تحديد الاستقطاعات والاستحقاقات حسب الرتبة من البينات المسبقة لكل من العلاوات والاستحقاقات لكل رتبه والمرتب الاساسى المعد مسبقا

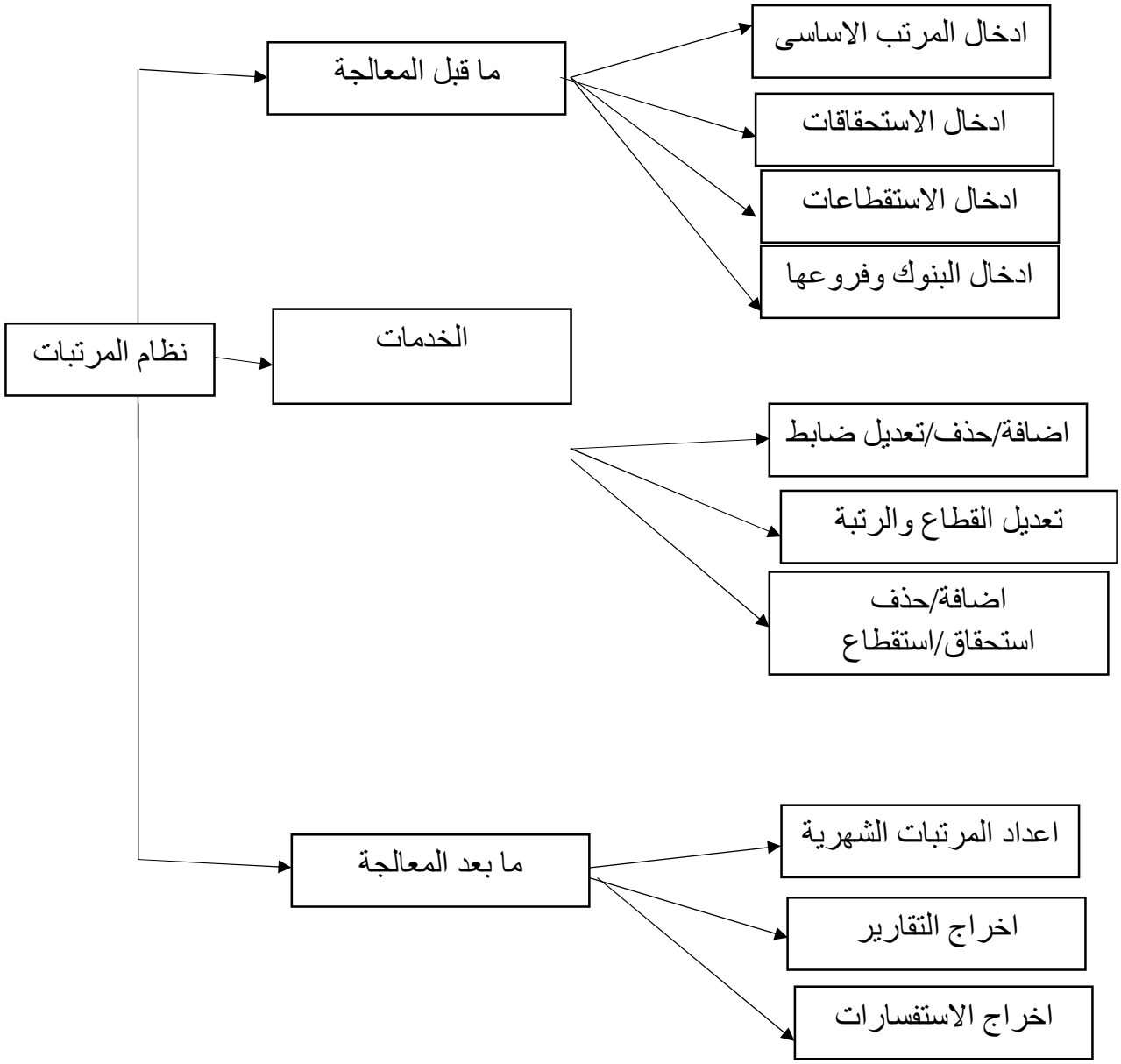
5-3-2-1 نظرة الي نموذج الوظائف :

يتكون هذا النظام من ثلاثة مكونات ادارية هي : وظيفة ما قبل المعالجة إعداد كل البيانات التى يحتاج اليه النظام قبل اجراء المعالجة، الخدمات

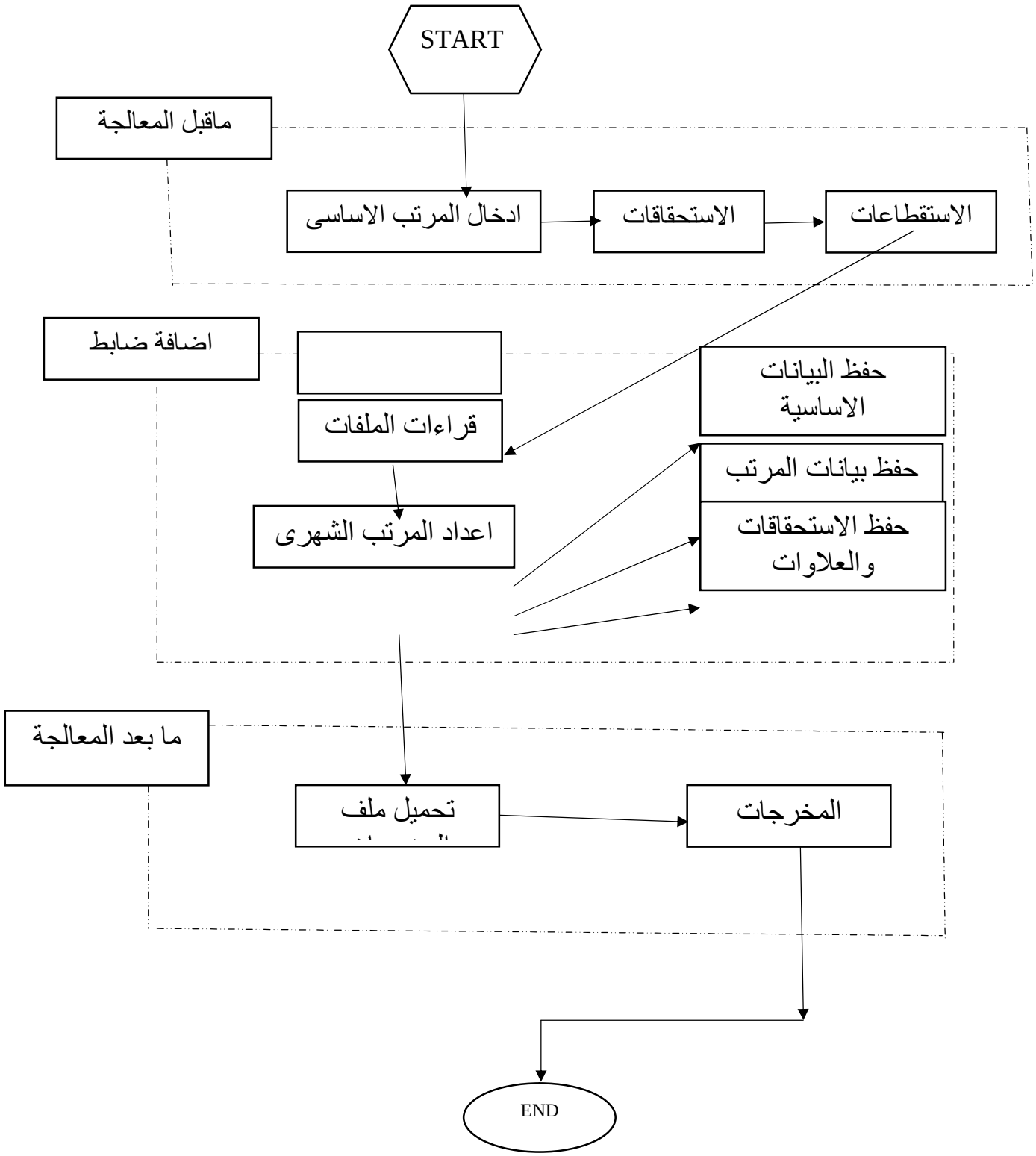
والشكل (5-2) يوضح الخدمات التى يقدمها نظام المرتبات وهى تقديم الخدمات المختلفه للوصول لاهداف النظام اعدا المرتبات والتقارير والاستفسارات هي وظيفة ما بعد المعالجة ووصف الوظائف الفرعية كما في الشكل (5-2) الشكل (5-3) يوفر نظرة شاملة من الاجراءات المتبعة من قبل المستخدم لتنفيذ وظائف النظام

ما قبل المعالجة :

هي عبارة عن معالجة لاعداد تنفيذ الخدمات المختلفة ويشمل الاعداد ادخال بيانات المرتبات الاساسية واسماء وقيم الاستحقاقات والاستقطاعات واسماء البنوك والافرع وكل البيانات الاساسية التى يمكن استخدمها لتنفيذ الخدمات المختلفه . ثم بعد ذلك تنفيذ تلك الخدمات وجلب كل الملفات المطلوبه التى تم اعدادها قبل اجراء العملية ثم تخزين البيانات ومن ثم استخراج التقارير اللازمة



الشكل (5-2) خدمات نظام المرتبات .



الشكل (5-3) نظرة شاملة لتنفيذ وظائف النظام

5-4 تقييم نظام المرتبات

يتم اجراء تقييم النظام لالتقاط المتطلبات الوظيفية والمتطلبات غير الوظيفية .

5-4-1 تقييم المتطلبات الوظيفية :

المتطلبات الوظيفية للنظام تشير الي القدرة على اداء الحسابات وكل الوظائق بدقة معقوله وبناء على اختبار وردود فعل المستخدمين نتائج الحساب والخدمات مقبولة الي حد ما ، مما يدل على تنفيذ المتطلبات الوظيفية للنظام كما انه يوحى بانه يمكن ان يتم ترحيل المتطلبات الوظيفية عن طريق اعادة الهندسة دون تغيير منطق الوظائف

5-4-2 تقييم المتطلبات الغير وظيفية :

يشير الاختبار على النظام من ردود فعل المستخدم وجود قضايا من المتطلبات غير الوظيفية وتغطي هذه الجوانب واجهة المستخدم والوثائق وسوف نتناقش بعض هذه القضايا في القسم التالي الذي يشمل غياب الوثائق وتنفيذ التكرار لمجموعة من التطبيقات .

جدول (5-1) القضايا في نظام مرتبات الضباط

التطبيق	المشكلة	تعريف المشكلة
قابلية الصيانة Maintainability	التكرار التكيف رفض الانماط	بعض الكود يكرر في اماكن اخري عدم العمل مع الواجهات الغرضية التوجه(ADF) الحساب والشاشة الرئيسية مع بعض صعوبة استخراج تقارير مختلفه من كائن واحد
سهولة الاستخدام Usability	تعقيد وظائف المنظمة عدم وجود تتبع للمعلومات دليل المستخدم عدم الرضا	لا تقدم العمليات بطريقة بديعية لا توجد معلومات للعملية الحاليه لا يوجد دليل للمستخدم المستخدم غير راضى عن المتطلبات
الموثوقية Reliability	معالجة الاخطاء غير مكتملة عدم وجود الثبات في البيانات	لم يتم وضع كلالاستثناءات فى الاعتبار لا توجد منتجات احتمالية مقامه من خبرات المستخدم
التوثيق Documentation	غياب الوثائق	لا توجد وثائق المتطلبات والتحليل والتصميم

5-4-2-1 التكرار :

التكرار يشير على ان واحدة او اجزاء من التعليمات البرمجية متكررة اماكن متعددة . يمكن ان نسب الكثير من قضايا الصيانة ، واحدة منه عدم الكفاءة في اصلاح الخلل . اذا كان هنالك خلل في الكشف عن التكرار هذا ربما جميع الوحدات خاطئة وتحتاج لتغيير في النظام الحالي وتستخدم بعض الثوابت مرات عديدة بنفس الاسم وتاخذ قيمة متاحة بدلا من تنسيق متغير ايضا يتم ترميز بعض المناطق المشتركة بطريقة متكرره وهذا ما يعوق عمليا اعادة الاستخدام ويزيد

الي حد كبير فرصة للخطا الطبيعي وتجاهل الاصلاح . المنهج الصحيح هو استخراج منطق مشترك باعتبارة وظيفة معزولة .

5-4-2-2 التكيف

تعنى تكيف النظم للعمل مع برامج اخرى او واجهات اخرى حديثة مع قواعد البيانات مما يعوق استخدام التكنلوجيا الحديثة واستخدام الواجهات المبنية على الغرضى التوجه لذلك هنا لابد من تغير البنية حتى تتكيف للعمل مع البرامج المبنية على غرضية التوجه.

5-4-2-3 مكافحة النمط

انماط التصميم الناتجة واثبتت الاطر الصحة من ممارسات التطوير وانشاء قوالب تصميم معينه من المتطلبات والسيناريوهات تطبق انماط التصميم تسهل عملية التصميم وضمان سهولة الاستخدام والمرونة، والاستقرار والفهم . مكافحة الانماط هي الضد للتصميم النمطي فى النظام الحالي تم الكشف عن العديد من معيبات التصميم في الترميز والتصميم – علي سبيل المثال لمحافحة التصميم هو الخلط بين النماذج والمشاهدات تشير الي النموذج لحساب المرتب والعلاوة المعاشيه واعداد المرتب والعمليات الحسابية ، و عملية ادخال البيانات واخراج البيانات ، وضع كود فى شاشة واحدة فى مكان واحد .

5-4-2-4 وظائف /وتنظيم واجهة المستخدم الرسومية :

هذه مسالة قابلية الاستخدام (usability) وهي عبارة عن كيف يصل المستخدم للوظائف المتوقعه من خلال الواجهة الرسومية عندما يتم استخدام نظام مع واجهة مستخدم جيدة المستخدم يستطيع التبديل بين الوظائف بطريقة سهلة باقل عدد من العمليات . في النظام الحالي يتم ترتيب واجهة المستخدم في شكل شجرة فانه اذا كان المسار للبرنامج الفرعى طويل يجعل المستخدم يحس بالملل ولا يوجد دليل للمستخدم ولا مساعدة ودعم وايضا يوجد عدم رضا عن المتطلبات التى يريده لذلك لابد من تطبيق المبادى التوجيهيه لسهولة الاستخدم الموجودة فى الجدول رقم . وايضا تصميم واجهات فعالة لنظم البرمجيات ويجب ان يتمركز تصميم الواجهة على المستخدم وان الواجهة متماسكة ومنطقية وقادرة على توفير المساعدة للمستخدم للتغلب على الأخطاء لذلك لابد من تطبيق مبادي تصميم واجهة المستخدم الموجودة فى الجدول وهى الالفة للمستخدم User familiarity و الانساق consistency و تقليل المفاجاة Minimal surprise

والاسترداد Recoverability و توجيه المستخدم User guidance و التنوع للمستخدمين User diversity .

5-4-2-4 معالجة الاستثناءات :

معالجة الاستناء هي الالية الاخري التى تشد النظام لتعقب الحالات الشاذه على وجه التحديد ولدية ثلاثة مهمات منع فشل النظام وتوفر اشارات تفسير عن حل معقول ومع ذلك فإن النظام الحالي لا يقدم هذه الالية ، فكثير النظام يفشل النظام دون اعطاء استثناءات للنظام .

5-4-2-5 لا يوجد نسخ لادخال المستخدم :

لا يتم ادخال بيانات المستخدم مثلا عند ادخال بيانات ضابط جديد يتم ادخال البيانات الاساسية فى جدول البيانات الاساسية ثم جدول المرتبات ثم الاستحقاقات والاستقطاعات مباشر فاذا ما اردنا بيانات الادخال مثلا لا يمكن الرجوع له اذ لابد من عمل نسخ بيانات المستخدم من خلال خلق كائنات جديدة والتحكم بها بحيث تكون موجودة يمكن الرجوع له فى اى وقت .

5-5 استخراج المتطلبات فى نظام مرتبات الضباط

تم تطبيق الاجراء الذي نوقش في 2.4 لاعادة التقاط المتطلبات فيما يلي بعض النقاط التى يتم الاهتمام به فى وثيقة المتطلبات الوظيفية :

- **توثيق المتطلبات الوظيفية** :

لان وظائف النظام القائم جيدة من حيث دقة الحسابات والتقارير فانه يعني الوفاء بالكثير من المتطلبات الوظيفية الحالية ومع ذلك ليس هنالك صلة بالوثائق والنظام العامل لذلك سوف يكون مهمة المتطلبات من النظام القائم اعادة البناء وخلق الوثائق .

- **التركيز على المتطلبات الغير وظيفية** :

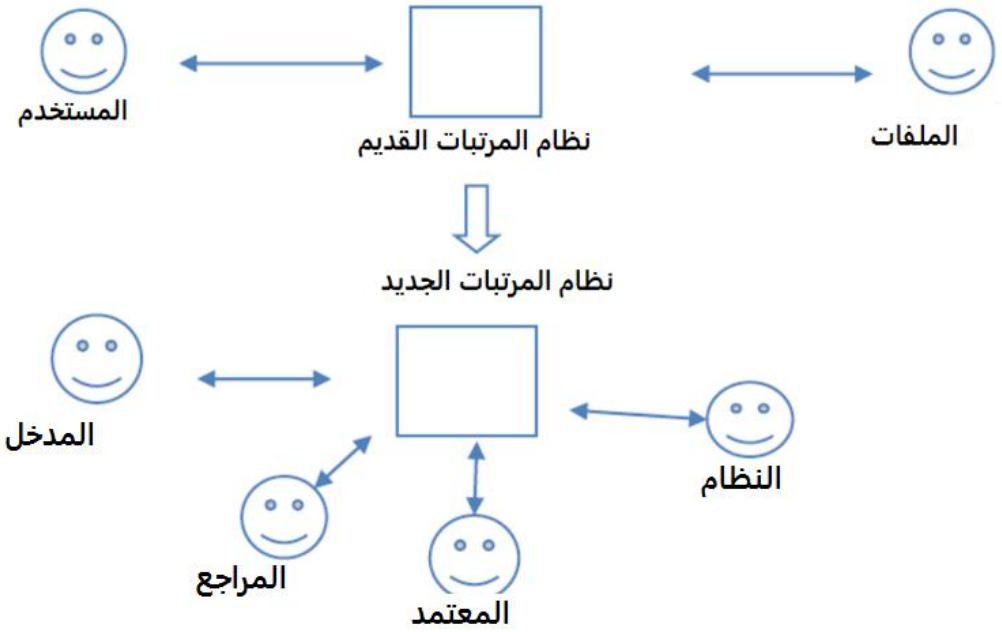
معظم الشكاوي ضد النظام القائم من فقر سهولة الاستخدام لذلك سوف يتم تحديد المتطلبات الغير وظيفة والتى تكون مهمة في هذه المرحلة .

- **تخطي تحديد السيناريو** :

الان لدينا النظام القائم وتشغيلة سوف يكون المصدر الرئيسي لالتقاط حالات الاستخدام بدلا من الذهاب الي تحديد سيناريوهات من الصفر .

5-5-1 تحديد الفاعلين :

الجهات الفاعلة هي الجهات تتفاعل مع النظام يتم تغيير تعريف الجهات الفاعلة بشكل مستمر بسبب ادخال الملفات الجديده لتحقيق افضل استمرار للبيانات الشكل (5-4) يمثل الجهات الفاعلة في نظام مرتبات الضباط



الشكل (5-4) الجهات الفاعلة فى نظام المرتبات .

5-5-2 السيناريو

يتم تحديد تطبيق السيناريو في الغالب فى حالة بناء نظام جديد حيث يبدا التطوير من نقطة الصفر . بناء السيناريوهات ضروري لفهم مجال المشكلة في هذا البحث يجري إعادة الهندسة على اساس النظام القائم والتي يمكن تحديد حالات الاستخدام مباشرة حتى هذه الخطوة يتم تجاهلة هنا في هذا المشروع .

5-5-3 حالات الاستخدام

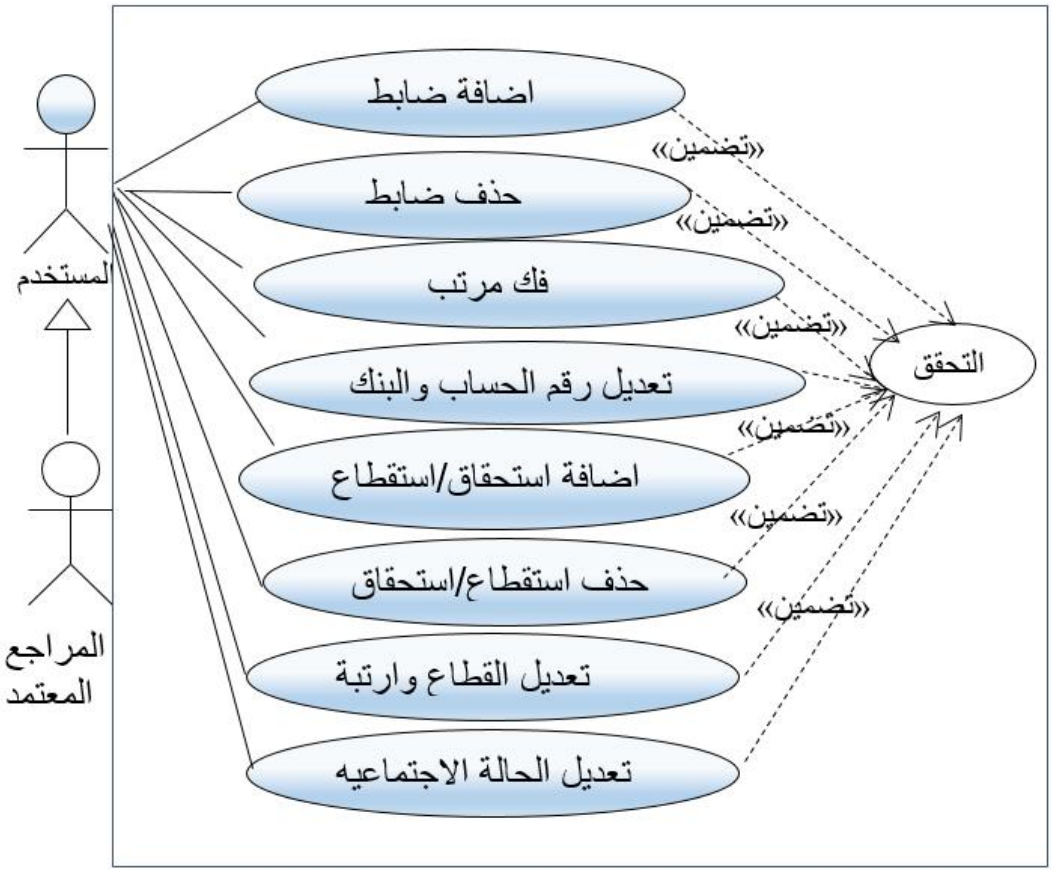
وتنقسم حالات الاستخدام الي ثلاثة اجزاء حالات استخدم الخدمات التى تودى بواسطة المستخدم وحالات استخدام ما قبل المعالجة او البيانات المساعدة وحالات استخدام ما بعد المعاجة وهى التقارير والبحث.

الجدول (5-2) :نظرة عامة عن حالات الاستخدام فى نظام مرتبات الضباط

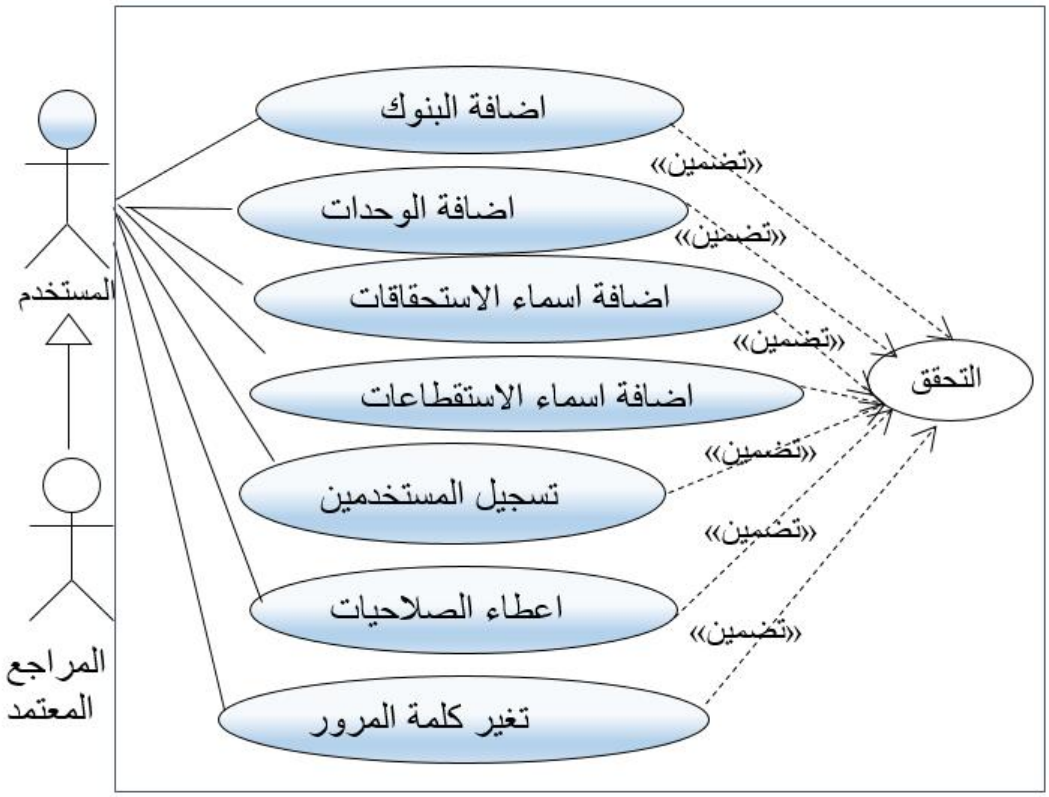
حالة الاستخدام	تعريف الطريقة
* اضافة البنك والفرع * اضافة الوحدات * اضافة اسماء الاستحقاقات *اضافة اسماء الاستقطاعات	حالة الاستخدام مرتبطة بالنظام الموجود وللقيام بالخدمات الرئسيه هى عبارة عن البيانات المساعدة وهى تتم قبل المعالجة لاداء الخدمات
* اضافة ضابط جديد * حذف ضابط * فك مرتب * تعديل رقم الحساب والبنك * اضافة استحقاق * حذف استحقاق * اضافة استقطاع * حذف استقطاع * تعديل القطاع والرتبة * تعديل الحالة الاجتماعيه	الحالة ترجع في التغيير في ملفات النظام وهى تعد الخدمات الرئيسية للنظام
* اضافة المستخدمين *اعطاء الصلاحيات *تغير كلمة المرور	حالة الاستخدام مرتبطة بالنظام القديم لادارة النظام

ويتم تحديد ها من خلال استكشاف النظام القديم من خلال قراءة الوثائق التى يمكن الوصول اليه والكود وكيفية تشغيل النظام الحالى وتشغيل الحالات القائمة ومناقشتة مع المستخدمين والمحاسبين والمراجعيين .

في الجدول (5-2) قائمة من حالات الاستخدام التى تم تحديدها ل.نظام مرتبات الضباط كل حالة مستخدمة هي بالتركيز على اليه عمل النظام وبناء واعطاء اوصاف ذات صلة .



الشكل (5-5) حالات استخدام الخدمات الرئسية للمرتبات .



الشكل (5-6) حالات استخدام الخدمات المساعدة .

الجدول (5-3) أضافة ضابط جديد(الادخال)

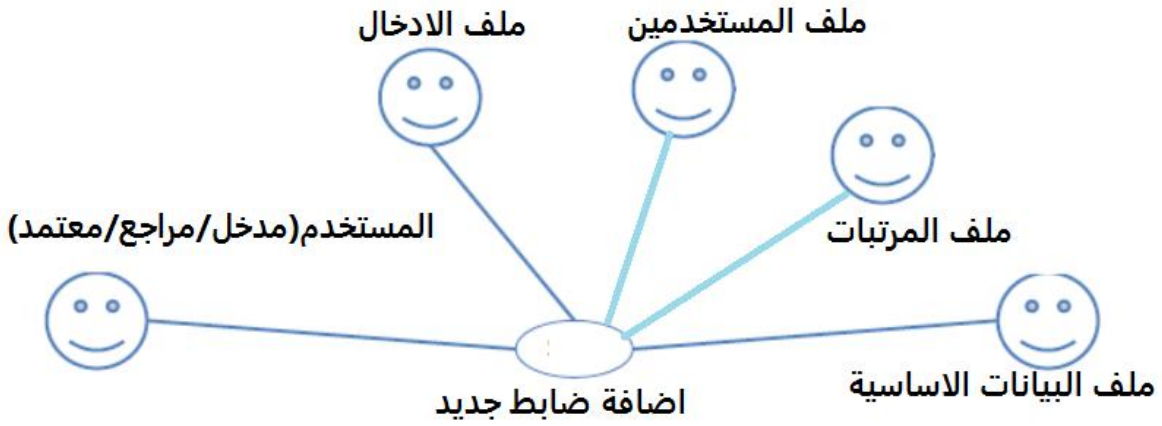
واقعة الإستخدام	أضافة ضابط جديد(الادخال)
وصف موجز	يقوم المستخدم بانزال بيانات الضابط الاساسية بعد اصدار قرار تعين الضابط من شئون الضباط وارساله الى ادارة الشئون الماليه
اللاعبون(الفاعلون)	المدخل والنظام
المتطلبات	صلاحية الدخول.
شروط مسبقة	ان يتم التعليق من رئس الشعبة بالانزال. قرار التعين من شئون الضباط/ملف الضابط الجديد/تحديد البنك والفرع
شروط لاحقة	يكتمل الانزال بنجاح ويعطى النظام رسالة بذلك.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار اضافة ضابط جديد. 4- ادخال البيانات. 5- حفظ البيانات.
تدفق بديل	إذا حدثت أي مشكلة يمكن الادخال مرة أخرى.

الجدول (5-4) أضافة ضابط جديد(المراجعه)

واقعة الإستخدام	أضافة ضابط جديد(المراجعه)
وصف موجز	يقوم المراجع بالبحث عن ضابط محدد ثم يطابق المعلومات المنزلة مع ملف الضابط ثم يقوم بتاكيد صحة البيانات واذ كانت غير صحيحه ترجع مرة اخرى الى المدخل لتصحيحه
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات .
شروط لاحقة	تكتمل المراجعة بنجاح ويعطى النظام رسالة بذلك.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيارمراجعة اضافة ضابط جديد. 4- البحث عن ضابط معين. 5- اذا كانت البيانات صحيحة يضغط على مراجعة.

الجدول (5-5) أضافة ضابط جديد(الاعتماد)

واقعة الإستخدام	أضافة ضابط جديد(الاعتماد)
وصف موجز	يقوم المعتمد بالبحث عن الضابط المراد اعتماده بعد التاكد من صحة البينات يتم اعتماده وبذلك يكون الضابط اعتمد ضمن كشوفات المرتبات والا يتم ارجاعه الى المراجعه مرة اخرى
اللاعبون(الفاعلون)	المعتمد والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكوتم مراجعة البيانات .
شروط لاحقة	يكتمل الاعتماد بنجاح ويعتمد ضمن كشوفات المرتبات.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختياراعتماد اضافة ضابط جديد. 4- البحث عن ضابط معين. 5- اذا كانت البيانات صحيحة يضغط على اعتماد.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.
تدفق إستثنائي	اذا كانت البيانات غير صحيحه يضغط على ارجاع



الشكل (5-7) حالة استخدام اضافة ضابط جديد

الجدول (5-6) حذف ضابط (الانزال)

واقعة الإستخدام	حذف ضابط (الانزال)
وصف موجز	يقوم المستخدم بادخال بيانات الضابط المراد حذفه من النظام
اللاعبون(الفاعلون)	المدخل والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون الضابط منزل اصلا وله مرتب/قرار الحذف مع بيان السبب .
شروط لاحقة	يكتمل الانزال بنجاح واعطى رسالة بذلك.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار انزال حذف ضابط. 4- البحث عن ضابط المعين. 5- اذا كانت البيانات صحيحة يضغط على حفظ.

الجدول (5-7) حذف ضابط (المراجعه)

واقعة الإستخدام	حذف ضابط(المراجعه)
وصف موجز	يقوم المراجع بالبحث عن الضابط المراد مراجعته بعد التاكد من صحة البينات يتم المراجعه والا يتم ارجاعه الى الانزال مرة اخرى
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات .
شروط لاحقة	تكتمل المراجعه بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة حذف ضابط 4- البحث عن ضابط معين. 5- اذا كانت البيانات صحيحة يضغط على مراجعه.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.
تدفق إستثنائي	اذا كانت البيانات غير صحيحه يضغط على ارجاع

الجدول (5-8) حذف ضابط (الاعتماد)

واقعة الإستخدام	حذف ضابط(الاعتماد)
وصف موجز	يقوم المعتمد بالبحث عن الضابط المراد اعتماده بعد التاكد من صحة البينات يتم الاعتماد وحذفه من كشوفات المرتبات والا يتم ارجاعه الى المراجعه مرة اخرى
اللاعبون(الفاعلون)	المعتمد والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم مراجعة البيانات .
شروط لاحقة	يكتمل الاعتماد بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار اعتماد حذف ضابط 4- البحث عن ضابط معين. 5- اذا كانت البيانات صحيحة يضغط على اعتماد.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.
تدفق إستثنائي	اذا كانت البيانات غير صحيحه يضغط على ارجاع

الجدول (5-9) فك مرتب(الانزال)

واقعة الإستخدام	فك مرتب(الانزال)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه لارجاعة فاذا وجده يكمل بيانات الارجاع حسب قرار الارجاع
اللاعبون(الفاعلون)	المنزل والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون لدى الضابط بيانات سبق انزاله من قبل ثم اوقف /قرار الارجاع.
شروط لاحقة	يكتمل الانزال بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار انزال ارجاع ضابط 4- البحث عن ضابط معين. 5- اذا وجده يكمل البيانات ويحفظها.

الجدول (5-10) فك مرتب(المراجعة)

واقعة الإستخدام	فك مرتب(المراجعة)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للمراجعه فاذا وجده يتاكد من صحة البيانات مع مسند الفك ثم يضغط على مراجعه
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات فى فك مرتب /قرار فك المرتب.
شروط لاحقة	تكتمل المراجعه بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة فك مرتب 4- البحث عن ضابط معين. 5- اذا وجده يراجع البيانات ويحفظها.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.
تدفق إستثنائي	اذا كانت البيانات غير صحيحة يضغط على ارجاع

الجدول (5-11) فك مرتب(الاعتماد)

واقعة الإستخدام	فك مرتب(الاعتماد)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للمراجعه فاذا وجده يتاكد من صحة البيانات مع مسند الفك ثم يضغط على مراجعه
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات فى فك مرتب /قرار فك المرتب.
شروط لاحقة	تكتمل المراجعه بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة فك مرتب 4- البحث عن ضابط معين. 5- اذا وجده يراجع البيانات ويحفظها.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.

الجدول (5-12) تعديل رقم الحساب والبنك (الانزال)

واقعة الإستخدام	تعديل رقم الحساب والبنك (الانزال)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم ثم يقوم بتعديل رقم الحاب والبنك والفرع
اللاعبون(الفاعلون)	المنزل والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون لدى الضابط بيانات سبق انزاله من قبل.
شروط لاحقة	يكتمل الانزال بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار انزال تعديل رقم الحساب والبنك 4- البحث عن ضابط معين. 5- يدخل بيانات تعديل البنك الجديده
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.

الجدول (5-13) تعديل رقم الحساب والبنك(المراجعة)

واقعة الإستخدام	تعديل رقم الحساب والبنك (المراجعة)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للمراجعه فاذا وجده يتاكد من صحة البيانات مع مستند تغير البنك الجديد
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات تعديل رقم الحاب والبنك /مستند البنك الجديد.
شروط لاحقة	تكتمل المراجعه بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة تعديل رقم الحساب والبنك 4- البحث عن ضابط معين. 5- اذا وجده يراجع البيانات ويحفظها.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.
تدفق إستثنائي	اذا كانت البيانات غير صحيحة يضغط على ارجاع

الجدول (5-14) تعديل رقم الحساب والبنك(الاعتماد)

واقعة الإستخدام	تعديل رقم الحساب والبنك (الاعتماد)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للاعتماد فاذا وجده يتاكد من صحة البيانات معالتعديل ا ثم يضغط على اعتماد فيتم تعديل رقم الحساب والبنك والفرع
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات فى تعديل رقم الحساب والبنك ومراجعته /مستند البنك الجديد
شروط لاحقة	يكتمل الاعتماد بنجاح ويتم تعديل رقم الحساب والبنك.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار تعديل رقم الحساب والبنك 4- البحث عن ضابط معين. 5- اذا وجده يعتمد البيانات ويحفظها.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.

الجدول (5-15) اضافة استحقاق (الانزال)

واقعة الإستخدام	اضافة استحقاق (الانزال)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم ثم يقوم بانزال بيانات تعديل الاستحقاق وقيمة الاستحقاق
اللاعبون(الفاعلون)	المنزل والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون لدى الضابط بيانات سبق انزاله من قبل.
شروط لاحقة	يكتمل الانزال بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار اضافة استحقاق 4- البحث عن ضابط معين. 5- يدخل بيانات تعديل الاستحقاق الجديد

الجدول (5-16) اضافة استحقاق (المراجعة) اضافة استحقاق

واقعة الإستخدام	اضافة استحقاق (المراجعة)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للمراجعه فاذا وجده يتاكد من صحة البيانات مع مستند تغير اضافة الاستحقاق
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات اضافة استحقاق جديد /مستند الاستحقاق الجديد.
شروط لاحقة	تكتمل المراجعه بنجاح ويتم اضافة الاستحقاق فى المرتب.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة اضافة استحقاق 4- البحث عن ضابط معين. 5- اذا وجده يراجع البيانات ويحفظها.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.
تدفق إستثنائي	اذا كانت البيانات غير صحيحة يضغط على ارجاع

الجدول (5-17) حذف استحقاق (الانزال)

واقعة الإستخدام	حذف استحقاق (الانزال)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم الضابط ثم يقوم بانزال بينات حذف استحاق .
اللاعبون(الفاعلون)	المنزل والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون لدى الضابط بيانات سبق انزاله من قبل ولديه الاستحقاق المراد حذفه.
شروط لاحقة	يكتمل الانزال بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار حذف استحقاق 4- البحث عن ضابط معين. 5- يدخل بيانات حذف الضابط

الجدول (5-18) حذف استحقاق (المراجعة)

واقعة الإستخدام	حذف استحقاق (المراجعة)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للمراجعه فاذا وجده يتاكد من صحة البيانات مع مستند تغير حذف الاستحقاق ثم يتم حذف الاستحقاق وتخصم قيمته من المرتب
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات حذف استحقاق /مستند حذف الاستحقاق.
شروط لاحقة	تكتمل المراجعه بنجاح ويتم حذف الاستحقاق وخصم قيمته من المرتب.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة حذف استحقاق 4- البحث عن ضابط معين. 5- اذا وجده يراجع البيانات ويحفظها.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.
تدفق إستثنائي	اذا كانت البيانات غير صحيحة يضغط على ارجاع

الجدول (5-19) اضافة استقطاع (الانزال)

واقعة الإستخدام	اضافة استقطاع (الانزال)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقة ثم يقوم بانزال بيانات اضافة استقطاع وقيمة الاستقطاع
اللاعبون(الفاعلون)	المنزل والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون لدى الضابط بيانات سبق انزاله من قبل.
شروط لاحقة	يكتمل الانزال بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار اضافة استقطاع 4- البحث عن ضابط معين. 5- يدخل بيانات الاستقطاع

الجدول (5-20) اضافة استقطاع (المراجعة) اضافة استقطاع

واقعة الإستخدام	اضافة استقطاع (المراجعة)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للمراجعه فاذا وجده يتاكد من صحة البيانات مع مستند تغير اضافة الاستقطاع ثم يراجع ويخصم من المرتب
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات اضافة استقطاع جديد /مستند الاستقطاع الجديد.
شروط لاحقة	تكتمل المراجعه بنجاح ويتم اضافة الاستطاع وتخصم من المرتب.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة اضافة استقطاع 4- البحث عن ضابط معين. 5- اذا وجده يراجع البيانات ويحفظها.

الجدول (5-21) حذف استقطاع (الانزال)

واقعة الإستخدام	حذف استقطاع (الانزال)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم الضابط ثم يقوم بانزال بيانات حذف استقطاع .
اللاعبون(الفاعلون)	المنزل والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون لدى الضابط بيانات سبق انزاله من قبل ولديه الاستقطاع المراد حذفه.
شروط لاحقة	يكتمل الانزال بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار حذف استقطاع 4- البحث عن ضابط معين. 5- يدخل بيانات حذف الاستقطاع

الجدول (5-22) حذف استقطاع (المراجعة)

واقعة الإستخدام	حذف استقطاع(المراجعة)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للمراجعه فاذا وجده يتاكد من صحة البيانات مع مستند حذف الاستقطاع ثم يتم حذف الاستقطاع وتخصم قيمته من المرتب
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات حذف استقطاع /مستند حذف الاستقطاع.
شروط لاحقة	تكتمل المراجعه بنجاح ويتم حذف الاستقطاع وخصم قيمته من المرتب.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة حذف استقطاع 4- البحث عن ضابط معين. 5- اذا وجده يراجع البيانات ويحفظها.

الجدول (5-23) تعديل القطاع والرتبة (الانزال)

واقعة الإستخدام	تعديل القطاع والرتبة (الانزال)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم الضابط ثم يقوم بانزال بيانات تعديل القطاع والرتبة .
اللاعبون(الفاعلون)	المنزل والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون لدى الضابط بيانات سبق انزاله من قبل .
شروط لاحقة	يكتمل الانزال بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار تعديل القطاع والرتبة 4- البحث عن ضابط معين. 5- يدخل بيانات القطاع الجديد والرتبة
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.

الجدول (5-24) تعديل القطاع والرتبة (المراجعة)

واقعة الإستخدام	تعديل القطاع والرتبة (المراجعة)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للمراجعه فاذا وجده يتاكد من صحة البيانات مع مستند تعديل القطاع والرتبة ويتم تغير قيمه المرتب .
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات تعديل القطاع والرتبه /مستند تعديل القطاع والرتبه.
شروط لاحقة	تكتمل المراجعه بنجاح ويتم حذف الاستقطاع وخصم قيمته من المرتب.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة تعديل القطاع والرتبه4- البحث عن ضابط معين. 5- اذا وجده يراجع البيانات ويحفظها.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.
تدفق إستثنائي	اذا كانت البيانات غير صحيحة يضغط على ارجاع

الجدول (5-25) تعديل الحالة الاجتماعيه (الانزال)

واقعة الإستخدام	تعديل الحالة الاجتماعيه (الانزال)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم الضابط ثم يقوم بانزال بيانات تعديل الحالة الاجتماعيه .
اللاعبون(الفاعلون)	المنزل والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون لدى الضابط بيانات سبق انزاله من قبل .
شروط لاحقة	يكتمل الانزال بنجاح.
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار تعديل الحالة الاجتماعيه 4- البحث عن ضابط معين. 5- يدخل بيانات تعديل القطاع والرتبه

الجدول (5-26) تعديل الحالة الاجتماعيه (المراجعة) تعديل الحالة الاجتماعيه

واقعة الإستخدام	تعديل الحالة الاجتماعيه (المراجعة)
وصف موجز	يقوم الممستخدم بالبحث عن بيانات الضابط عن طريق رقم البطاقه للمراجعه فاذا وجده يتاكد من صحة البيانات مع مستند تعديل الحالة الاجتماعيه ويتم تغير قيمه المرتب .
اللاعبون(الفاعلون)	المراجع والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون تم انزال البيانات تعديل الحالة الاجتماعيه /مستند تعديل القطاع الحالة الاجتماعيه.
شروط لاحقة	تكتمل المراجعه بنجاح ويتم تعديل الحالة الاجتماعيه
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار مراجعة تعديل الحالة الاجتماعيه- البحث عن ضابط معين. 5- اذا وجده يراجع البيانات ويحفظها.
تدفق بديل	إذا حدثت أي مشكلة يمكن البحث مرة أخرى.
تدفق إستثنائي	اذا كانت البيانات غير صحيحة يضغط على ارجاع

الجدول (5-27) اضافة البنك والفرع

واقعة الإستخدام	اضافة البنك والفرع
وصف موجز	يقوم المستخدم بادخال البنك والفرع وترقيمه بعد التاكد من انه غير موجود.
اللاعبون(الفاعلون)	المستخدم والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان يكون غير مدخل البنك والفرع.
شروط لاحقة	يكتمل الادخال بنجاح ويضاف البنك والفرع
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار البنوك5- يتم ادخال بيانات البنك والفرع والحفظ .
تدفق بديل	*********

الجدول (5-28) اضافة وحدة جديده

واقعة الإستخدام	اضافة وحدة جديده
وصف موجز	يقوم المستخدم بادخال بيانات الوحدة الجديدة.
اللاعبون(الفاعلون)	المستخدم والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان تكون غير مدخله من قبل.
شروط لاحقة	يكتمل الادخال بنجاح وتضاف الوحدة الى النظام
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار الوحدات 5- يتم ادخال بيانات الوحدة والحفظ .
تدفق بديل	*********

الجدول (5-29) اضافة اسم استحقاق جديد

واقعة الإستخدام	اضافة اسم استحقاق جديد
وصف موجز	يقوم المستخدم بادخال بيانات استحقاق جديد.
اللاعبون(الفاعلون)	المستخدم والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان تكون غير مدخله من قبل.
شروط لاحقة	يكتمل الادخال بنجاح ويضاف اسم الاستحقاق الى النظام
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار اضافة اسم الاستحقاق 5- يتم ادخال اسم الاستحقاق والحفظ .
تدفق بديل	*********

الجدول (5-30) اضافة اسم استقطاع جديد

واقعة الإستخدام	اضافة اسم استقطاع جديد
وصف موجز	يقوم المستخدم بادخال بيانات اسم استقطاع جديد.
اللاعبون(الفاعلون)	المستخدم والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان تكون غير مدخله من قبل.
شروط لاحقة	يكتمل الادخال بنجاح ويضاف اسم الاستقطاع الى النظام
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار اضافة اسم استقطاع 5- يتم ادخال اسم الاستقطاع والحفظ .
تدفق بديل	*********

الجدول (5-31) اضافة اسم مستخدم جديد

واقعة الإستخدام	اضافة اسم مستخدم جديد
وصف موجز	يقوم المستخدم بادخال بيانات المستخدم الجديد.
اللاعبون(الفاعلون)	المستخدم والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	ان تكون غير مدخله من قبل.
شروط لاحقة	يكتمل الادخال بنجاح ويضاف اسم المستخدم الى النظام
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار اضافة مستخدم جديد 5- يتم ادخال بيانات المستخدم والحفظ .

الجدول (5-32) اعطاء الصلاحيات

واقعة الإستخدام	اعطاء وحذف وتعيل الصلاحيات
وصف موجز	يقوم المستخدم بالبحث عن المستخدم عن طريق اسم المستخدم ذا وجد يضيف او يعدل او يحذف الصلاحيات .
اللاعبون(الفاعلون)	المستخدم والنظام
المتطلبات	صلاحية الدخول .
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار الصلاحيات 5- يتم ادخال بيانات الصلاحيات والحفظ.

الجدول (5-33) تغير كلمة المرور

واقعة الإستخدام	تغير كلمة المرور
وصف موجز	يقوم المستخدم بالبحث عن المستخدم عن طريق اسم المستخدم ذا وجد يعدل او كلمة المرور الصلاحيات .
اللاعبون(الفاعلون)	المستخدم والنظام
المتطلبات	صلاحية الدخول .
شروط مسبقة	وجود المستخدم اصلا فى النظام
شروط لاحقة	يكتمل الادخال بنجاح وتعدل كلمة المرور .
التدفق الرئيسي	1-الدخول على برنامج مرتبات الضباط. 2- ادخال اسم المستخدم وكلمة المرور 3- اختيار تعديل كلمة المرور 5- يتم ادخال بيانات كلمة المرور الجديده والحفظ .
تدفق بديل	*********

5-5-4 المتطلبات الغير وظيفية : Non-functional requirements

تحليل المتطلبات الغير وظيفية للنظام القائم بالفعل فى القسم 5-4-2 يتم التقاط القضايا المتعلقة به ايضاً يمكن لهذه القضايا ان يكون حلها من خلال توفير الاحتياجات الغير الوظيفية التالية كما هو موضح في الجدول (5-34).

الجدول (5-34) المتطلبات الغير وظيفيه

المشكلة	المتطلبات غير الوظيفية
التكرار	تقليل التكرار
التكيف مع واجهات ADF	التصميم الغرضى التوجه
مضاد الانماط	يجب فصل الحسابات وواجهة المستخدم الرسومية
تعقيد واجهة المستخدم للمنظمة	تصميم واجهة المستخدم الرسومية والواجهات الرسومية التفاعلية
عدم تتبع المعلومات	تتبع المعلومات حتى تكون كاملة
مساعدة الاخطاء غير مكتملة	منع النظام من الخروج بشكل غير طبيعي
عدم وجود وثائق	تحليل وتصميم وتطوير الوثائق

التكرار هو واحد من العقبات الرئيسية لاعمال الصيانة التى تجعل من نظام مرتبات الضباط ان يكون مربكا للمطوريين لذلك ينبغي القضاء عليها لتحقيق جيد للمرونة والقراءة وامتداد واجهة المستخدم الرسومية وينبغي ان تكون منظمة بديهيا . تساعد عمليات التتبع في وضع تعقب المستخدم من العمليات الجارية ، ، معالجة الاستثناءات باعلام المستخدم مع معلومات الخطا ويقترح الحلول الممكنة ، من خلال تنفيذ تلك المتطلبات غير الوظيفية ينبغى يكون النظام الجديد اكثر سهولة في الاستخدام وسهولة فى التطوير وتطبيق المبادى التوجهية لسهولة الاستخدام الموجودة فى الجدول و مبادي تصميم واجهة الاستخدام المستخدم الموجودة فى الجدول .

5-6 التحليل الغرضي التوجه OO Analysis

قائمة على المتطلبات السابقة سيتم اجراء التحليل الغرضي التوجه في هذا القسم للتعرف على الكائنات الاساسية والخصائص والسلوكيات .

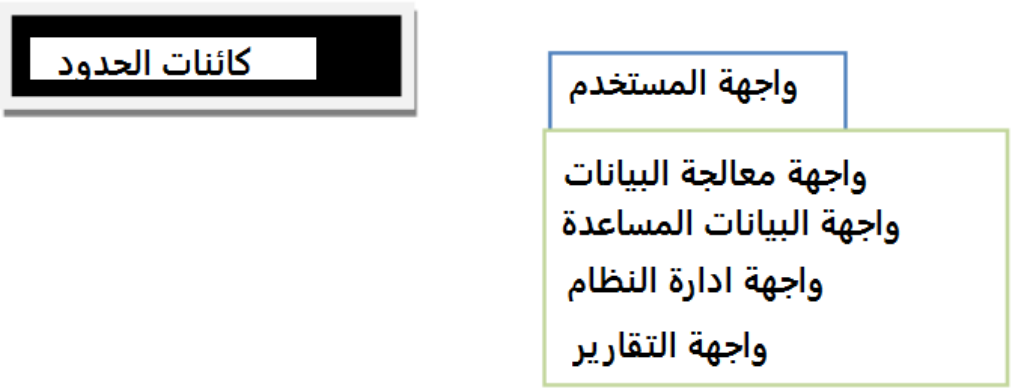
5-6-1 تعريف كائنات الحدود ، كائنات التحكم كائنات العناصر كائنات الاجسام

كائنات الاجسام هي كائنات للحفاظ على البيانات الثابتة في نظام مرتبات الضباط وهى تنقسم الى ثلاثة انواع نموذج ما قبل المعالجة وهى ملف المستخدمين وملف الصلاحيات و ملف اسماء الاستحقاقات وملف اسماء الاستقطاعات وملف فئات المرتبات الاساسية وملف اسماء الرتب وملفات نموذج تنفيذ الخدمات وهى ملف الادخال وملف البيانات الاساسيية وملف المرتبات و ملف الاستقطاعات وملف الاستحقاقات و ملف بيانات ونموذج مابعد المعالجة وهى اعداد المرتبات واستخراج التقارير المختلفه للمرتبات .

الشكل (5-8) كائنات العناصر

كائن الحدود (Boundary objects)

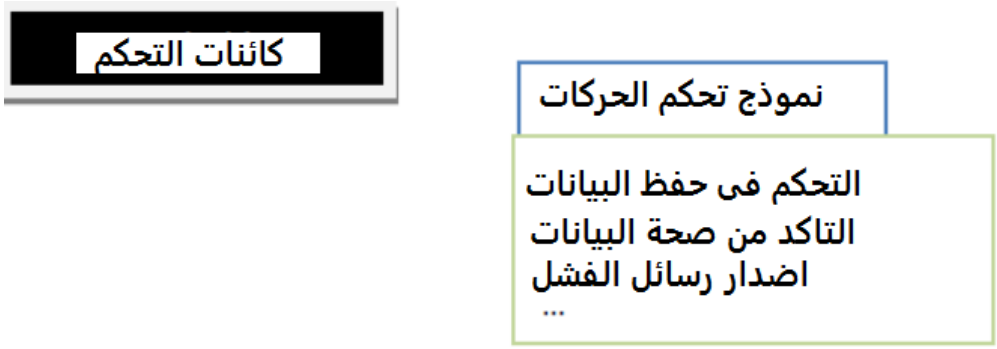
التعامل مع التفاعل بين النظام وعناصر الفاعلة في نظام مرتبات الضباط هنالك نوعان من الجهات الفاعلة المستخدم والملفات المختلفة كائنات الحدود هي التى تتعامل مع تفاعل المستخدم في واجهات المستخدم للتفاعلات بين الملفات والنظاموسوف يتم التفاعل معها من قبل النماذج ذالت الصلة بدلاً من تعريف اجسام الحدود .



الشكل (5-9) حدود الكائنات : واجهة المستخدم

كائنات التحكم: (control object)

هي المسئولة عن تنيسيق الكائنات ورسم خرائط الحدث للوظائف الصحيحة في نظام مرتبات الضباط . وظيفة التحكم مثل الكائن تتحكم على سلسلة من الاجراءات المتتابعة مثل التاكد من كلمة المرور قبل فتح النافذه ، وعملية حفظ النتائج .



الشكل (5-10) كائنات التحكم

5-6-2 نموذج التعاون modeling collaboration

نموذج التعاون يستكشف التفاعالات بين حالات الاستخدام والتحديد سلوكيات النظام . علي وجه التحديد التحليل لتحديد اوجه التعاون للمشاركة في الاحداث وتعريف الكائنات اضافة ضابط جديد الموضحة في الجدول (5-2)

يتم التحليل في هذا القسم ويتم عرض النتيجة فى الجدول (5-27) ويتكون العمود الاول من تدفق الاحداث من حالة استخدام اضافة ضابط جديد يحتوي العمود الثانى على التعاون و العمود الثالث هو قائمة الكائنات التى تعمل في كل حدث .

الشكل (5-9) هو مقابلة مخطط التعاون والذي هو رسم تفسير التعاون الذي يحدث بين الكائنات في هذا المخطط وحالات الاستخدام يتم استبدالها مع الكائنات ويتم تقديم التعاون لاتصال الكائنات.

الجدول (5-35)التعاون والكائنات في حالات استخدام اضافة ضابط جديد(الانزال)

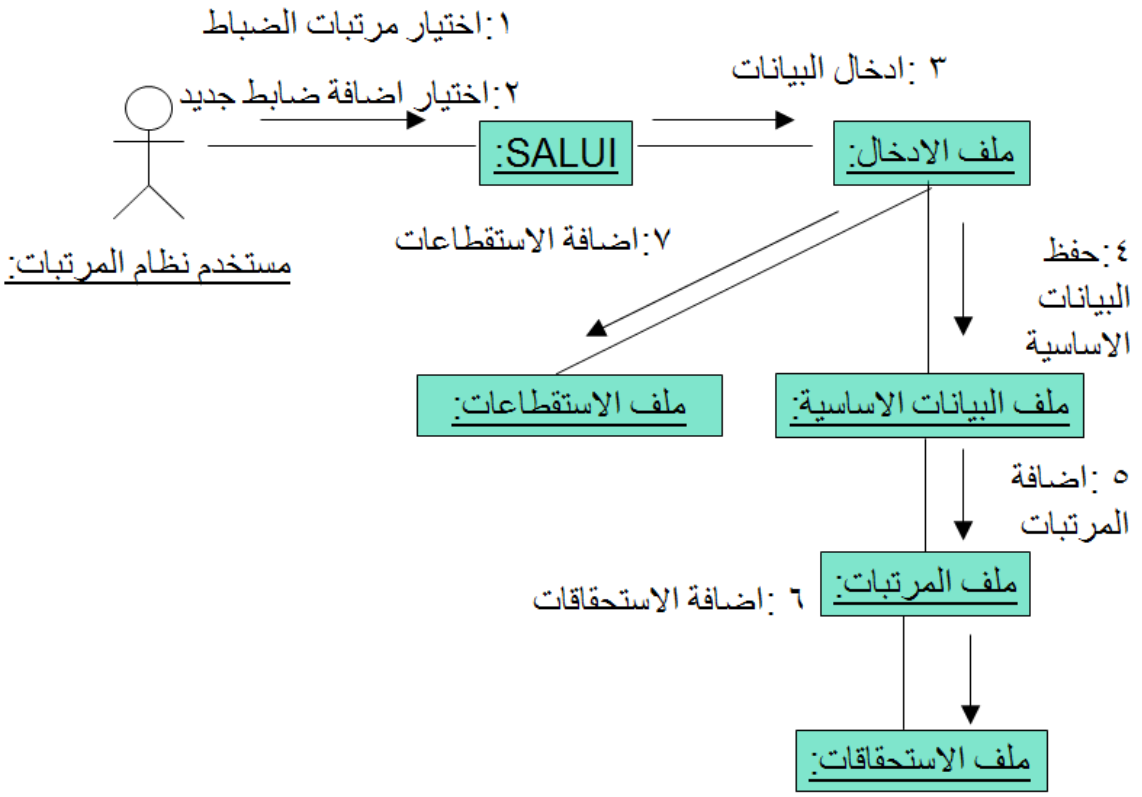
تدفق الاحداث في حالة الاستخدام	التعاون	الكائنات
1.المستخدم يقوم بفتح نظام المرتبات	1.النظام يطلب اسم المستخدم وكلمة المرور من المستخدم.	واجهة المستخدم نموذج ما قبل المعالجة(ملف المستخدم)
2.المستخدم يقوم بادخال اسم المستخدم وكلمة المرور	1.النظام ياخذ اسم المستخدم ويرجع رقم المستخدم . 2.النظام يقوم بارجاع صلاحية المستخدم وتمريره لواجهة المستخدم	واجهة المستخدم نموذج ما قبل المعالجة(ملف المستخدم +ملف الصلاحيات)
2.المستخدم يختار اضافة ضابط جديد	1.واجهة المستخدم تقوم بفتح واجهة اضافة ضابط جديد وتمريره الى نموذج الخدمات بعد التاكد من الصلاحيات 2.نموذج الخدمات يقوم بفتح ملف الادخال	واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط)
3.المستخدم يقوم بادخال البيانات للضابط الجديد وحفظها	واجهة المستخدم تقوم بارسال البيانات الى نموذج الخدمات(ملف اضافة ضابط)	واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط

الجدول (5-36)نموذج التعاون والكائنات في حالات استخدام اضافة ضابط جديد(المراجعه)

تدفق الاحداث في حالة الاستخدام	التعاون	الكائنات
1.المستخدم يقوم بفتح نظام المرتبات	1.النظام يطلب اسم المستخدم وكلمة المرور من المستخدم.	واجهة المستخدم نموذج ما قبل المعالجة(ملف المستخدم)
2.المستخدم يقوم بادخال اسم المستخدم وكلمة المرور	1.النظام ياخذ اسم المستخدم ويرجع رقم المستخدم . 2.النظام يقوم بارجاع صلاحية المستخدم وتمريره لواجهة المستخدم	واجهة المستخدم نموذج ما قبل المعالجة(ملف المستخدم +ملف الصلاحيات)
2.المستخدم يختار مراجعة اضافة ضابط جديد	1.واجهة المستخدم تقوم بفتح واجهة مراجعة اضافة ضابط جديد وتمريره الى نموذج الخدمات بعد التاكد من الصلاحيات 2.نموذج الخدمات يقوم بفتح ملف الادخال اضافة ضابط جديد	واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط)
3.المستخدم يقوم بادخال رقم البطاقه للبحث عن ضابط معين	1.النظام تقوم بارسال البيانات الى نموذج الخدمات(ملف اضافة ضابط) 2.النظام يقوم باسترجاع بيانات الضابط من نموذج الخدمات(ملف اضاقة ضابط)	واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط) واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط)
4.المستخدم يراجع البيانات ثم يضغط على المراجعه	.النظام تقوم بارسال البيانات الى نموذج الخدمات(ملف اضافة ضابط) وتعدل الحالة الى مراجعه	واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط)

الجدول (5-37) نموذج التعاون والكائنات في حالات استخدام اضافة ضابط جديد (الاعتماد)

تدفق الاحداث في حالة الاستخدام	التعاون	الكائنات
1.المستخدم يقوم بفتح نظام المرتبات	1.النظام يطلب اسم المستخدم وكلمة المرور من المستخدم.	واجهة المستخدم نموذج ما قبل المعالجة(ملف المستخدم)
2.المستخدم يقوم بادخال اسم المستخدم وكلمة المرور	1.النظام ياخذ اسم المستخدم ويرجع رقم المستخدم . 2.النظام يقوم بارجاع صلاحية المستخدم وتمريره لواجهة المستخدم	واجهة المستخدم نموذج ما قبل المعالجة(ملف المستخدم +ملف الصلاحيات)
2.المستخدم يختار اعتماد اضافة ضابط جديد	1.واجهة المستخدم تقوم بفتح واجهة لاعتماد اضافة ضابط جديد وتمريره الى نموذج الخدمات بعد التاكد من الصلاحيات 2.نموذج الخدمات يقوم بفتح ملف الادخال اضافة ضابط جديد	واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط)
3.المستخدم يقوم بادخال رقم البطاقه للبحث عن ضابط معين	1.النظام تقوم بارسال البيانات الى نموذج الخدمات(ملف اضافة ضابط) 2.النظام يقوم باسترجاع بيانات الضابط من نموذج الخدمات(ملف اضاقة ضابط)	واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط) واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط)
4.المستخدم يراجع البيانات ثم يضغط على اعتماد	1.النظام تقوم بارسال البيانات الى نموذج الخدمات(ملف اضافة ضابط) وتعدل الحالة الى اعتماد 2.يقوم النظام بتمرير البينات الى نموذج الخمات. 3.يتم اضافة البينات الاساسية 4.يتم اضافة بينات الاستحقاقات والاستقطاعات 5.يتم اضافة بيانات المرتب	واجهة المستخدم نموذج الخدمات(ملف اضاقة ضابط) نموذج الخدمات(ملف البيانات الاساسية+ملف المرتبات الشهريه+ملف الاستحقاقات الشهريه+ملف الاستقطاعات الشهريه)

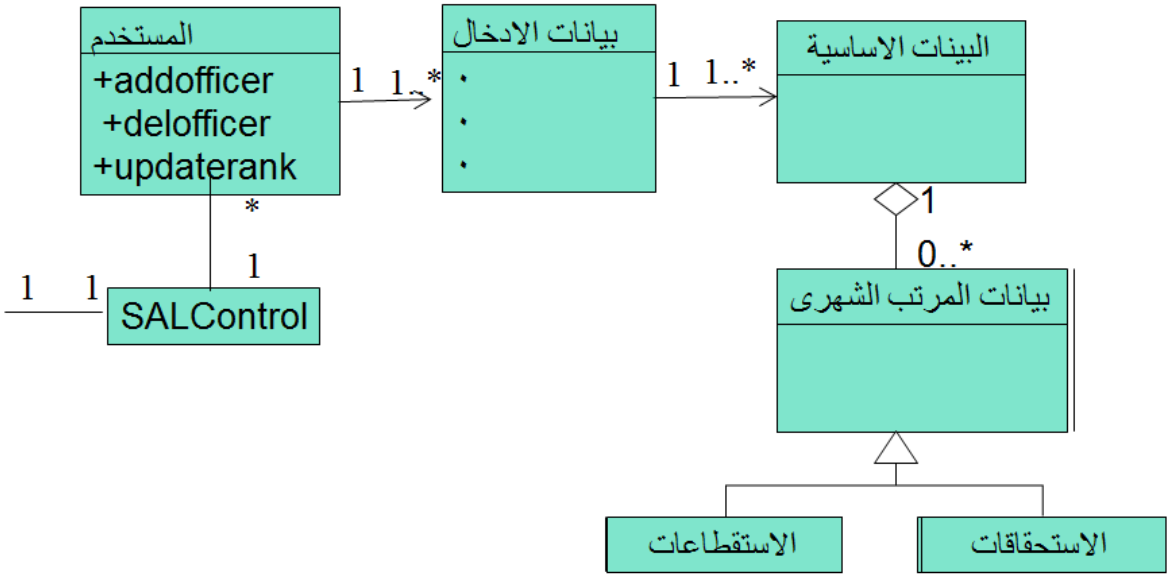


الشكل (5-11) مخطط التعاون لاضافة ضابط

5-6-3 التعميم والتخصيص :

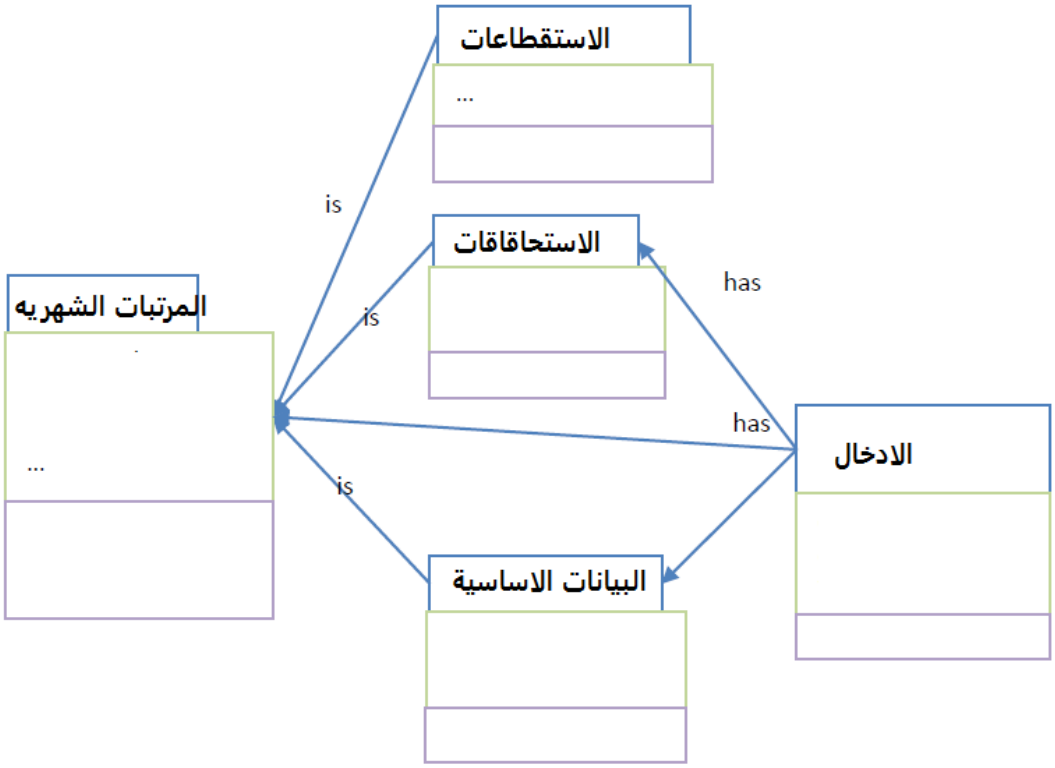
وتركز هذه العملية على تصميم السمات المشتركة (خصائص وسلوكيات) وتخصيصها السمات وكيف الاساليب التى تم حفظها في5-5-3 في الاجسام التالية خطوة بخطوة .

5-6-3-1تدفق اجسام الكائنات :

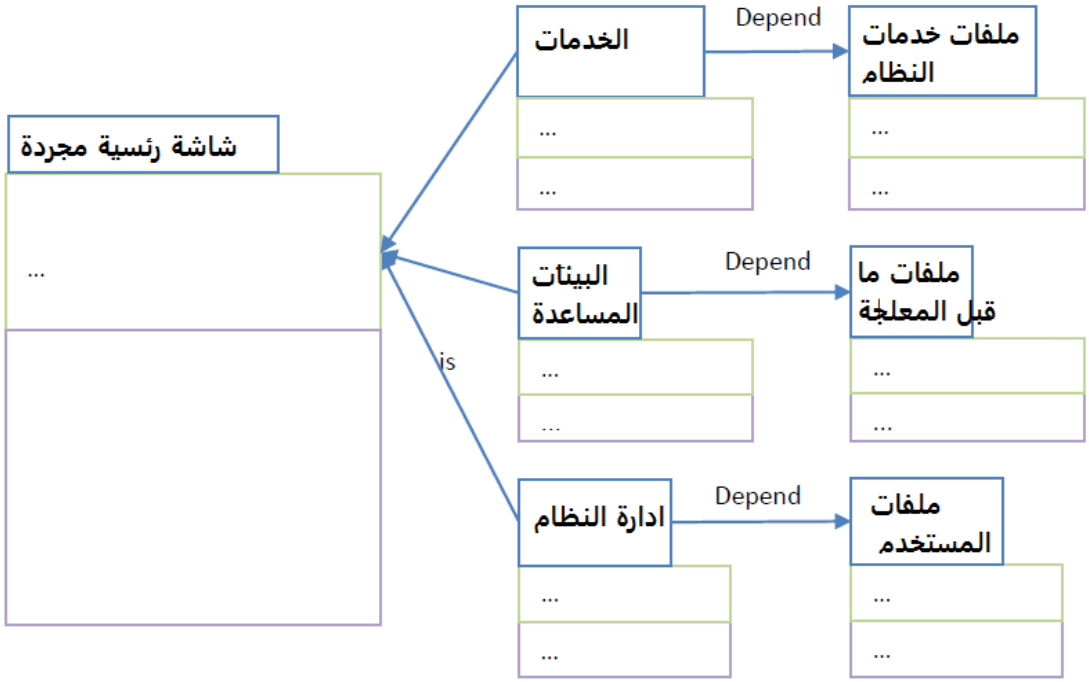


الشكل (5-12) مخطط الصنفيات

فى نظام مرتبات الضباط نجد الكائنات الثابته هى صنف المستخدمين والصلاحيات ونجد ان الصلاحيات ترث من المستخدم وبذلك يكون المستخدم اكثر عمومية وايضا نجد البيانات الاساسية هى وحيدة لاتكرر ولكن هذه البيانات الاساسية يرث من مرتبات الضباط الشهرية والمرتبات ترث من كل من الاستقطاعات والاستحقاقات وبالتالى تكون المرتبات الشهريه اكثر عمومية وبالتالى ذلك يقلل من عملية التكرار فى البيانات وفى كل العمليات تتم عن طريق الادخال ول يظهر فى الشكل السابق لم تظهر ولكن يظهر فى الشكل التالى



الشكل (5-13) كائنات العناصر



الشكل (5-14) كائنات التحكم

5-7 التصميم الغرضي التوجه OO Design

في هذه المرحلة قد يتحلل نظام مرتبات الضباط الي انظمة فرعية وسوف يتم تعيين الكائنات التى يتم تحديدها والتطور الفرعية للتحقيق فيما اذا كانت جميع الانظمة الفرعية بها الاجسام المقابلة فيما كانت جميع الانظمة الفرعية بها الاجسام المقابلة محددة وسوف يتم تحديد كائنات إضافية للنظم الفرعية التى يتم تنفيذها من قبل الكائنات الموجودة وذلك من خلال مخطط الانشطة ومخطط التحزيم .

5-7-1 استخراج اهداف التصميم :

وكما ذكرنا سابقاً اهداف التصميم لها تاثير كبير على بنية النظام ويمكن استخراجها من المتطلبات الغير وظيفية بعد مراجعة المتطلبات الغير وظيفية الموجودة فى الجدول يمكن تقسيم المتطلبات الغير وظيفية كالاتى كاهداف للتصميم .

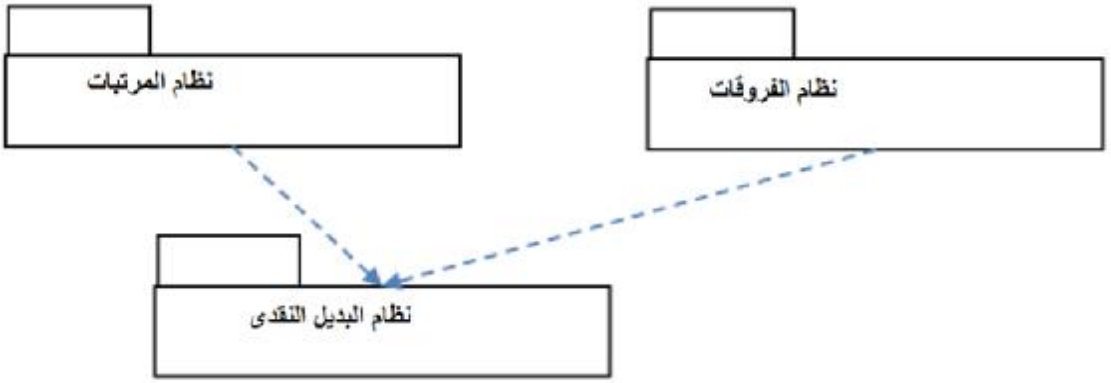
التفاصيل من لكل اهدف التصميم سيتم مناقشتها لاحقا في تصميم الكائن (5-38) وسيتم اقتراح الاستراتيجيات المتعلقة بها ومناقشتها.

الجدول (5-38) اهداف التصميم من المتطلبات الغير وظيفيه

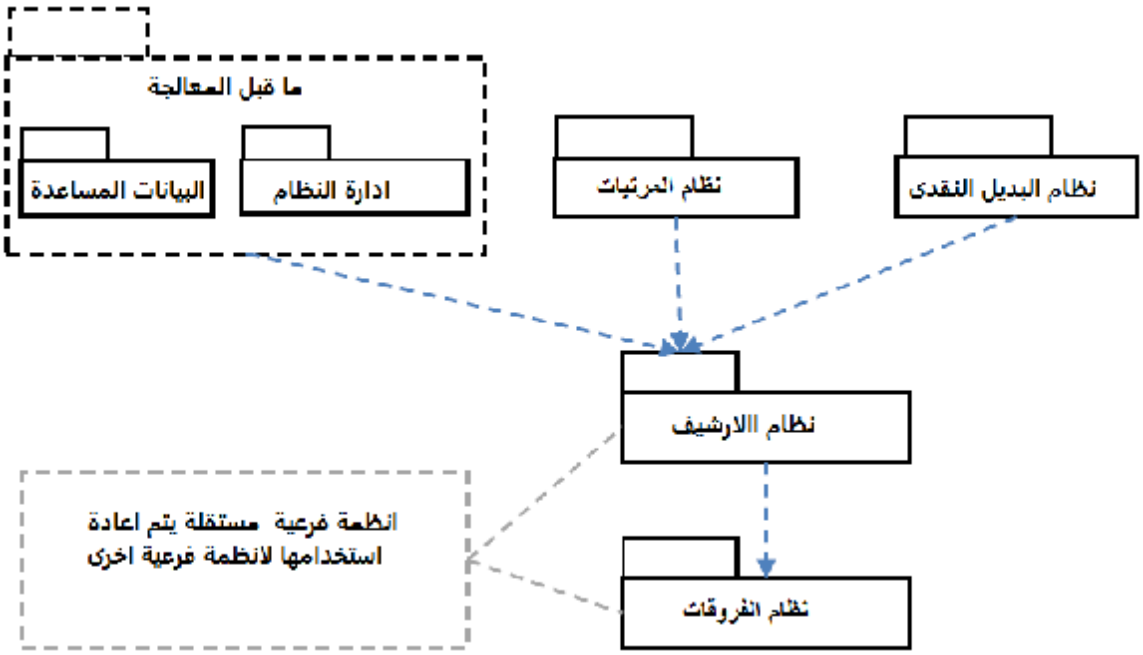
المتطلبات غير الوظيفية لاهداف التصميم	التاثير
التكيف مع الانظمة وتقليل التكرار	هيكلية الكائنات
الحسابات وواجهة المستخدم يجب ان تكون منفصلة لتقليل الاقتران	ابعاد الحسابات من واجهة المستخدم
التصميم البديعي لواجهة المستخدم الرسومية	اسلوب واجهة المستخدم الرسومية والبنية الاساسية
المساعدة فى معالجة الاخطاء	استخدام جيد للنظام
توثيق النظام	يسهل عملية الصيانة
توزيع النماذج منفصلة	تصميم الانظمة الفرعية

5-7-2 اعادة تصميم الانظمة الفرعية:

الانظمة الفرعية فى مرتبات الضباط القديم هى ثلاثة انظمة فرعية ولكن الانظمة الفرعية فى النظام الجديد تم توسيعه حيث الانظمة الفرعية ما قبل المعالجة هى نظام البيانات المساعدة بالاضافة الى ادارة النظام الذى به بيانات المستخدمين واعطاء الصلاحيات ثم نظام المرتبات ثم البديل النقدى وكل من نظام الارشيف والفروقات هى انظمة مستقلة ولكن يعاد استخدامها فى الانظمة الفرعية الاخرى وهذا شكل من اشكال التعميم والتخصيص الذى تم مناقشته سابقا الشكل يوضح ذلك



الشكل (4-15) :الانظمة الفرعية فى نظام مرتبات الضباط القديم.



الشكل (4-16) :اعادة تصميم الانظمة الفرعية فى نظام مرتبات الضباط الجديد .

5-7-3 تصميم الكائنات Object Design:

في هذه المرحلة، سيتم فحص التصميم الحالي ضد أهداف التصميم لاكتشاف تحسينات في هذا الشأن وفقا لأهداف التصميم في الجدول (5-38) وضعت استراتيجيات مرتبطة بها كما في الجدول (5-39).

الجدول (5-39) الاستراتيجية الموضوعة لكل من المتطلبات الغير وظيفيه

المتطلبات غير الوظيفية لاهداف التصميم	التاثير
التكيف مع الانظمة وتقليل التكرار	التعميم والتخصيص واعادة استخدام الكود
التصميم البديعي لواجهة المستخدم الرسومية	اعادة تصميم الواجهات
توزيع النماذج منفصلة	تصميم التحزيم **Package Design**

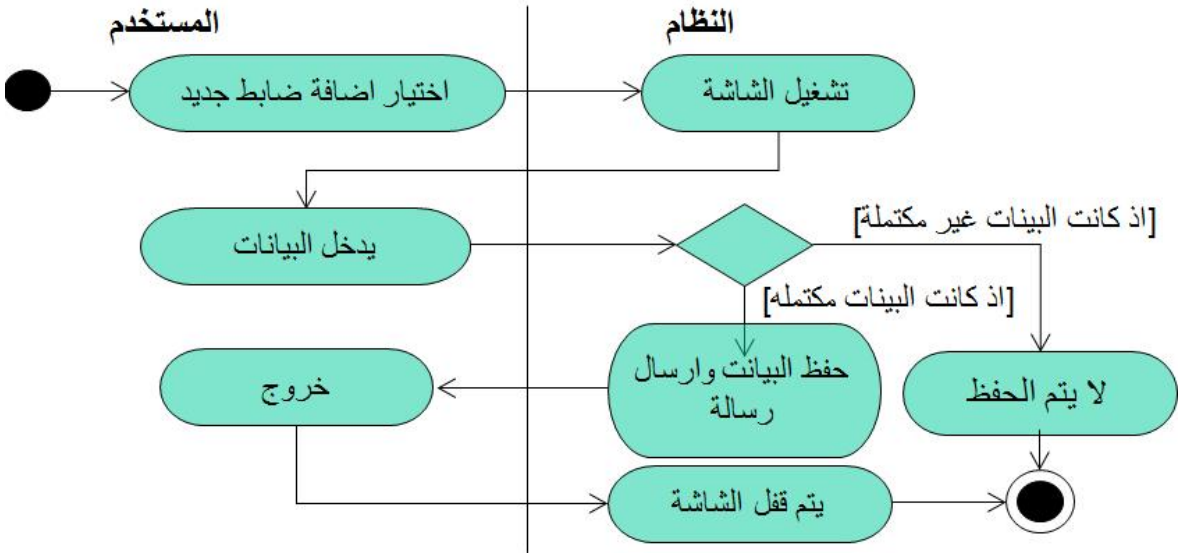
تكرار التطبيقات أمر شائع جدا. لتقليل التكرار يبنى على قاعدة التصميم الغرضيى التوجه وذلك بان التعميم أو التجريد يجب أن يطبق تحت أي وضع ممكن.أيضا، من المهم إعادة استخدام كائن موجود بدلا من خلق رموز جديدة.

تنظيم واجهة المستخدم الرسومية تلعب دورا مهما في قابلية الاستخدام. المراقبة والتعلم والمماثلة لأنظمة هي اختصار لتصميم واجهة المستخدم الرسومية السهلة الاستخدام بسرعة.

5-7-4 مخطط الأنشطه :Activity Diagram

مخطط الأنشطة هو نوع من خرط التدفق التي تستخدم لتصف تدفق عمل النظام بالتـــالي فهـــي تصف تدفق التحكم من نشاط إلى آخر في النظام, وما هي الأنشـــطة التـــي يمكـــن أن تـــؤدى بالتوازي وأي مسار بديل خلال التدفق فهي تدعم الأنشطة المتوازية التي لا تقوم بهـــا خـــرط التدفق العادية.

مخطط الأنشطة يوضح الأفعال والتدفق في التحكم والبيانات بين هذه الأحداث وتستخدم بصورة أساسية في النواحي الديناميكية للنظام مثل التدفق بين وقائع الإستخدام وفي الخطوات المتتابعة في العملية المحوسبة وفي تدفق التحكم لعملية معينة. تركز مخططات الأنشطة على الأنشطة التي تحدث نتيجة لمعالجة داخلية بدلاً عن الخارجية



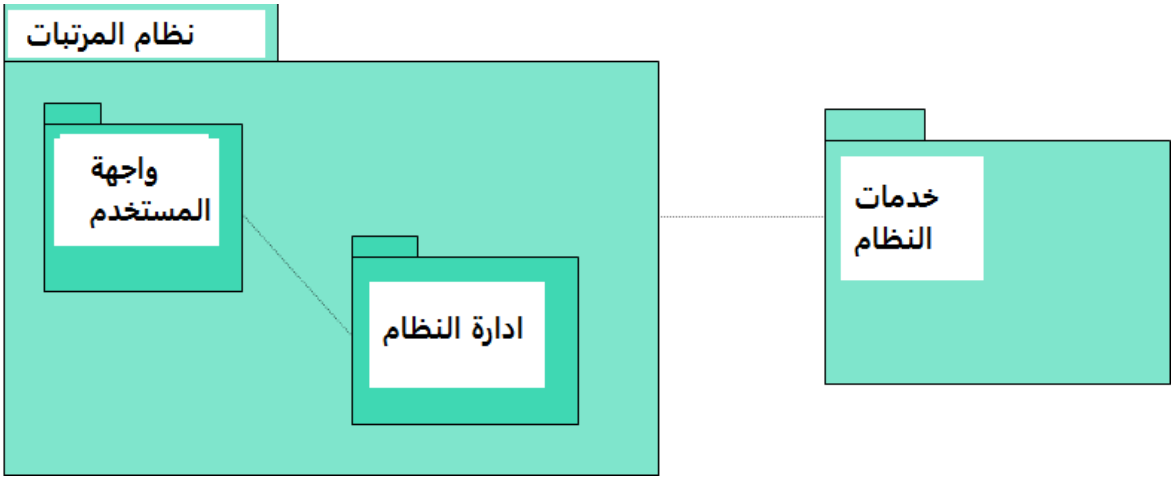
الشكل (5-17) مخطط الانشطه لاضافة ضابط

5-7-5 مخطط التحزيم

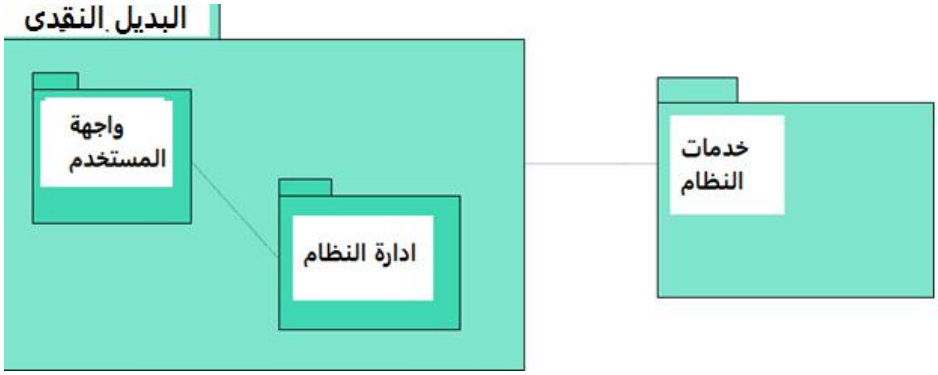
هو عبارة عن مجموعة من عناصر نماذج النظام تجمع هذه العناصر بطريقة دلالايه واخفاء التفاصيل .

مما يظهر من مخطط التخريم ان نظام المرتبات ينقسم الى عدة نظم فرعية ترتبط مع بعضها لاداء الخدمات يقلل ذلك من الارتباط الكبير بين الانظمة وايضا الشكل (5-15) يوضح

الانظمة الفرعية لنظام البديل النقدى رغم وجود علاقة كبيرة بين ملفات النظامين وهما نظام المرتبات ونظام البديل النقدى



الشكل (5-18) مخطط الحزمة لنظام المرتبات



الشكل (5-19) مخطط الحزمة لنظام البديل النقدى

5-7-6 اعادة تصميم واجهة المستخدم

تم تصميم واجهة المستخدم باستخدام اوراكل ديفلوبر6i لقد اصبحت شاشة المستخدم ملونه وذكيه وتم استخدام كل المصلحات الخاصة بالمستخدمين فى المجال المحاسبى بدلا عن استخدام مصطلحات الحاسوب مثل اضافة استحقاق خصم استحقاق مراجعة واعتماد وتعمل هذة الشاشات بتناسق تام وبطريقة سهلة وعمل كل البرامج يكون بطريقة واحدة مثلا انزال البيانات ثم المراجعة ثم الاعتماد لو عرفته فى برنامج فرعى واحد يمكن ان تعمل على البرنامج الاخر دون مساعدة من خلال التنبو ايضا وضع فى الاعتبار خطا المستخدم بالتالى يمكن الرجوع وفق قيود محددة وايضا وفرنا مساعدة فى بعض الشاشات لكيفية استخدام الشاشة والمفاتيح مراعاة لتنوع

المستخدين واختلاف المستوى التعليمى والخبرة والمهارات وتم تطوير هذة الشاشات بعد عقد اكثر من لقاء مع المستخدين للاستفادة من خبرات المستخدمين حتى ان بعض المستخدمين ذكروا بان لهم دور كبير فى تصميم هذه الشاشات .

صممت الشاشات بطريقة بسيطة، هذا التصميم يجعل من السهل للتبديل بين وظائف مختلفة. بصريا،

يجعل واجهة المستخدم الرسومية تبدو أكثر تنظيما ومريحة.

5-7-6 -1 دعم المستخدم

تم انشاء دليل للمستخدم به مجموعة من الارشادات للمستخدم وايضا تعليمات فورية عند الانزال والحفظ او اجراء اى عملية وايضا رسائل بتسلسل تنفيذ الاحداث فى الحالات التى يكون فيه الامر ينفذ اكثر من حدث مثلا عند اعتماد اضافة ضابط اولا يتم انزال البيانات الاساسية من بيانات الادخال المسبق بطريقة تلقائية مثل الانزال فى ملف المرتبات ثم اضافة الاستحقاقات ثم الاستقطاعات ثم يحسب الصافى ففى كل حلة يعطى رسالة بذلك .

5-7-6 -2 توثيق المستخدم

تم توفير معلومات ورقية مع النظام و تصميم وثائق وتكون الوثائق وفقا للانواع الاتية :

5. وصف وظيفى :وصف موجز لما يمكن ان يفعله النظام .
6. دليل استخدام تقديمى : مقدمة للنظام .
7. مرجع دليل استخدام النظام : وصف كل امكانات النظام بالتفصيل .
8. دليل تثبيت النظام وصف كيفية تثبيت النظام .

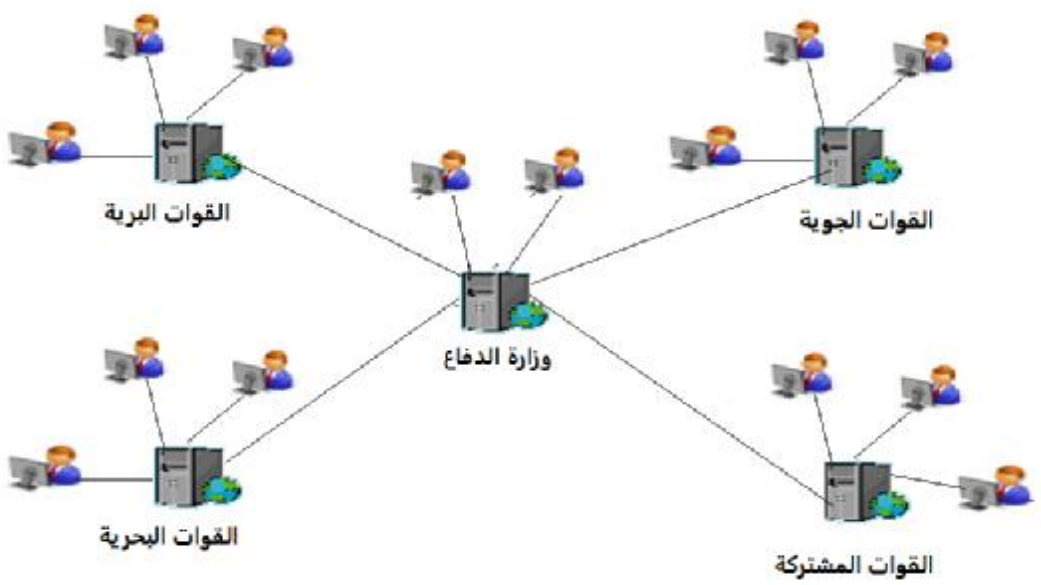
5-7-7 تصميم قواعد البيانات

لقد كان النظام يعمل بقاعدة بيانات اوراكل9i وقد تم ترقية البرنامج الى اوراكل 11g باعتباره اخر تقنية فى قواعد البيانات عند اجراء هذا البحث .

5-8 توزيع الانظمة للادارة العامة للشئون المالية

كل الانظمة فى الادارة العامة هى انظمه موزعه والأنظمة الموزعة هي عباره عن انظمه تعمل على العديد من الأجهزة مرتبطة مع بعض بشبكه.,وتدعم النظم الموزعة مشاركة الموارد ,الانفتاح والالتقاء والتزامن والتعاون والنمو وسماحية الخطأ والشفافية.

حيث نجد انة توزع هذه الانظمة باستخدام معماريات الخادم العميل حيث توجد شبكة داخلية لكل من القوات البرية/البحرية/الجوية/المشتركة ووزارة الدفاع مربوطة بمخدم محلى كما تبين الاشكال التالية هذا الترابط وكل قوات ترتبط بكل وحداتها الداخلية والخارجية المربوطة بها البرامج .



الشكل (5-21) توزيع انظمة الشئون الماليه.

الشكل (5-21) الشاشة الرئيسية لبرنامج المرتبات

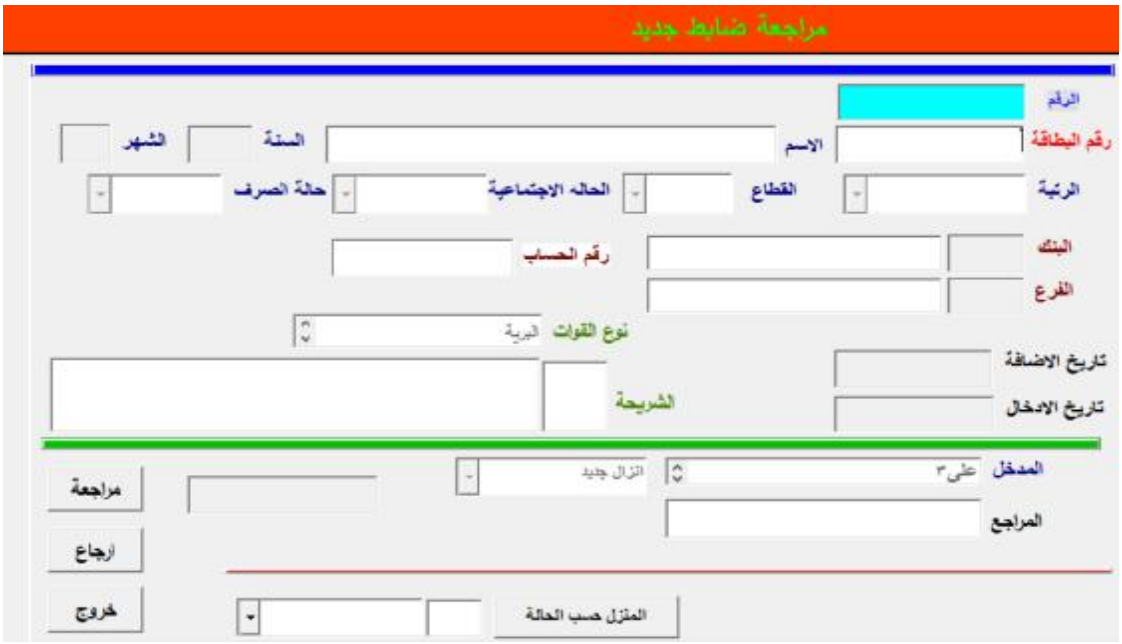
الشكل (5-22) شاشة تفاصيل الخدمات

الشكل (5-23) شاشة تفاصيل ادارة النظام

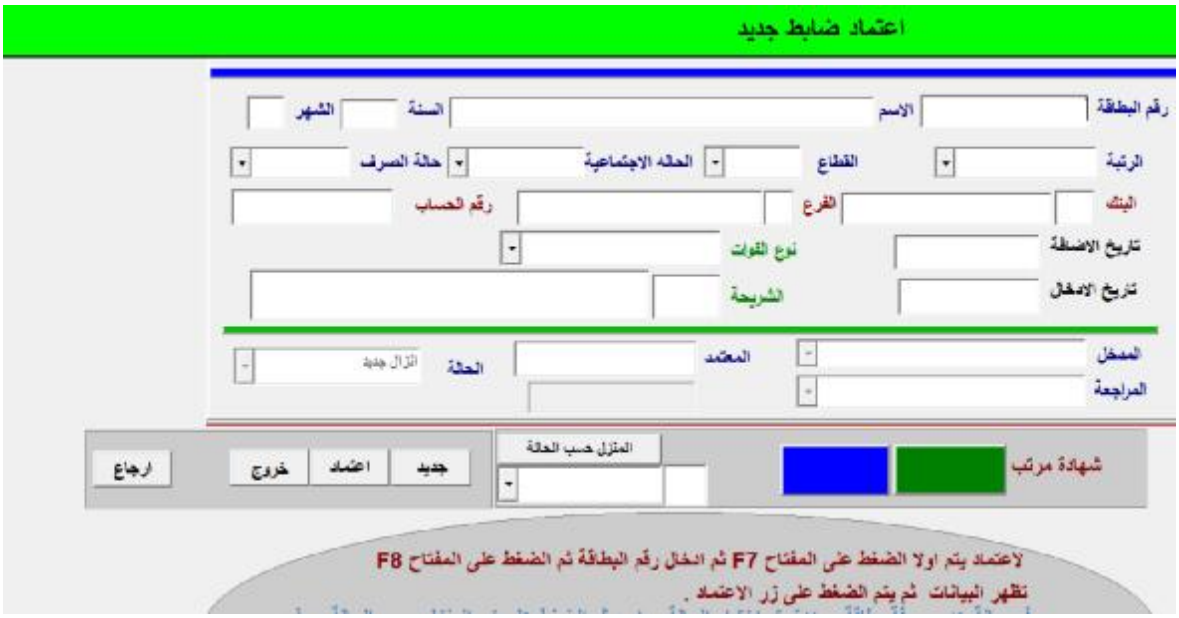
الشكل (5-24) شاشة تفاصيل الخدمات المساعدة

الشكل (5-25) شاشة تفاصيل البحث

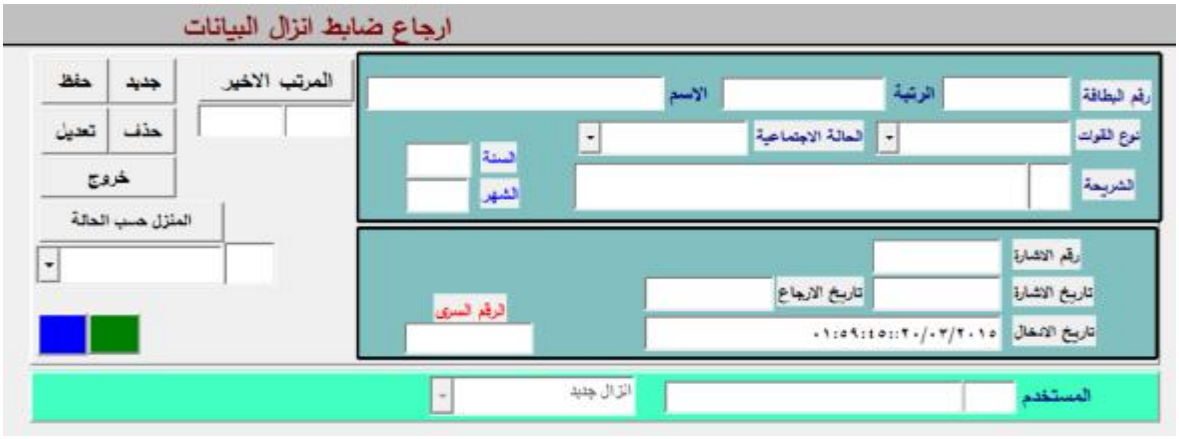
الشكل (5-26) شاشة ادراج ضابط جديد



الشكل (5-27) شاشة مراجعة ضابط جديد



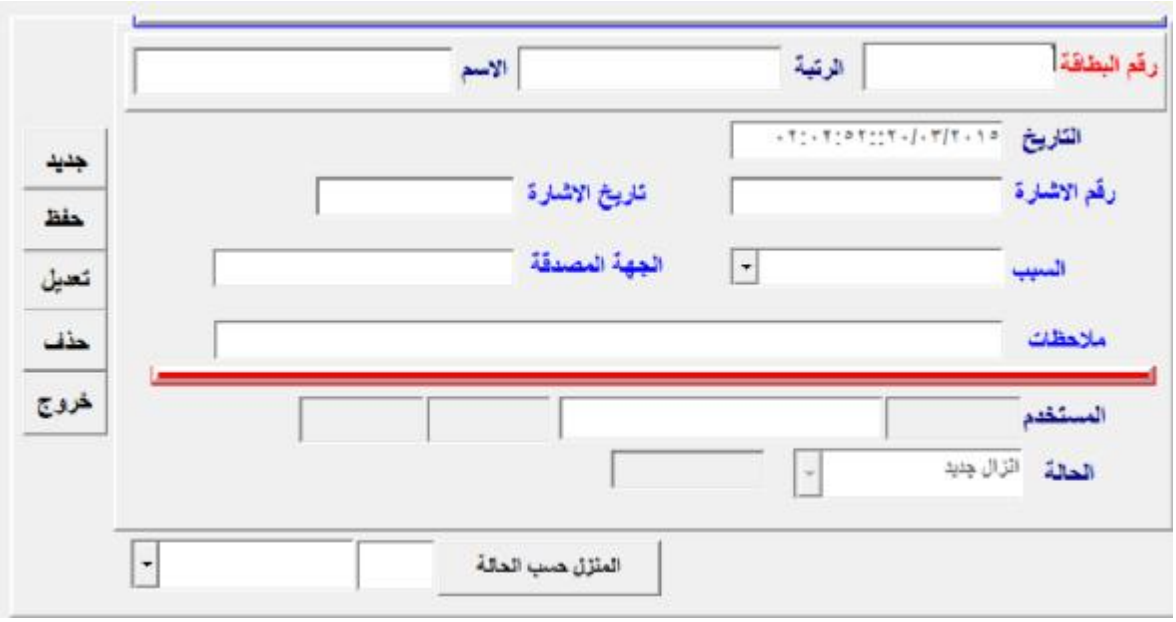
الشكل (5-28) شاشة اعتماد ضابط جديد



الشكل (5-29) شاشة ارجاع ضابط جديد

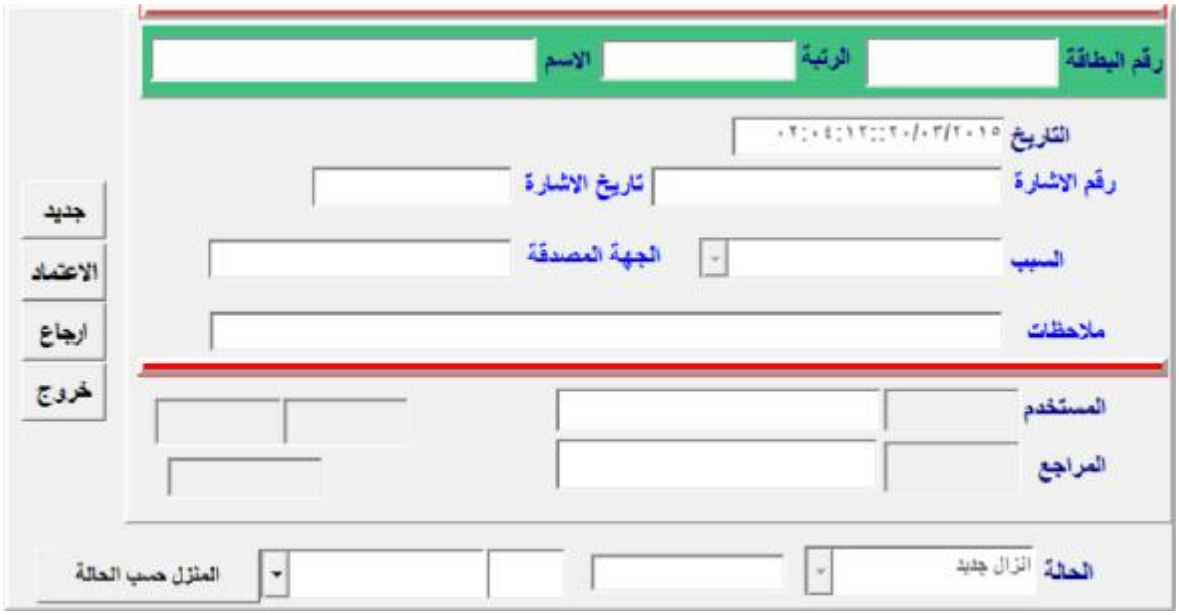
الشكل (5-30) شاشة ارجاع ضابط المراجعة

الشكل (5-31) شاشة ارجاع ضابط الاعتماد

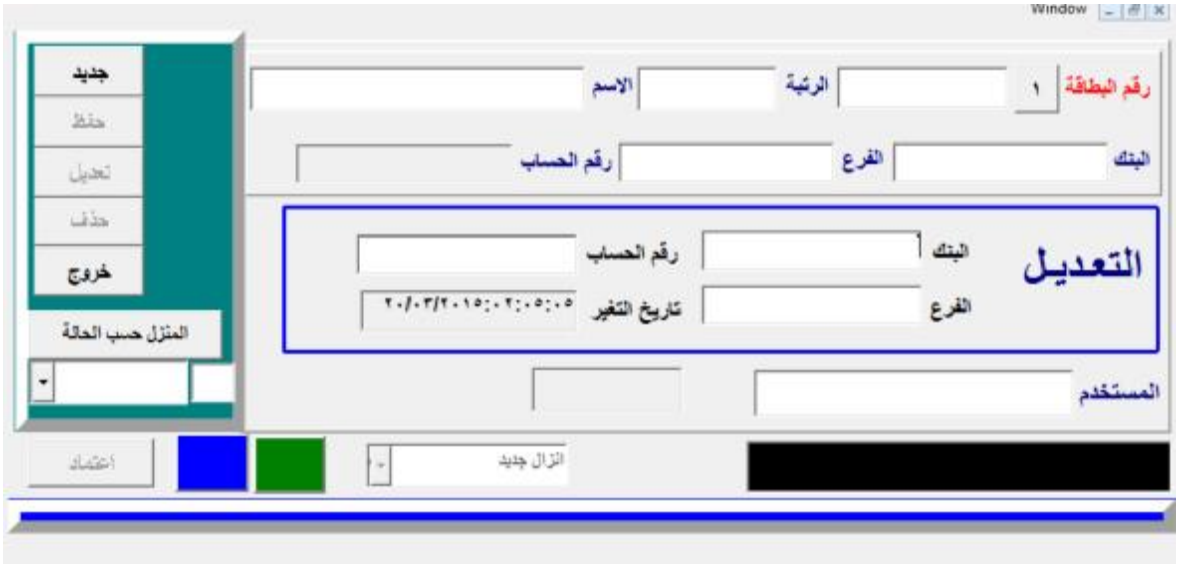


الشكل (5-32) شاشة قك مرتب ضابط

الشكل (5-33) شاشة قك مرتب ضابط المراجعة



الشكل (5-34) شاشة قك مرتب ضابط الاعتماد



الشكل (5-35) شاشة تعديل القطاع والرتبة.

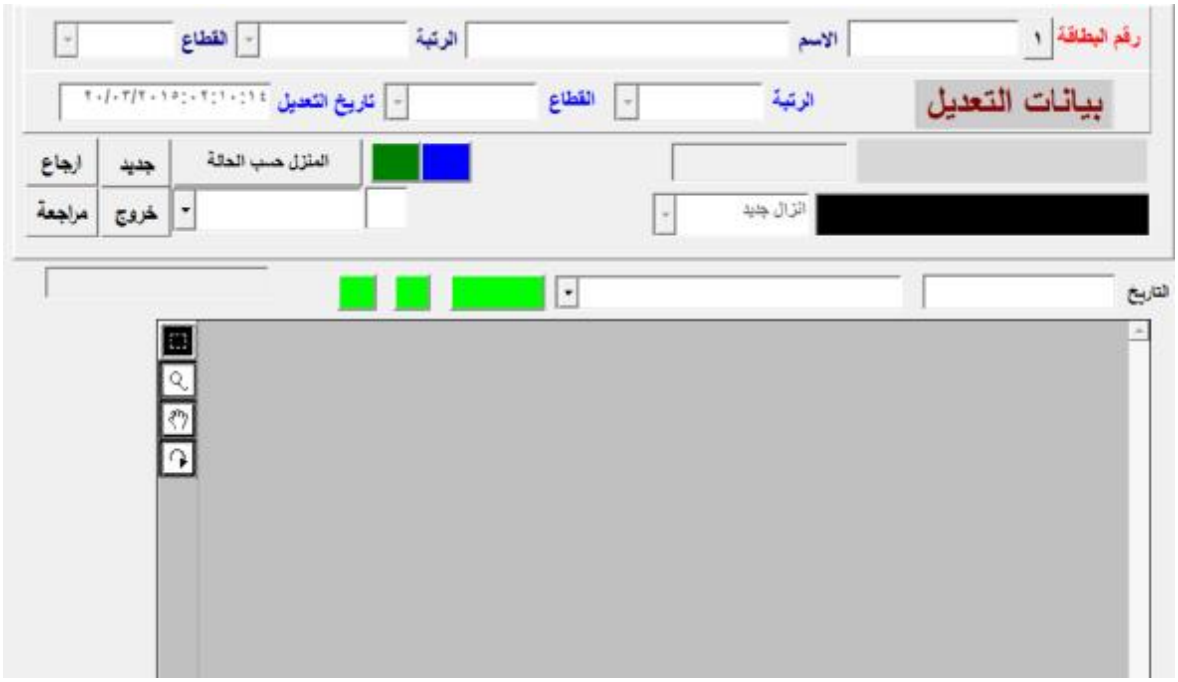
الشكل (5-36) شاشة تعديل القطاع والرتبة المراجعه.

الشكل (5-37) شاشة تعديل القطاع والرتبة الاعتماد.

الشكل (5-38) حذف/اضافة/استقطاع/استحقاق.

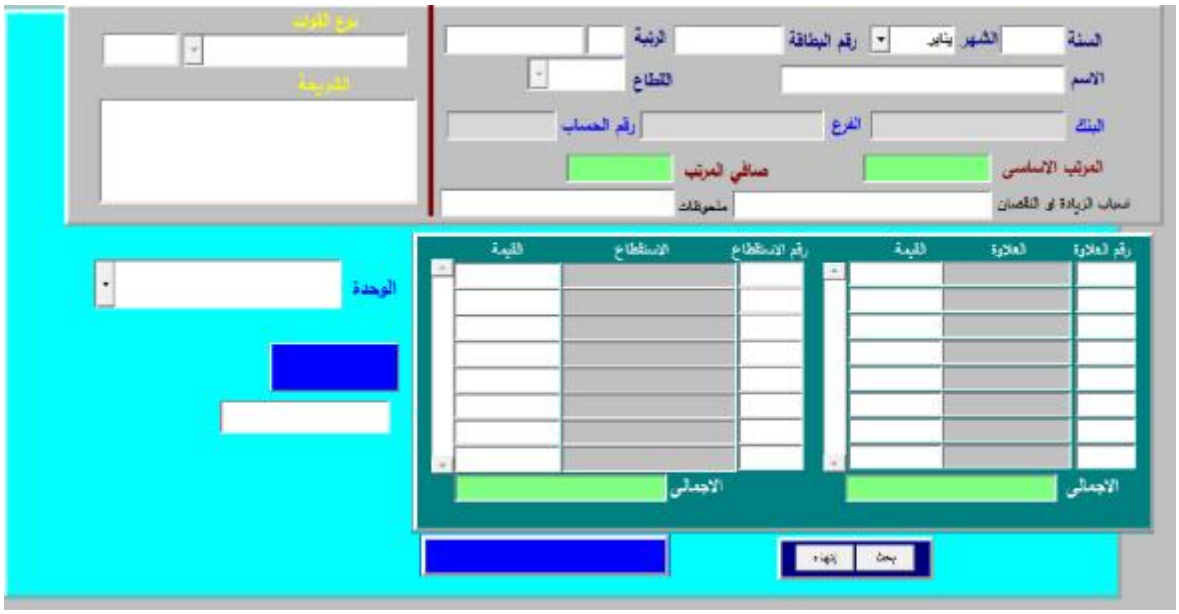
الشكل (5-39) حذف/اضافة/استقطاع/استحقاق المراجعه.

الشكل (5-40) حذف/اضافة/استقطاع/تعديل القطاع والرتبة.



الشكل (5-41) حذف/اضافة/استقطاع/تعديل القطاع والرتبة المراجعه.

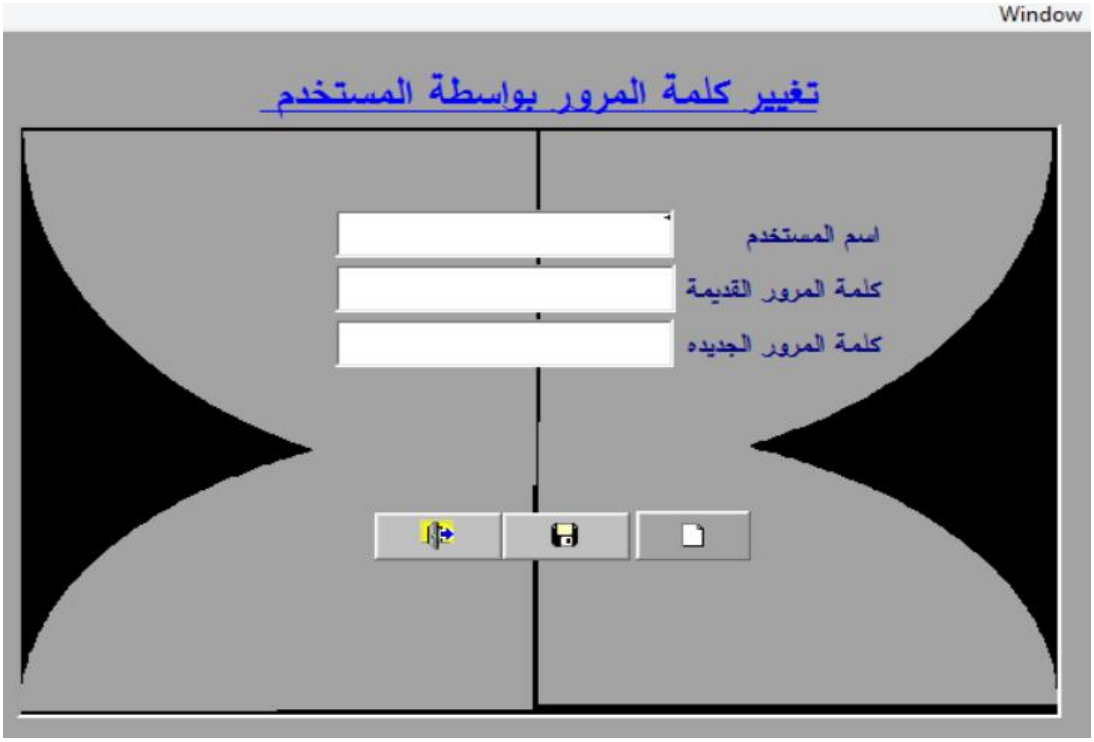
الشكل (5-42) شاشة بحث البيانات الاساسية.



الشكل (5-43) شاشة بحث المرتبات الشهريه.

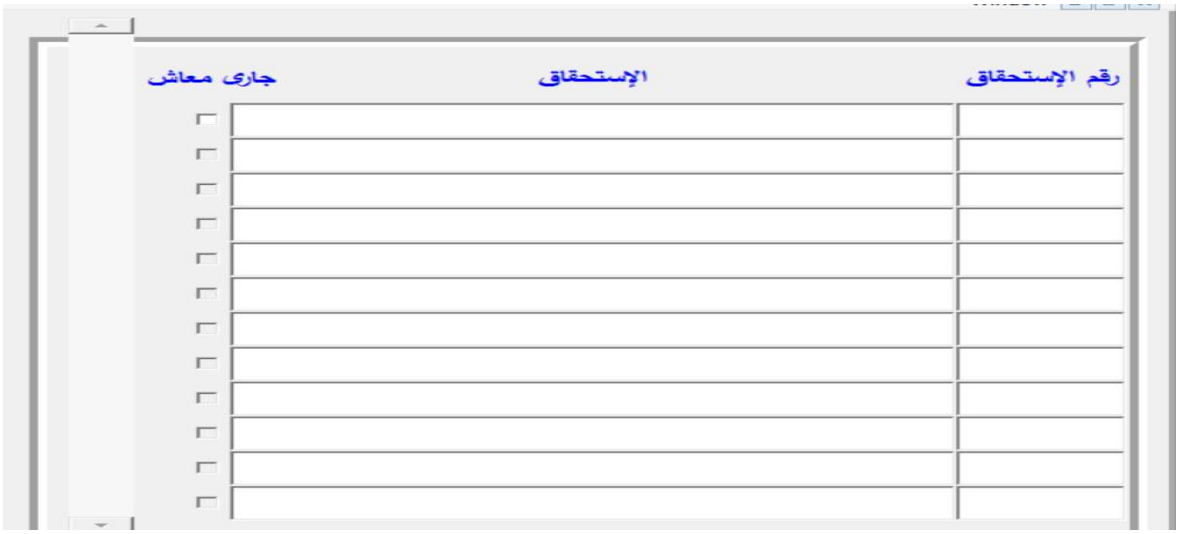
الشكل (5-44) اضافة مستخم جديد.

الشكل (5-45) شاشة اعطاء الصلاحيات.



الشكل (5-46) شاشة تغير كلمة المرور.

الشكل (5-47) شاشةاضافة البنوك وفروعها.



الشكل (5-48) شاشةاضافة اسماء الاستحقاقات.

الشكل (5-49) شاشةاضافة اسماء الاستقطاعات.

الشكل (5-50) شاشة قيم المرتب الاساسى.

5-9 التحقق واثبات الصحة Verification and Validation

التثبت والتحقق هما التيقن من ان نظم المعلومات تلبى احتياجات المستخدم ويرمز لها بالرمز V&V اختصارا للكلمتين .

بالرغم من ان الترجمة لهاتين الكلمتين تقتربان من بعضهما البعض.الا ان هنالك فرق بين المصطلحين فى اللغة , فكلمة التحرى verify تعنى ان يتحقق من او يثبت وكلمة التحقق validate تعنى ان يويد او يثبت او يجعله شرعيا وفيما يلى توضيح الفرق :

التحرى او التثبت Verification :يجب على السوال هل نبنى المنتج بطريقة صحيحة بمعنى ان البرمجيات يجب ان تطابق مواصفاته .

اما التحقق Validation :يجب على السوال هل نبنى المنتج الصحيح بمعنى أن البرمجيات يجب ان تفعل ما يريده المستخدم حقا .

التحقق والتثبت ليس بنفس المعنى التثبت يعنى التطابق مع المواصفات ويبين التحقق ان البرنامج يلبى حاجة المستهلك .

5-9-1 التيقن V&V process

فى كل عملية دورة حياة يجب تطبيق V&V فى كل مرحلة من عمليات البرمجيات وله هدفان:

1.اكتشاف خلل او قصور فى النظام .

2.تقدير او تخمين ما اذا كان النظام يمكن استخدامه فى مواقف التشغيل ام لا .

5-9-2 التيقن التثبت الاستاتيكى والديناميكى static and dynamic Verification

تهتم عملية التفتيش البرمجيات بتحليل تمثيل النظام الساكن باكتشاف المشاكل وقد تكتمل بتحليل معتمد على الادوات للمستند وشفرة البرنامج .

تهتم عملية اختبار البرمجيات بتجربة ومراقبة تصرف المنتج حيث يتم تشغيل وتنفيذ النظام ببيانات اختبار وتتم مراقبة تصرفه التشغيلى .

5-9-3 اختبار البرنامج program testing

يمكن ان يبين الاخطاء وليس غيابه , ويعد الاختبار الناجح هو ذلك الاختبار الذى يكتشف خطا واحدا او اكثر من خطا, وهو التقنية الوحيدة القابلة للتحقق من المتطلبات غير الوظيفية ويجب ان يستخدم مقترنا مع التثبيت الساكن لتقديم تغطية تيقن كاملا .

5-9-4 انواع الاختبارات types of testing

1.اختبارات الخلل defect testing:

وهى اختبارات مصممة لاكتشاف خلل النظام ,واختبار الخلل الناجح هو الذى يظهر وجود خلل فى النظام.

2.اختبارات احصائية statistical testing :

وهى اختبارات مصممة لتعكس تواتر مدخلات المستخدم ,وتستخدم لتقدير وتقيم الاعتمادية .

5-9-5 أهداف التيقن v&v Goals

يجب ان يثبت التيقن الثقة فى ان البرمجيات تلائم الغرض المصصمة من اجله,ولا يعنى هذا الخلو من العيوب او الخلل تماما ,بل يجب ان تكون جيده كفاية لمستخدمها المقصود واستخدامها المستهدف , وسيحدد نوع الاستخدام درجة الثقة المطلوبة .

5-9-6 ثقة التيقن v&v confidence

تعتمد على الغرض من النظام او وظيفة وتوقعات المستخدم وبيئة التسويق .

وظيفة البرمجيات : يعتمد مستوى الثقة على مدى حرج البرمجيات فى الموسسة .

توقعات المستخدم: قد تكون للمستخدمين توقعات قليلة من نوع معين من البرمجيات .

بئية التسويق :قد يكون دفع المنتج الى الاسواق مبكرا اكثر اهمية من اكتشاف الخلل فى البرنامج .

5-9-7 التحرى او التثبت Verification

تم التحرى والتثبت وذلك من خلال العمل الفعلى للنظام فى خط الانتاج وادخال البيانات الحقيقية فوجدت بعض الاخطاء عند ادخال اسم مركب لا يتم الحفظ .

وايضا عند عملية ارجاع ضابط كان له مرتب وتم أيقافه ثم الارجاع مرة اخرى يرجع البيا نات برتبة قديمة وقطاع قديم وتكون الرتبة الحالية مختلفة .

ايضا من المفروض عند اجراء اى عملية توثر على المرتب بالاضافة او النقصان يتم تعديل صافى المرتب هنالك بعض العمليات لا يتم فيه ذلك .

ايضا هنالك بعض عمليات الاعتماد لا تتم فى البرنامج وخلاف ذلك الان يعمل النظام بطريقة جيده ويودى الوظائف المطلوبة ويقوم باعداد المرتبات والفروقات بطريقة دقيقة تم مقارنة الارقام من قبل المراجعيين

ايضا تعمل الشاشات بصورة جيدة تلبى متطلبات المستخدم وتم معرفة ذلك من خلال عدة لقاءات مع المستخدمين والمحاسبين والمراجعين .

5-9-8 تحليل التثبت

نجد ان عدم ادخا الاسم يرجع لضيق الحقل فتمت توسيع الحقل والشاشة المصاحبه له باخذ كل التوقعات الممكنه لسعت الحقل .

ايضا الحالة الثانية تم الحل بوضع خيارات امام المستخدم وتم وضع تنبيهات لذلك اما الحالة الحالة لعملية تحديث المرتب تم مراجعة الكود ولمن كان صحيح ولكن ربما تكون هنالك خطا فى ادخال البيانات او انقطع الاتصال اثناء اجراء العملية فلم نعرف الاسباب الجازمة لذلك ولكن وضعنا حل وهو عملية التحديث قبل اجراء اى اعداد للمرتبات ووضعت ضوابط برمجية تحل هذة المشكلة ولكن لم تغطى الجوانب.

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الباب السادس

نموذج الربط المقترح

6-1 المقدمة

قام الباحث بتعديل نموذج الربط الذى قدمه ظاهر شاهر يوسف القشي باشراف الأستاذ الدكتور نعيم دهمش فى رسالة دكتورا باسم مدى فاعلية نظم المعلومات المحاسبية في تحقيق الأمان والتوكيدية والموثوقية في ظل التجارة الإلكترونية

هدفت الدراسة إلى ما يلي:

1- التعرف على المشاكل التي تواجه أنظمة المعلومات المحاسبية في ظل استخدام التجـــارة الإلكترونية.
2- تطوير نموذج للربط بين نظام المعلومات المحاسبي والتجارة الإلكترونية .
3- اقتراح بعض التوصيات المناسبة والكفيلة بتقوية النظام المحاســبي الــذي يــتم ربطــه بالتجارة الإلكترونية .

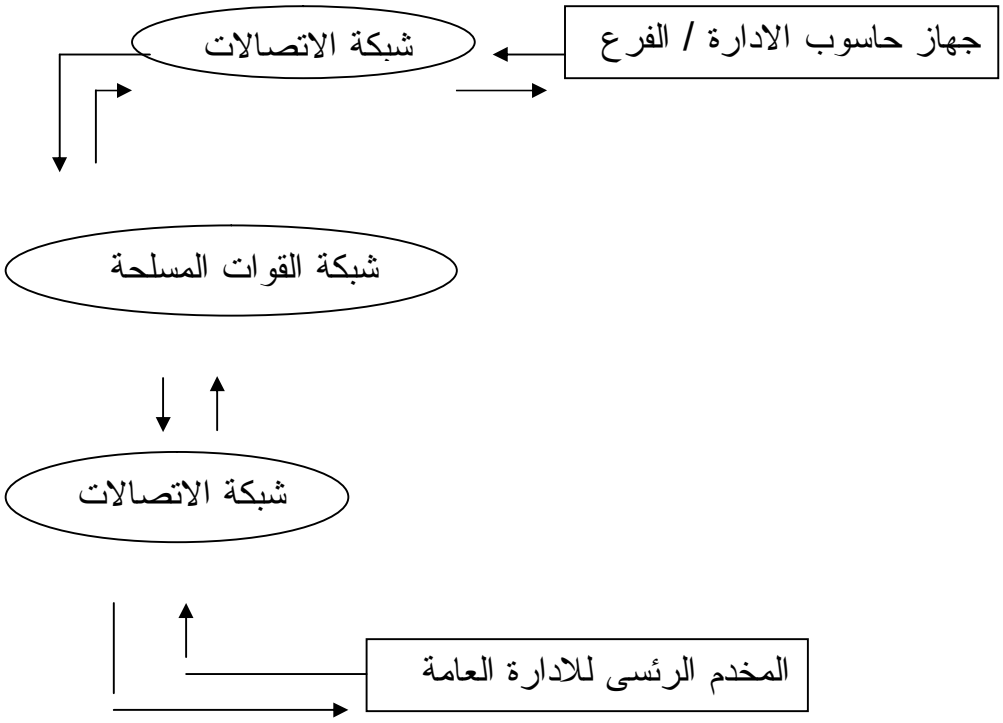
ولتحقيق هذه الأهداف قام الباحث بالاطلاع على معظم الدراسات السابقة المتعلقة بموضوع التجارة الإلكترونية وما يرافقها من مشاكل تقنية ومحاسبية وسبل معالجتها ، وقام أيضا بتحليل إحدى أهم مشاريع تدقيق التجارة الإلكترونية الذي تم بالاشتراك بين معهد المحاسبين القانونيين الأمريكي ومعهد المحاسبين القانونيين

استفاد الباحث من الدراسه اعلاه باضافة بعض التعديلات عليه لإيجاد حلول مناسبة للمشاكل لربط البرامج المالية الوصول إلى تطوير مشروع نظام يربط بين نظام الادارة العامة للشئون المالية وموقعها المختلفة فى ولايات السودان.

6-2 مشروع تطوير نظام الربط بين الادارة العامة للشئون المالية والافرع والادرات

6-2-1 دورة الاتصال بين الزبون/العميل

من خلال النظر إلى الشكل (6-1) يمكن التعرف وبشكل مبسط على آلية الاتصال والتعامل بين الادرة/الفرع والادارة العامة ، وكما نلاحظ بأن عملية الاتصال مربوطة بينهما من خلال عدة وسائط يمكن وصفها بالوسائط الإلكترونية التكنولوجية غير الملموسة.



الشكل (6-1): دورة الاتصال بين الادارة/الفرع والادارة العامة

تتم دورة الاتصال وإتمام العمليات بين الادارة/الفرع والادارة العامة بالشكل التالي:

تبدأ بدخول الادارة/الفرع إلى جهازالادارة العامة الموجود على شبكة القوات المسلحة من خلال جهاز الحاسوب الخاص به والمربوط عبر شبكة الاتصالات المحلية ثم الشبكة الخاصة بالقوات

المسلحة ، ومن ثم استخدام الالية المبرمجة من قبل الادارة العامة للحصول على الخدمة التي يرغب بها.

يتلقى نظام الادارة العامة وبشكل مباشر طلب الادارة/الفرع ثم يقوم النظام بتنفيذ الطلب بشكل سريع جدا مستندا على سياسات وإجراءات مبرمجة بشكل مسبق.

يتلقى الادارة/الفرع عبر الشبكة رسالة إلكترونية تتضمن كلا من : موافقة نظام الادارة العامة على العملية ، والإجراءات الواجب عليه اتباعها للحصول على الخدمة.

بالأخذ بحقيقة أن العمليات المالية عبر الشبكة ذات طابع سريع وغير ملموس ، وحقيقة أن الانظمة المالية انظمة عامة لا تحدها قوانين محلية يمكن أن نستنتج بأن نظام الادارة العامة المالى معرض لمخاطر كثيرة لا يمكن حصرها ؛ ولكن يمكن تحديد إطارها بالشكل التالي:

1- **مشاكل تقنية** ، تتضمن مشاكل حماية النظام من الاختراقات بشتى أنواعها وأشكالها ، وحماية النظام أمر حساس لا يمكن الاستغناء عنه حيث إن اختراقه يولد عواقب خطيرة يمكن تلخيص آثارها بالشكل التالي:

- فقدان ثقة الادارة/الفرع بالادارة العامة.
- دخول فيروسات لنظام الشركة؛ مما قد يؤدي إلى فقدان البيانات.
- خسائر قد تكون كبيرة بشكل قد يؤدي إلى انهيار النظام المالى.

2- **مشاكل محاسبية** ، تتضمن:

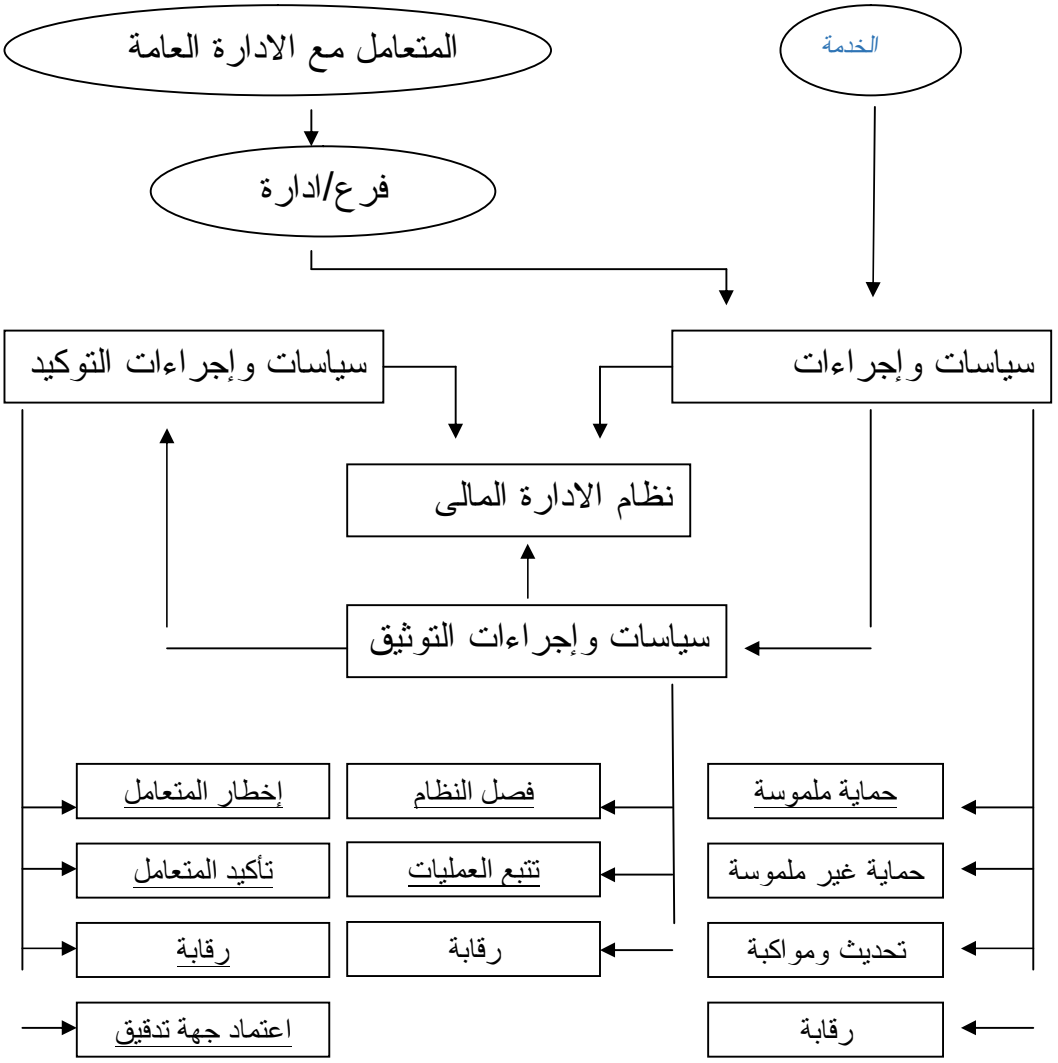
- **الاعتراف** بالإيراد والمصروفات، يعد الإيراد المتولد عبر الشبكة إلكترونية إيرادا مشكوكا فيه من قبل جميع جهات الرقابة.

3-6 الأسس التي يعتمد عليها مشروع النظام المقترح:

بما أن الانظمة المالية للادارة العامة المتعاملة عبر الشبكة يعتمد على عنصرين رئيسيين وهما:

1- الخدمة التى تقدمها الادارة العامة.
2- المتعامل مع الادارة العامة (الادارةأو الفرع).

فسوف يعتمد الباحث هذين العنصرين كأساس انطلاق لتصميم مشروع نظام الــربط الموضــح بالشكل (6-2).



الشكل (6-2) : مشروع نظام الربط

كما يلاحظ من الشكل رقم (6-2) بأن هنالك دورتين رئيستين ينبع منها ثلاث دورات فرعية ، وكل دورة فرعية مكونة من عدد من الأجزاء كالتالي:

6-3-1 دورة السياسات والإجراءات المتعلقة بالخدمة والتي ينبع منها الدورات الفرعية التالية:

1. دورة سياسات وإجراءات الحماية.
    1. حماية ملموسة.
    2. حماية غير ملموسة.
    3. تحديث ومواكبة.
    4. رقابة.
2. دورة سياسات وإجراءات التوثيق.
    1. فصل النظام.
    2. تتبع العمليات.
    3. رقابة.
3. دورة سياسات وإجراءات التوكيد.
    1. إخطار المتعامل.
    2. تأكيد المتعامل.
    3. رقابة.
    4. اعتماد جهة تدقيق.

6-3-2 دورة السياسات والإجراءات المتعلقة بالادارة /الفرع والتي ينبع منها الدورات الفرعية التالية:

1. دورة سياسات وإجراءات الحماية.
    1. حماية ملموسة.
    2. حماية غير ملموسة.
    3. تحديث ومواكبة.
    4. رقابة.
2. دورة سياسات وإجراءات التوثيق.
    1. فصل النظام.
    2. تتبع العمليات.

3. رقابة.

3. دورة سياسات وإجراءات التوكيد.

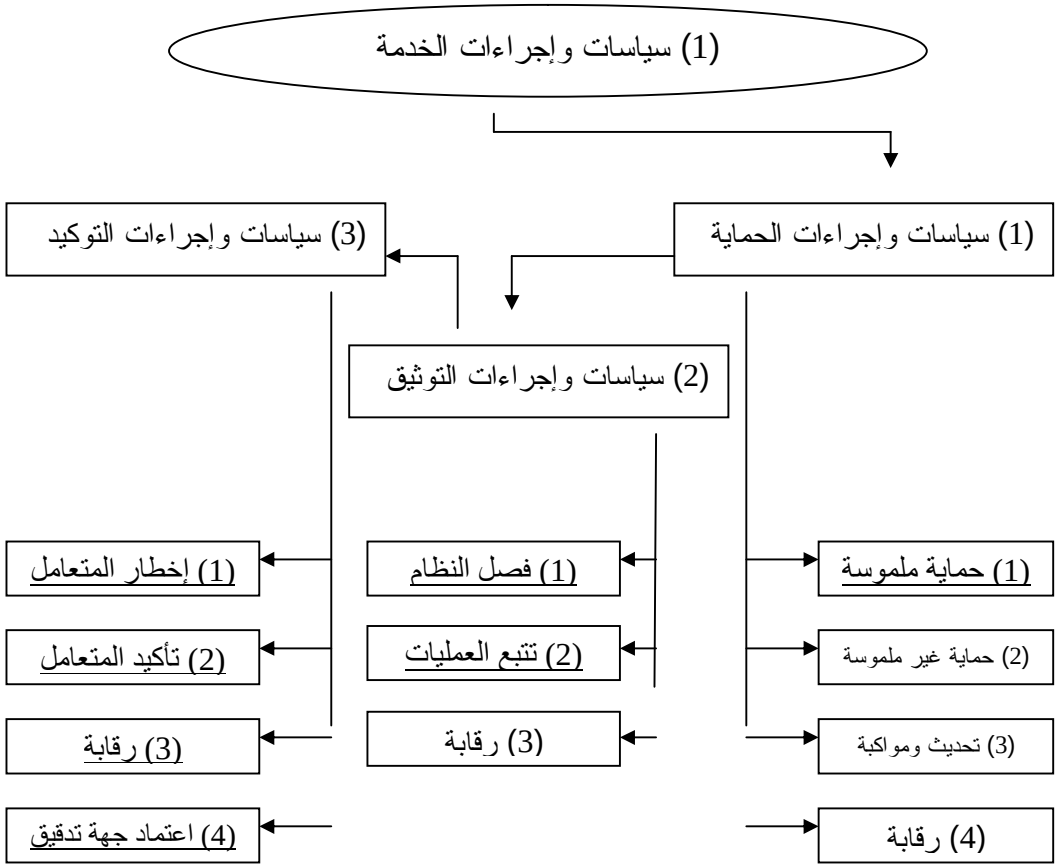
1. إخطار المتعامل.
2. تأكيد المتعامل.
3. رقابة.
4. اعتماد جهة تدقيق.

6-4 شرح النموذج بالتفصيل

سوف يقوم الباحث بشرح الدورات الرئيسية من خلال البدء بوضع نموذج رئيسي لكل دورة منها مع التعليق عليه ؛ ولكي يتمكن القارئ من تتبع النموذج فقد قام الباحث بوضع رقم أمام كل فقرة بالنموذج ، وكمثال توضيحي وبالنظر إلى الشكل رقم (5-3) ففقرة (1-1-1) تعود إلى (1) سياسات وإجراءات الخدمة و(1) سياسات وإجراءات الحماية و(1) الحماية الملموسة وبالتالي فان فقرة (1-1-1) تعني سياسات وإجراءات الحماية الملموسة للخدمة.

6-4-1: دورة السياسات والإجراءات المتعلقة بالخدمة

لكي نتمكن من تتبع دورة السياسات والإجراءات المتعلقة بالخدمة لا بد من التقيد بالدورة الموضحة بالشكل رقم (5-3) ، وبالشكل التالي:



الشكل (5-3):دورة السياسات والإجراءات المتعلقة بالخدمة

قبل البدء بشرح هذه الدورة لا بد من التعرف على ماهية الخدمة ، ويستطيع الباحث أن يعرف الخدة بأنها "جميع البرامج المالية من البرامج المحاسبية والميزانية" وتشمل التالي:

- البرمجيات الخاصة بجميع االاعمال المالية عبر شبكة الادارة مثل:
  - برنامج المحاسبة .
  - برنامج الميزانية .
  - برنامج مرتبات الضباط .
  - برنامج مرتبات الصف والجنود.
  - برامج السجلات.
  - برامج المعاشات.

ويود الباحث أن ينوه بأن هذه البرامج فريدة الشكل والخصائص اذ تم تصميمها ثم تعميمها على كل الادرات والافرع والقوات البرية والبحرية والجوية والمشتركة ووزرارة الدفاع.

6-4-1-1: سياسات وإجراءات الحماية الملموسة للخدمة

الحماية مقصورة على حماية البرمجيات نفسها ، وبما أن الخدمة غير ملموسة الطابع ، فسياسات وإجراءات الحماية الملموسة عليها مقصورة على حماية كل من:

1- الاختراع ، وذلك بالحصول على براءة اختراع للبرنامج الذي يتم تطويره.
2- توثيق جميع السياسات والإجراءات الخاصة بهذه الخدمه لدى الجهات المعنية بالدولة.
3- إنشاء قسم خاص بالادارة العامة تناط به مهمة تطوير هذه البرمجيات.
4- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-1-2: سياسات وإجراءات الحماية غير الملموسة للخدمة

يقصد بالحماية غير الملموسة جميع السياسات والإجراءات الواجب على الادارة العامة توفيرها لحماية البرمجيات والتي يمكن تلخيصها بالشكل التالي:

1- تحديد الأشخاص المصرح لهم بدخول نظام الحاسوب (الخدمة) وتوثيق أسمائهم بلائحة إلكترونية خاصة لدى إدارة العامة.
2- برمجة عملية الدخول للنظامين وتقييدها بأرقام سرية مكونة من عدة خانات بحيث لا تقل عن ثماني خانات ، وجعل جزء من الخانات رقمية والأخــرى حرفية ، بحيث يكون من الصعب على أي موظف غير مصرح له يريد اختراق النظام اختراقه نظرا لتعدد الاحتمالات.
3- برمجة عملية الدخول إلى النظامين بالأرقام السرية بشكل يجعلها عملية ذات محاولات محدودة ، بحيث يتوقف النظام عن الاستجابة إن حاول أحدهم الدخول بأرقام سرية خاطئة لأكثر من محاولتين أو ثلاث محاولات.
4- برمجة آلية عملية الدخول إلى النظامين بالأرقام السرية بشكل تمكن الموظف المصرح له بتغيير الرقم السري الخاص به كلما دعت الحاجة إلى ذلك خوفا من تسربه للآخرين.

5- تزويد النظامين بآلية مبرمجة تمكن الإدارة من معرفة أوقات دخول النظام ومن قبل من ؛ وذلك لتسهيل عملية تحديد المسؤوليات.

6- برمجة آلية إبلاغ عن أي اختراق أو محاولة اختراق تتم وربطها مع القسم المختص بشكل مباشر.

7- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-1-3: سياسات وإجراءات تحديث ومواكبة التغيرات التكنولوجية الخاصة بنظم حماية الخمة

نظرا لان التطور سريع فى البرمجيات والشبكات فإنها تعمل ببيئة ذات طابع سريع التغير والتطور ؛ ولهذا يجب على الادارة العامة مواكبة تلك البيئة بعمل التالي:

1- إنشاء قسم خاص يناط به مهمة متابعة التغيرات التكنولوجية وتحديث نظـــام الادارة العامة وفقا لتلك التغيرات.

2- يجب على الادارة العامة وعند بدء عملية تصميم نظامها الأخذ بعين الاعتبار إمكانية حدوث تطورات مستقبلية وبالتالي جعل نظامها نظاما مفتوحا قابلا للتطوير ، وبهذا تتمكن من تفادي التكاليف الإضافية بإنشاء نظام جديد عند حدوث أي تغيرات جوهرية بالبيئة التي يعمل فيها النظام.

3- الاشتراك في جميع مواقع الشركات المتخصصة بالبرمجيات ، والهدف من ذلك الحصول على جميع التقنيات الحديثة وعند صدورها ودون أدنى تأخير.

4- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الشركة الإلكتروني.

6-4-1-4: سياسات وإجراءات رقابية لحماية الخدمة

تعتبر السياسات والإجراءات الرقابية الخاصة بحماية الخدمة من السياسات والإجراءات الضرورية والهامة ، وذلك كي تتمكن الادارة العامة من متابعة أداء النظام وتقييمه بشكل منطقي وفاعل ، ويمكن أن تتكون تلك السياسات والإجراءات من التالي:

1- وضع آلية عمليات الخدمة بالشكل التالي:

- وصول طلب الخدمة الى الادارة العامة عبر الشبكة إلى المخدم الرئسى .

- قبل أن ينفذ النظام العملية يجب أن يتأكد من صحة المعلومات عبر آلية تتبع العمليات في الفقرة (1-2-3) ومن ثم ينفذ العملية.
- بعد أن ينفذ النظام العملية يقوم بإصدار أمر إلكتروني.

2- مراجعة اللائحة الإلكترونية التي تحوي أسماء الأشخاص المصرح لهم بالدخول إلى النظام بشكل دوري ، وبرمجتها بشكل يمنع دخول أي شخص تم إزالته منها مع ضرورة تقييد عملية الدخول إلى تلك القائمة بأشخاص الإدارة المصرح لهم بذلك فقط.

3- برمجة النظام بشكل يمنع أي عملية من التمام دون مرورها خلال جميع الإجراءات السابقة.

4- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-2-1: سياسات وإجراءات فصل النظام لتوثيق عمليات النظام الخاصة الخدمة

والمقصود بفصل النظام جعل النظام الخاص بالخدمة التي يتم تداولها عبر مخدم الادارة العامة نظاما مستقلا؛ ولكن بشكل مؤقت يتم ربطه بنظام الادارة العامة الرئيسي في نهاية كل يوم تعامل. والهدف من ذلك السيطرة على العمليات بشكل أفضل وتفادي أي مشاكل تحدث ومنعها من الوصول لنظام الادارة العامة الرئيسي. ويمكن تحقيق ذلك بواسطة التالي:

1- برمجة نظام فرعي مصغر عن النظام المالى العام بالادارة العامة تناط به مهمة معالجة تعاملات البرامج المالية.

2- يصمم هذا النظام بشكل يضم جميع البرامج المالية الموجودة في نظام الادارة العامة الرئيسي.

3- فحص إتمام العمليات في النظام الفرعي المصغر في فترات دورية لا تتعدى ثلاث إلى أربع ساعات وطباعتها.

4- يتم ربط النظام الفرعي المصغر في نهاية اليوم وبعد التأكد من سلامة عملياته ، وطباعتها بنظام الادارة العامة الرئيسي ، وتفريغ جميع الحسابات الفرعية المؤقتة منه إلى الحسابات الرئيسية في النظام الرئيسي.

5- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الشركة الإلكتروني.

6-4-2-2: سياسات وإجراءات التوثيق لتتبع العمليات الخاصة بالخدمة

1- برمجة آلية تسجيل العمليات بناءً على اقتفاء أثر المتعامل (ادارة/فرع).

2- تخزين أو طباعة جميع العمليات التي تتم بشكل دوري.

3- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-2-3: سياسات وإجراءات رقابية لتوثيق عمليات السلع غير الملموسة

تعتبر السياسات والإجراءات الرقابية الخاصة بتوثيق عمليات الخدمة من السياسات والإجراءات الضرورية والهامة ؛ وذلك كي تتمكن الادارة العامة من متابعة أداء النظام وتقييمه بشكل منطقي وفاعل ، ويمكن أن تتكون تلك السياسات والإجراءات من التالي:

1- وضع آلية لتوثيق عمليات الخدمة بالشكل التالي:

- برمجة طلب الخدمة للادارة العامة بشكل يحتم على (الادرة/الفرع) تعبئة كل مما يلي:
  - اسم (الادراة/الفرع/المستخدم) من أربعة مقاطع.
  - المنطقة(ادارة/فرع)..
- يجب أن يبرمج نظام الادارة وقبل موافقته على قبول العملية وإصدار أمر تحميل بها إلى جهاز (الادارة/الفرع) أو إصدار تصريح بالدخول إلى الخدمة بالتحري الفوري على أن كلا من المعلومات صحيحة وفاعلة .

2- جعل عملية التسجيل في الحسابات الخاصة مبنية وبشكل كامل على الآليات السابقة وعدم تنفيذ أي عملية استثنائية وتحت أي ظروف خاصة.

3- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-3-1: سياسات وإجراءات توكيد إخطار المتعامل مع الخدمة

1- يتم إخطار (الادارة/الفرع) بتمام العملية أو عدم تمامها مع توضيح الأسباب.

2- جعل آلية الإبلاغ آلية مؤتمتة الرد.

3- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-3-2: سياسات وإجراءات تأكيد المتعامل (الادارة/الفرع) باستلامه لتعليمات الادارة العامة الخاصة بنتائج طلب الخدمة

1. لا بد من وصول تأكيد من طالب الخدمة باستلامه للرسالة الإلكترونية الخاصة بنتائج طلب الخدمة .
2. جعل آلية الإبلاغ آلية مؤتمتة الرد.
3. الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-3-4: سياسات وإجراءات رقابية لتأكيد عملية إخطار المتعامل ووصول التعليمات الخاصة بالخدمة إليه

تعد السياسات والإجراءات الرقابية الخاصة بتأكيد عملية إخطار المتعامل ووصول التعليمات الخاصة بالخدمة إليه من السياسات والإجراءات الضرورية والهامة ، وذلك كي تتمكن الادارة العامة من متابعة أداء النظام وتقييمه بشكل منطقي وفاعل ، ويمكن أن تتكون تلك السياسات والإجراءات من التالي:

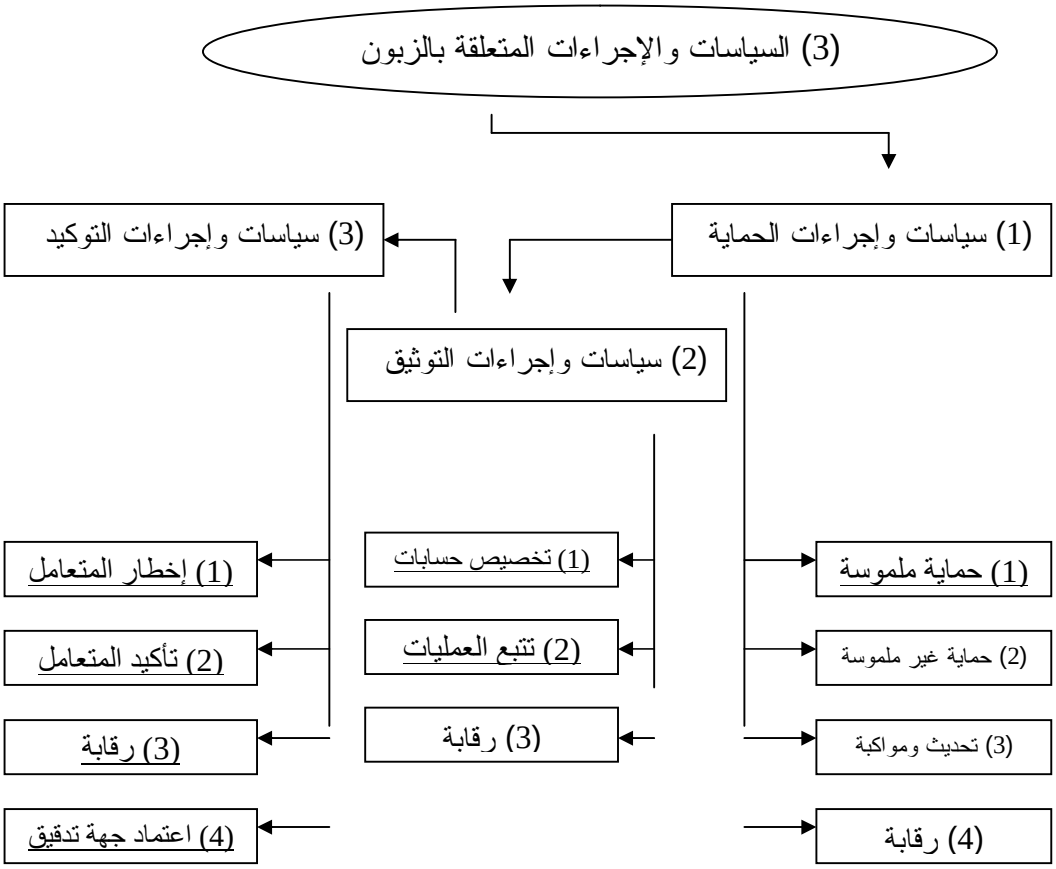
1- برمجة آلية ظهور الرسالة التلقائية الحاوية للتعليمات بعد إتمام الطلب بشكل تلقائي وفوري.
2- برمجة آلية بعث الإخطار إلى (الادارة/الفرع) بشكل رقمي وذلك بجعل النظام وعند بعثه رسالة الإخطار ترقيمها بترقيم إلكتروني تسلسلي بحيث وعند ارتداد الإجابة بوصول التعليمات تتطابق الأرقام الإلكترونية مما يسهل عملية تتبع العملية.
3- برمجة النظام بشكل يمنع أي عملية من التمام دون مرورها خلال جميع الإجراءات السابقة.
4- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الشركة الإلكتروني.

6-4-3-4: اعتماد جهة تدقيق تؤكد على جميع السياسات والإجراءات السابقة

يجب أن تتضمن سياسات الادارة العامة تعيين واعتماد جهة تدقيق خارجية مؤهلة إلكترونيا تستطيع تأكيد التزام إدارة الادارة العامة بتطبيق جميع سياساتها الخاصة بالبرمجيات المالية. والإفصاح عن هذه الجهة وإبراز توقيعها الإلكتروني في موقع الادارة العامة الإلكتروني.

6-4-2: دورة السياسات والإجراءات المتعلقة بالادارة /الفرع المتعامل مع الادارة العامة

لكي نتمكن من تتبع دورة السياسات والإجراءات المتعلقة بالادارة/الفرع المتعامل مع الادارة العامة لا بد من التقيد بالدورة الموضحة بالشكل رقم (5-4) ، وبالشكل التالي:



الشكل (5-4):دورة السياسات والإجراءات المتعلقة بالادارة/الفرع المتعامل مع الادارة العامة

قبل البدء بشرح الدورة لا بد من التعرف أولا على المقصود بكلمة الادارة/الفرع ، يعرف الباحث الادارة/الفرع "بالمتعامل مع الادارة العامة من الوحدات ذات المستويات المختلفة".

6-4-2-1: سياسات وإجراءات الحماية الملموسة للزبون

بما أن تعامل الادارة/الفرع مع الادارة العامة يقتصر على التعامل المالى فان الادارة العامة ملزمة بوضع سياسات وإجراءات ملموسة تساهم بحماية بيانات الادارة/الفرع وتوفير الخصوصية اللازمة له ؛ وذلك في سبيل نيل ثقته بسرية وخصوصية نظام الادارة المالية بشكل عام. ومن أهم البيانات التي يمكن تصنيفها ببياناتالادارة/الفرع الخاصة والسرية التالي:

- أسم وعنوان الادارة/الفرع.
- رقم الوحيد المميز للادارة/الفرع.
- رقم المستخدم من الادارة /الفرع .
- أي بيانات أخرى يحددها الادارة/الفرع بأنها خاصة وسرية.

ورغم أن وضع سياسات وإجراءات حماية ملموسة لبيانات ذات طبيعة غير ملموسة أمر بالغ الصعوبة إلا أن الباحث يرى أن الادارة العامة ملزمة بتوفير التالي:

1- تعيين قسم فني مختص بتطوير وتحديث سياسات وإجراءات الحماية.
2- توفير أنظمة وبرامج حفظ وحماية خاصة لبيانات الادارات/الافرع الخاصة.
3- توثيق بند تعهد في عقود تشغيل الموظفين المناط بهم مسؤولية التعامل مع بيانات الادارات/الافرع مفاده الالتزام بخصوصية وسرية البيانات الخاصة بهم.
4- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الشركة الإلكتروني.

6-4-2-2: سياسات وإجراءات الحماية غير الملموسة للادارة/الفرع

يقصد بالحماية غير الملموسة جميع السياسات والإجراءات الواجب على الادارة العامة توفيرها لحماية البرمجيات الخاصة بالعمليات التي تتم مع الادارة/الفرع ، والتي يمكن تلخيصها بالشكل التالي:

1- تحديد الأشخاص المصرح لهم بالتعامل والإطلاع على بيانات الادارة/الفرع وتوثيق أسمائهم بلائحة إلكترونية خاصة لدى إدارة الادارة العامة.
2- برمجة عملية الدخول لنظام بيانات الادارة/الفرع وتقييدها بأرقام سرية مكونة من عدة خانات بحيث لا تقل عن ثماني خانات ، وجعل جزء من الخانات رقمية والأخرى حرفية ، بحيث يكون من الصعب على أي شخص يريد اختراق النظام اختراقه نظرا لتعدد الاحتمالات.
3- برمجة عملية الدخول إلى النظام بالأرقام السرية بشكل يجعلها عملية ذات محاولات محدودة ، بحيث يتوقف النظام للاستجابة إن حاول أحدهم الدخول بأرقام سرية خاطئة لأكثر من محاولتين أو ثلاث محاولات.

4- برمجة آلية عملية الدخول إلى النظام بالأرقام السرية بشكل يمكن المستخدم المصرح له بتغيير الرقم السري الخاص به كلما دعت الحاجة إلى ذلك خوفا من تسربه للآخرين.

5- تزويد النظام بآلية مبرمجة تمكن الإدارة من معرفة أوقات دخول النظام من قبل من ؛ وذلك لتسهيل عملية إناطة المسؤوليات.

6- تزويد النظام ببرنامج حماية ضد الفيروسات والبرامج الدخيلة ، وبرمجة هذا النظام بشكل يجعله حذرا من الأمور غير المعروفة له ، فأحيانا تكون الفيروسات حديثة جدا لدرجة أن نظام الحماية لا يتعرف عليها ، وفي هذه الحالة يجب أن يبرمج نظام الحماية ضد الفيروسات بعدم السماح للعملية بالتمام إن كان هناك شك فيها.

7- برمجة آلية إبلاغ عن أي اختراق أو محاولة اختراق تتم وربطها مع القسم المختص بشكل مباشر.

8- حفظ جميع البيانات الخاصة الادارت/الافرع على عدة نسخ من الأقراص الإلكترونية ، وبشكل دوري ويفضل أن تتم عملية الحفظ في نهاية كل يوم عمل خوفا من حدوث أي طارئ.

9- طباعة جميع بيانات الادارة/الافرع في نهاية كل يوم عمل ، والاحتفاظ بها في ملفات خاصة لتلك الغاية.

10- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الشركة الإلكتروني.

6-4-2-3: سياسات وإجراءات تحديث ومواكبة التغيرات التكنولوجية الخاصة بنظم حماية الادارة/الافرع

التقيد التام بجميع السياسات والإجراءات المتبعة في الخدماتوكما جاء بالفقرة رقم (1-3-1).

6-4-2-4: سياسات وإجراءات رقابية لحماية الادارة/الفرع

تعتبر السياسات والإجراءات الرقابية الخاصة بحماية الادارة/الفرع من السياسات والإجراءات الضرورية والهامة ، وذلك كي تتمكن الادارة العامة من متابعة أداء النظام وتقييمه بشكل منطقي وفاعل ، ويمكن أن تتكون تلك السياسات والإجراءات من التالي:

1- وضع آلية لتسلسل عمليات الخدمة بالشكل التالي:

- يقوم النظام بتنفيذ العملية بعد أن يتأكد من صحة جميع المعلومات الخاصة بالادارة.
- بعد أن ينفذ النظام العملية يقوم بإصدار أمر إلكتروني باتمام التنفيذ.
- تتابع بقية الإجراءات وفقا للفقرة رقم (1-1-4) الخاصة بحماية الخدمة .

2- مراجعة اللائحة الإلكترونية التي تحوي أسماء الأشخاص المصرح لهم بالدخول إلى النظام بشكل دوري وبرمجتها بشكل يمنع دخول أي شخص تم استبعاده منها مع ضرورة تقييد عملية الدخول إلى تلك القائمة بأشخاص الإدارة المصرح لهم بذلك فقط.

3- برمجة النظام بشكل يمنع أي عملية من التمام دون مرورها خلال جميع الإجراءات السابقة.

4- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الشركة الإلكتروني.

6-4-2-5: سياسات وإجراءات فصل النظام لتوثيق عمليات النظام الخاصة بالادارة/الفرع.

والمقصود بفصل النظام جعل النظام المحاسبي الخاص بالتعاملات النقدية مع الادارة/الفرع عبر الادارة العامة نظاما مستقلا ولكن بشكل مؤقت يتم ربطه بنظام الشركة الرئيسي في نهاية كل يوم تعامل ، والهدف من ذلك السيطرة على العمليات بشكل أفضل وتفادي أي مشاكل تحدث ومنعها من الوصول لنظام الادارة العامة الرئيسي ويمكن تحقيق ذلك بواسطة التالي:

1- برمجة نظام فرعي مصغر عن النظام المحاسبي العام بالادارة العامة تناط به مهمة معالجة التعاملات المالية الخاصة للادارة/الافرع.

2- فحص تمام العمليات في النظام الفرعي المصغر بفترات دورية لا تتعدى ثلاث إلى أربع ساعات وطباعتها.

3- يتم ربط النظام الفرعي المصغر في نهاية اليوم وبعد التأكد من سلامة عملياته وطباعتها بنظام الادارة العامة الرئيسي وتفريغ جميع الحسابات الفرعية المؤقتة منه إلى الحسابات الرئيسية في النظام الرئيسي.

4- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الشركة الإلكتروني.

6-4-2-6: سياسات وإجراءات التوثيق لتتبع العمليات الخاصة بالادارة/الفرع

التقيد التام بجميع السياسات والإجراءات المتبعة في الخدمة وكما جاء بالفقرة رقم (1-2-3).

6-4-2-7: سياسات وإجراءات رقابية لتوثيق عمليات لادارة/الفرع

تعتبر السياسات والإجراءات الرقابية الخاصة بتوثيق عمليات الادارة/الافرع من السياسات والإجراءات الضرورية والهامة ؛ وذلك كي تتمكن الادارة العامة من متابعة أداء النظام وتقييمه بشكل منطقي وفاعل ، ويمكن أن تتكون تلك السياسات والإجراءات من التالي:

1- وضع آلية لتوثيق عمليات النظام .
2- جعل عملية التسجيل في الحسابات الخاصة مبنية وبشكل كامل على الآليات السابقة، وعدم تنفيذ أي عملية استثنائية وتحت أي ظروف خاصة.
3- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الشركة الإلكتروني.

6-4-2-8: سياسات وإجراءات توكيد إخطار الادارة/الفرع

يتم إخطار الادارة/الفرع بتمام العملية أو عدم تمامها مع توضيح الأسباب والإجراءات في حالة تمام العملية بطريقتين وبالشكل التالي:

1- بأن تظهر رسالة إلكترونية على لدى الادارة العامة عبر الشبكة عند طلب الخدمة .
2- بأن تبعث رسالة إلكترونية من قبل نظام الادارة العامة مباشرة إلى الادارة/الفرع بقبول الطلب بعد الفحص.
3- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-2-9: سياسات وإجراءات تأكيد الادارة/الفرع باستلامه لتعليمات الادارة العامةالخاصة بنتائج طلب الخدمة

لا بد من وصول تأكيد من الادارة/الفرع طالب الخدمة باستلامه للرسالة الإلكترونية الخاصة بنتائج طلب الخدمة وتعليمات الاستلام ، وتستطيع الادارة العامة التأكد من أن الادارة/الفرع طالب الخدمة قد استلم التعليمات .

1- جعل آلية الإبلاغ آلية مؤتمتة الرد.

2- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-2-10: سياسات وإجراءات رقابية لتأكيد عملية إخطار الادارة/الفرع ووصول التعليمات الخاصة بالخدمة إليه

تعتبر السياسات والإجراءات الرقابية الخاصة بتأكيد عملية إخطار الادارة/الفرع ووصول التعليمات الخاصة بالخدمة إليه من السياسات والإجراءات الضرورية والهامة ، وذلك كي تتمكن الادارة العامة من متابعة أداء النظام وتقييمه بشكل منطقي وفاعل ، ويمكن أن تتكون تلك السياسات والإجراءات من التالي:

1- برمجة آلية ظهور الرسالة التلقائية الحاوية للتعليمات بعد إتمام العملية بشكل تلقائي وفوري.

2- برمجة آلية بعث الإخطار إلىالادارة/الفرع بشكل رقمي وذلك بجعل النظام وعند بعثه رسالة الإخطار يقوم بترقيمها برقم إلكتروني تسلسلي بحيث وعند ارتداد الإجابة بوصول التعليمات تتطابق الأرقام الإلكترونية مما يسهل عملية تتبع العملية.

3- برمجة النظام بشكل يمنع أي عملية من التمام دون مرورها خلال جميع الإجراءات السابقة.

4- الإفصاح عن جميع السياسات والإجراءات السابقة في موقع الادارة العامة الإلكتروني.

6-4-2-11: اعتماد جهة تدقيق تؤكد على جميع السياسات والإجراءات السابقة

يجب أن تتضمن سياسات الادارة العامة تعيين واعتماد جهة تدقيق خارجية مؤهلة إلكترونيا تستطيع تأكيد التزام إدارة الادارة العامة بتطبيق جميع سياساتها للادرة/الفرع. والإفصاح عن هذه الجهة وإبراز توقيعها الإلكتروني في موقع الادارة العامة الإلكتروني.

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الباب السابع

النتائج والمناقشة

7-1 مقدمة

يلخص هذا الفصل نتائج الأبحاث في ثلاثة اقسام في عمليات البرمجيات واعادة هندسة البرمجيات ونموذج ربط البرمجيات باتباع المعايير المالية لتحسين الممارسات و التطوير .
ومعظم هذه النتائج موجودة في الاوراق وفى داخل هذا البحث' ولكن يقدم هذا الفصل بعض البيانات التي لم يتم نشره ويناقش تاثيره علي النتائج .

7-2 اعادة هندسة البرمجيات

ورقة بعنوان صيانة البرمجيات مقابل أعادة هندسة البرمجيات

وتصف هذه الورقة مصطلح الصيانة واعادة الهندسة بشكل وثيق مع بعضها البعض. ظهر هذين المصطلحين من عالم الكئنات اللملموسة. الآن دخلت عالم البرمجيات وهي مناسبة تماما لأنظمة البرمجيات. فمن الصعب استخلاص خط واضح بين هذين المصطلحين. فى كثير من الأوقات تستخدم بالتبادل. إعادة هندسة البرمجيات النظم هو موضوع ذو أهمية في الفترة المقبلة وسيكتسب المزيد من الاهتمام في عالم برمجيات النظم.كثيرا ما يجرى الخلط بين الصيانة وإعادة الهندسة من مديرى البرمجيات .لذلك يجب فصل هذين المصطلحين لانه يسبب مشكلة للبعض. فتم محاولة وضعهم في مناطق مختلفة غير متداخلة.
الصيانة وإعادة الهندسة هما مجالات مختلفة في هندسة البرمجيات. الصيانة لتشغيل النظام حتى سن معينة و إعادة الهندسة جعل النظام الجديد يعمل لفترة حياة أخرى. نطاق إعادة الهندسة هو واسع وصعب بالمقارنة مع الصيانة. إعادة الهندسة هى تقليل النفقات على أنظمة البرمجيات في المنظمات. إعادة الهندسة لديها مجالا أكبر في عالم البرمجيات مقارنة بالكاتنات اللملموسة فى العالم الحقيقى .
فى هذه الورقة تم عرض ثلاثة نماذج للتميز بين الصيانة واعادة الهندسة وهى :

1. نموذج الطريق (Thoroughfare model).
2. النموذج المبنى على التكلفة(Cost based Model) .
3. النموذج المبنى على الكائنات (Object based Model) .

هذه النماذج هى مجدية لمديرى البرمجيات لاعادة الهندسة فى الوقت المناسب,اعادة الهندسة غير مجدية قبل وبعد المرحلة الانتقالية.

سوف تساعد هذه النماذج لاعادة هندسة البرمجيات والهروب من عب شراء البرمجيات الجديدة المكلفة . وسوف يتم الاستفادة من البرامج القديمة الموجودة .

مناقشة Discussion :

بينن واخرون يعرف نظام الارث بأنه " البرامج التى تكون حيوية لدي المنظمة ولكننا لانعرف مايجب القيام به " واعادة الهندسة هي عالية التكلفة, ونشطاتهاعالية المخاطر مع الفوائد التجارية غير الواضحة [2] احياناً تكون اعادة هندسة للرمز وقد تكون ايضاً للبيانات الهامة للمنظمات للتكامل مع انظمة التكنلوجيا الجديده.

بينيت واخرون كتب أنه يبدو واضحاً أن وجود مستوي عالي من التصميم المعمارى مع التتبع المستوي العالي والادني للتصميم تساعد الصيانة . وهنالك القليل من الادلة التجريبية الذي يساعد هذا الواقع لموظفي الصيانة

الا من خلال الممارسة ، ويبقى التتبع فقط من خلال شفرة المصدر ولكنها تبقى غير متناسقة مع المتطلبات الحالية للنظام ولايمكن الوثوق بها لفترة طويلة توجد هذه المشكلة حتي قبل مرحلة الصيانة . التناقض بين الرمز و المتطلبات ليست فقط مشكلة الصيانة ولكن ايضا تفقد معمارية النظام ولكن المهم أيضاً خلال التطوير ولاسيما التزايدى يبين كل اصدار علي الاصدار السابق وذلك بالتوثيق المستمر .

7-3 نماذج تطوير البرمجيات

ورقة بعنوان مسح على نماذج العمليات المختلفة المستخدمة فى تطوير البرمجيات

وتصف الورقة نماذج العمليات المختلفه من نموذج الشلال وتدرج هذه النماذج حتى الوصول للنماذج التى ترتبط بالمنحنى الكائنى ومستخلص هذه الورقة هو:

خبراء ومطوري النظم والعملاء انهم يسعون لهدف مشترك وهو بناء انظمة المعلومات التى تدعم بشكل فعال اهدافها لكى يضمن انه فعالة من حيث التكلفة والوصول للجودة التى تلبى متطلبات المنظمة لذلك يتم توظيف نوع من انواع عمليات التطوير النموذجيه لتوجيه دورة حياة التطوير ونجد ان هنالك عدد كبير من نماذج تطوير البرمجيات وكان الغرض من الورقة هو اجراء مسح على النماذج المختلفه المستخدمه فى مجال تطوير البرمجيات

مناقشة :

كانت الدراسة استطلاعية وكانت المسوحات علي نطاق صغير النتائج تستخدم لتوليد الفرضيات للدراسات المستقبلية وكانت اكثر نماذج التطوير استخداما هى العملية الموحدة من راشونال RUP.

لكن لا توجد مناشط واضحة لاعادة هندسة البرمجيات فى هذه المنهجيه .

7-4 المنهجية المقترحة لاعادة هندسة البرمجيات للبرمجيات غيرر كائنية التوجه باتباع المنهج الكائنى :

المنهجية المطبقة في هذا البحث تبدا بتقييم النظام الحالى وذلك باستخدام طريقة بسيطة للتقييم وهى واضحه في الشكل (4-1) في القسم 4-1 يتم عرض اساليب تقييم النظام القديم ثم تحليل المتطلبات يركزعلي كيف يتم التقاط المتطلبات وتحليل النظام وتحديد الممثلين والسيناريوهات ثم يقدم تحديد الكائنات من المتطلبات التى تم التقاطها ثم التقاط اهداف التصميم من خلال المتطلبات والتطبيق من خلال تحديد الكائنات وتنظيمها التى تم تعريفها

مناقشة Discussion :

المنهجية قدمت طريقة سهلة لتقييم النظام وركزنا فيه على المتطلبات الغير وظيفيه لان المجال للبرمجيات الماليه يتطلب ذلك ثم استخلاص وتحليل المتطلبات وتوفير التوثيق المناسب للنظام من خلال نمذجة المتطلبات باتباع طريقة المنحنى الكائنى واعادة الاستخدام التدريجيه لاعادة هندسة النظام القائم

7-5 مشروع تطوير نظام الربط بين الادارة العامة للشئون المالية والافرع والادرات

قام الباحث بتعديل نموذج الربط الذى قدمه ظاهر شاهر يوسف القشي باشراف الأستاذ الدكتور نعيم دهمش فى رسالة دكتورا باسم مدى فاعلية نظم المعلومات المحاسبية في تحقيق الأمان والتوكيدية والموثوقية في ظل التجارة الإلكترونية

استفاد الباحث من الدراسه اعلاه باضافة بعض التعديلات عليه لإيجاد حلول مناسبة للمشاكل لربط البرامج المالية الوصول إلى تطوير مشروع نظام يربط بين نظام الادارة العامة للشئون المالية وموقعها المختلفة فى ولايات السودان.

وايضا لمعرفة امكانية تطبيق الطريقه التى قدمها ظاهر شاهر يوسف القشي واختباره فعليا

مناقشة Discussion :

انطلاقا من الأمانة العلمية الواجب توفرها في كل بحث ودراسة علمية، فإن الباحث يرى بأن مشروعه المقترح يمكن تطبيقه بالادارة العامة، وذلك بسبب :

1- هنالك الكثير من المعايير موجودة ضمنا فى البرامج التى تم تطويره

2- ربط شبكة القوات المسلحة لكل الولايات تم جزء كبير منه الان.

ولكن بالرغم من ذلك هنالك بعض المعوقات وهى:

1. تخوفها من العمل عبر الشبكات الواسعة النابع من المخاطر المرافقة لها.
2. عدم وجود جهات تدقيق مؤهلة لتدقيق عمليات المالية الالكترونية عبر الشبكة.

بالرغم من المعوقات المذكورة أعلاه، إلا أن الباحث يعتقد بأن انضمام الادارة المالية للمحافل الدولية الاقتصادية بشتى أشكالها سيحتم عليه الخوض بهذه التقنية عاجـــلا أم آجلا. وفي الآونة الأخيرة، ونتيجة السياسات الاقتصادية الموجهة نحو الخصخصة ظهر العديد من الشركات الكبيرة التي تستطيع استخدام تقنية الحديثه وبالتالي فان النظام المقترح من قبل الباحث سيجد موقعه المناسب في الوقت المناسب.

6-7 دراسة الحالة

فى دراسة الحالة تم تطبيق المنهجيه المقترحه لاعادة هندسة النظام القديم للمرتبات فتم تصميم نظام جديد يلبى متطلبات المستخدمين وان النظام الجديد يعمل بصورة جيدة حسب ردود افعال كل من المحاسبين والمراجعين والتقنيين بالادارة العامة للشئون المالية وادى ذلك بالنظرة المتفائلة لعمل النظام من حيث الشكل والوظائف وتخطيط واجهة المستخدم ولازال هذا النظام تحت التطوير وتم توزيعه على كل من القوات البرية والبحرية والجوية والمشتركة .

7-7 **الخلاصة**:

النتائج وعلاقتها بأسئلة البحث في الاقسام 7.4 – 7.1 وتغطي الدراسات عدة جوانب لتطوير البرمجيات بسبب البحوث الناشئة ونوع البيانات المتاحة وحقيقة أن هنالك من مناهج التطوير في السياق الحقيقي التي ينبغي دراستها ككل وقد بذلت محاولات لاستخدام كل البيانات المتاحة ولكن ليس الي لافراط ويكون علي بينة من القيود المفروضة علي نتائج المناقشة. استخدام اعادة هندسة البرمجياتله فوائد اما تتحقق منه هنا او في دراسات أخري هنالك تحديات ايضاً في التكيف لمطلبات البرمجيات التي يجب الرد عليها . التطوير واسع النطاق فمن المهم التحقق من أن هذه الطريقة تقدم حل.

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الباب الثامن

الخلاصة والتوصيات والدراسات المستقبليه

8-1 الخلاصة

في هذا البحث كانت المنهجية المقترحه لاعادة هندسة البرمجيات القديمه القائمةعلى تقنيات المنحي الكائنى وتطبيقها لاعادة هندسة مجموعة من البرامج المالية (نظام مرتبات الضباط) بالادارة العامة لادارة الشئون الماليه والتقييم على اساس النظم القائمة ويقترح المنهجيات وتطبيقها لتحسين تنظيم الكود الاساسي وتشمل منهجيات الهندسة الكائنية المنحني ثلاثة اجزاء رئيسية نظام التقييم والتحليل والتصميم الكائني ويركز تقيم النظام على اعادة تحليل واعادة جميع المتطلبات غير الوظيفية في الدراسة . ويركز التحليل الكائني اساسا على حالات الاستخدام ونموذج التعاون والتصميم والتخصيص الشامل وحالات الاستخدام المتوفرة . لتحديد الكائنات الاساسية والخصائص. والتعاون يساعد فى معرفة سلوكيات الكائنات واستخدام التصميم والتخصيص في المنحني الكائني للتعرف على الكائنات التى يعاد استخدامها يتم تخطي جميع السيناريوهات في دراسة الحالة لانه يمكن للنظام القائم على توفير المعلومات الكافية عن النظام التصميم الغرضي التوجه هوتحويل نموذج التحليل الي حلول ومعلومات قابلة للتنفيذ ويتم تحديد اهداف التصميم اولاً من خلال مراجعة المتطلبات الغير وظيفية واختيار بعض نماذج التحليل الغرضي التوجه لتحديد اهداف التصميم يتم تعريف كائنات إضافية لسد الفجوة بين اهداف التصميم والكائنات واعادة هندسة النظام القديم القائم على المنحنى الكائنى ايضا تغطى الانظمة التى اساسها لا يقوم على المنحنى الكائنى.

وقد اجريت دراسة الحالة لاعادة هندسة النظام المالى للادارة العامة لادارة الشئون الماليه على اساس مقترح المنحني الكائني وتجميع المنهجيات . هنالك عبر التحسينات التى اجريت على النظام القديم تحققت من النظام الجديد من حيث سهولة الاستخدام والمرونة والتمديد .

نرى ان النتيجة تشير الي انه المنهجيه المقترحه تقدم حل بشكل جيد ولكن من الصعب تعميمها فى لكل المجالات والظروف فالمجال والخبرات تلعب دورا مهما فى اختيار الاستراتيجيه المناسبه لاعادة الهندسة .

قام الباحث بتعديل نموذج الربط الذى قدمه ظاهر شاهر يوسف القشي باشراف الأستاذ الدكتور نعيم دهمش فى رسالة دكتورا باسم مدى فاعلية نظم المعلومات المحاسبية في تحقيــق الأمـــان والتوكيدية والموثوقية في ظل التجارة الإلكترونية

استفاد الباحث من الدراسه اعلاه باضافة بعض التعديلات عليه لإيجاد حلول مناسبة للمشاكل لربط البرامج المالية الوصول إلى تطوير مشروع نظام يربط بين نظام الادارة العامة للشئون المالية وموقعها المختلفة فى ولايات السودان.

8-2 التوصيات

كيف يمكنك تطبيق المنهجيه المقترحه لاعادة هندسة النظام القديم

لنري اولاً : من خلال فهم ماتحاول القيام به.

ثانياً: من خلال استغلال ما لديك بالفعل بذكاء.

ثالثاً: من خلال التركيز علي المبادي وليس بالضرورة التفصيل

واحدة من المحاذير

لقد راينا كيف تفشل المشاريع عندما يتم محاولة التغير في نفس الوقت الشي الرئيسي في تطور النظام القديم "علي سبيل المثال الهجرة الي منصات جديدة " في نفس الوقت التي تغير في العملية " فمن الأفضل الادخال الجديد يمكن أن تصبح مالوفة مع المنهجية فلسفاتها ومحتوياتها وكذلك الادوات التي تدعم ذلك تجنب مضاعفة خطر هذا المشروع من خلال تقيم الكثير من المجاهيل والتغيرات في وقت واحد.

8-3 الدراسات المستقبلية :

1. التقنيات الغرضية التوجه دائما تركز على بنا انظمة قابلة للاستخدام ومرنه وقابلة للتمديد وسهلة الصيانة ومع ذلك المرونة العالية تعنى مزيدا من التعقيد والتعقيد العالى يعتبر تحديا فى كل مكن التنفيذ والصيانة وهذا بدوره يحتاج الى مزيدا من الجهد والوقت لذلك فمن المهم قياس موارد المشروع المتاحه التى يمكن استخدامها فى فى التطوير والصيانة ووضعه فى التخطيط لوضع خطط التنفيذ التى تحقق مستوى مناسب من المرونة لتحقيق الجودة حسب الموارد المتاحه .
2. استكشاف حل بديل لتحليل النظام واعادة هندسة البرمجيات.
3. مقارنة المنهجية المقترحه مع نماذج التطوير المختلفه من حيث الكلفة والتعامل مع المخاطر وفق معايير محددة .

4. التعديل فى منهجية RUP لاستخدامها فى اعادة هندسة البرمجيات وذلك باضافة المناشط المناسبة لذلك .
5. اختبار المنهجية المقترحة بعدد مختلف من البرمجيات فى مجالات مختلفه حتى يتم اعتمادها او تحسينه او رفضه .
6. دراسة طرق اخرى للربط مبنية على الامكانيات المتاحه من المواررد فى الاجهزة والمعدات والشبكات

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

المســـــــتخلص

أذداد الاهتمام بالانظمة المالية بصورة كبيرة لذلك زاد الاهتمام بتطوير هذه الانظمة الى بيئة اكثر مرونه تسمح بردود فعل سريع لاحتياجات العملاء .

ولكن نجد أن هنالك انظمة قديمة موروثة تم تطويرها منذ اكثر من 10 اعوام , ولكنه لاتزال تعمل وموثره فى بيئتها وتقدم الخدمات .المشكلة الرئيسية فى الانظمة الموروثة هو ان هنالك فجوة فى التكنلوجيا بينها وبين النظم المحيطه بها وتقاوم التعديل والتطوير وغالبا الوثائق غير مكتملة وان وجدت تكون خارج نطاق التاريخ الحالى، مما يودى الى صعوبة وتكلفة عالية فى الصيانة .

فنجد أن اعادة هندسة البرمجيات هى التى تقوم بتحسين جودة هذه الانظمة من خلال تقديم تصميم جديد .

لذلك فى هذا البحث تم استكشاف المنهجيات الغرضية التوجه واقتراح منهجية لاعادة هندسة البرمجيات القائمة على غرضية التوجه واقتراح نموذج ربط .

بما أن لغة النمذجة الموحدة (UML) تعتبر لغة قياسية توفر رسوم بيانية لوصف وتوثيق النظام تم استخدامها فى توثيق النظام .

كما تم اقتراح نموذج ربط للربط بين الادارة العامة للشئون المالية وادارتها ووحداتها المختلفة .

كذلك تم تطبيق منهجية اعادة هنسة البرمجيات القائمة على غرضية التوجه المقترحة فى اعادة هندسة برمجيات انظمة الادارة العامة للشئيون المالية بوزارة الدفاع .

وتشير النتائج الى ان تطبيق منهجية أعادة الهندسة المقترح ادى الى المساعدة على تحسين مرونة النظام وسهولة الاستخدام والصيانة والتوسع ، ونموذج الربط أعطى موثوقية ومرونة .

وتشير النتائج على أن النظام الموجود كان به انظمة فرعية ولكنه غير مكتملة والهياكل غير معرفة بطريقة صحيحة .

كذلك تمت اضافة امكانيات جديدة فى النظام الجديد باستخدام قواعد بيانات ORACLE11G و developer6i .

Abstract

The Financial system interest increased significantly so increased interest in the development of these systems to a more flexible environment that allows quick reaction responses to customer needs.

But we find that there is a Legacy system has been developed for more than 10 years, but it still works and effect of in their environment and providing services. The main problems of legacy systems are the enlarging technology gaps between them and their surrounding systems and their resistance to modification and evolution. And Documentation is usually incomplete and out-of-date. Leading to difficulty and high cost of maintenance.

Re-engineering a legacy system mainly focuses on improving quality and maintainability of the existing system by introducing new designs.

In this research was to explore object-oriented methodologies and propose a methodology for re-engineering the software-based object-oriented model and propose a link.

The Unified Modeling Language (UML) is a standard language provides graphs to describe and document the system used in the system documentation.

It was also suggested linking model to link the financial systems between the units and the various departments and branches of public financial management.

In the case study, the proposed methodologies are successfully applied to Re-engineer the legacy system. Results indicate that objected-oriented Methodologies help improve system flexibility, usability, maintainability and extendibility.

Also results indicate that the existing system was its sub-systems, but it is incomplete and the structures are not defined properly.

Also it has added new possibilities in the new system using ORACLE11G databases and developer6i.

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

المحتويات Contents

الموضوع	الصفحة
الآية	أ
الإهداء	ب
شكر وعرفان	ج
المستخلص	د
Abstract	هـ
قائمة المحتويات	و
قائمة الجداول	ل
قائمة الاشكال	ن
الباب الاول: مقدمة	
1-1 مقدمـــــة	2
1-2 مشكلة البحث	4
1-3 اهداف البحث	6
1-4 محددات البحث	7
1-5 أهمية البحث	7
1-6 حدود البحث	7
1-7 الدراسات السابقة	8
1-8 التعليق على الدراسات السابقة	13
1-9 منهجية البحث	14
1-10 هيكل البحث	14
الباب الثانى: تطوير البرمحيات واعادة هندسة البرمجيات	
2-1 الانظمة القديمة الموروثة Legacy system	16
2-1-1 دورة حياة نموذج الشلال الاصلي water fall	16
2-1-2 تغير الاعمال Business change	17
2-1-3 أرتقاء البرمجيات software Evolution	17
2-2 هندسة البرمجيات للتطوير computer System Evolution	19
2-2-1 تطور الحاسوب computer System Evolution	19
2-2-2 خصائص نظم البرمجيات الكبيرة	20

2-2-3 استعراض حالات الممارسة في نموذج المتطلبات	21
2-3 هندسة البرمجيات software Engineering	21
2-4 جودة البرمجيات software Quality	25
2-5 البرمجيات في اي مكان Software everywhere	26
2-6 صيانة البرمجيات Software maintenance	28
2-6-1 تطوير البرمجيات	32
2-7 اعادة هندسة البرمجيات للتطوير Software Reengineering For Evaluation	33
2-7-1 دورة أعادة هندسة البرمجيات Software reengineering cycle	34
2-7-2 تصنيف اعادة هندسة البرمجيات Taxonomy of software reengineering	34
2-7-3 الهندسة العكسية Reverse engineering	39
2-7-4 الوضع الحالي للطرق الرسمية في اعادة الهندسة	41
2-8 التلخيص	42
الباب الثالث: نماذج عمليات البرمجيات	
3-1 المقدمة	44
3-2 عمليات البرمجيات Software Processes	44
3-3 نماذج عمليات البرمجيات	46
3-3-1 نموذج الشلال	46
3-3-2 :النموذج التكرارى	47
3-3-3 V-shaped Model	48
3-3-4: النموذج الحلزونى Spiral Model	50
3-3-5: Extreme Programming (XP) Model	51
3-4 العمل التكراري التزايدي Iterative, Incremental Framework	52
3-5 العملية الموحّدة من أي بي ام The Rational Unified Process	55
3-6 مفاهيم المنهج الغرضي التوجه في هندسة البرمجيات	58
3-6-1 التحليل الغرضى التوجه	59
3-6-2 التصميم الغرضى التوجه	60
3-6-3 الاختبارات الغرضى التوجه	61
الباب الرابع: المنهجية	
4-1 تقيم جودة النظام	65

2-4 استخراج المتطلبات Requirements Elicitation	66
3-4 تعريف الفاعلين Identifying Actors	67
4-4 تعريف السيناريو Identify Scenarios	68
1-4-4 تعريف حالات الاستخدام Identifying use case	69
2-4-4 النتقاط المتطلبات غير الوظيفية	71
2-4-4 سهولة الاستخدام usability	72
5-4 التحليل الغرضى التوجه OO Analysis	73
1-5-4 تحديد العناصرو الحدود والتحكم للكائنات	74
2-5-4 نموذج التعاونModeling Collaboration	76
3-5-4 التعميم والتخصيص Generalization and specialization	79
6-4 تصميم المنحني الكائنى(OO Design)	79
1-6-4 التقاط اهداف التصميم(Capture Design Goals)	80
2-6-4 معمارية النظام System Architecture	80
1-2-6-4 تصميم النظام الفرعي Subsystem Design	81
2-2-6-4 تصميم الكائن Object Design	82
3-2-6-4 انماط التصميم Design Patterns	82
3-6-4 سهولة الاستخدام	83
1-3-6-4 تصميم واجهة المستخدم	84
2-3-6-4 التصميم المتمركز على المستخدم	84
3-3-6-4 مبادى تصميم واجهات المستخدم	84
4-3-6-4 تفاعل المستخدم والنظام	84
5-3-6-4 دعم المستخدم	84
6-3-6-4 رسائل الخطا	85
1-3-6-4 توثيق المستخدم	85
7-4 الانظمة الموزعة	86
1-7-4 دوافع توزيع الانظمة	86
2-7-4 معمارية الخادم العميل	87
3-7-4 طبقات التطبيق	88
4-7-4 العميل النحيل والعميل البدين	89

4-7-5 معمارية الثلاثة طبقات	91
4-8 اعادة تنظيم النظام غير الكائني Non – OO System Reorganization	92
الباب الخامس: دراسة حالة	
5-1 دراسة حالة (case study)	96
5-2 الاهداف	96
5-3 مقدمة عن النظام	96
5-3-1 انظمة الادارة العامة للشئون المالية	97
5-3-1-1انظمة مرتبات الضباط	98
5-3-2 نظام مرتبات الضباط	98
5-3-2 -1 نظرة الي نموذج الوظائف	98
5-4 تقيم نظام المرتبات	102
5-4-1 تقييم المتطلبات الوظيفية	102
5-4-2 تقيمم المتطلبات الغير وظيفية	102
5-4-2-1 التكرار	103
5-4-2-2 مكافحة النمط	104
5-4-2-3 وظائف /واوجهة المستخدم للمنظمة	104
5-4-2-4 معالجة الاستثناءات	105
5-5 التقاط المتطلبات فى نظام مرتبات الضباط	105
5-5-1 الفاعلوب (اللاعب)	106
5-5-2 السيناريو	106
5-5-3 حالات الاستخدام	106
5-5-4 المتطلبات الغير وظيفية :Non-functional requirements	126
5-6 التحليل الكائني OO Analysis	126
5-6-1 تعريف كائنات الحدود ، كائنات التحكم كائنات العناصر كائنات الاجسام	127
5-6-2 نموذج التعاون modeling collaboration	128
5-6-3 التعميم والتخصيص	132
5-7 التصميم الكائني OO Design	135
5-7-1 اهداف تصميم النظام	135
5-7-2 اعادة تصميم الانظمة الفرعية	136

5-7-3 تصميم الكائنات	137
5-7-4 مخطط الأنشطه :Activity Diagram	138
5-7-5 مخطط التحزيم	138
5-7-6 أعادة تصميم واجهة المستخدم	139
5-7-6-1 دعم المستخدم	140
5-7-6-2 توثيق المستخدم	140
5-7-7 تصميم قواعد البيانات	140
5-8 توزيع الانظمة للادارة العامة	141
5-9 التحقق وأثبات الصحة	156
5-9-1 التيقن	156
الباب السادس: نموذج الربط المقترح	
6-1 :المقدمة	160
6-2: مشروع تطوير نظام الربط بين الادارة العامة للشئون المالية والافرع والادرات	161
6-2-1: دورة الاتصال بين الزبون/العميل	161
6-3 :الأسس التي يعتمد عليها مشروع النظام المقترح	162
6-3-1: دورة السياسات والإجراءات المتعلقة بالخدمة	164
6-3-2: دورة السياسات والإجراءات المتعلقة بالادارة/ الفرع والتي ينبع منها الدورات الفرعية	164
6-4 : شرح النموذج بالتفصيل	165
6-4-1: دورة السياسات والإجراءات المتعلقة بالخدمة	165
6-4-1-1: سياسات وإجراءات الحماية الملموسة للخدمة	167
6-4-1-2: سياسات وإجراءات الحماية غير الملموسة للخدمة	167
6-4-1-3: سياسات وإجراءات تحديث ومواكبة التغيرات التكنولوجية	168
6-4-1-4: سياسات وإجراءات رقابية لحماية الخدمة	168
6-4-2-1: سياسات وإجراءات فصل النظام لتوثيق عمليات النظام الخاصة الخدمة	169
6-4-2-2: سياسات وإجراءات التوثيق لتتبع العمليات الخاصة بالخدمة	170
6-4-2-3: سياسات وإجراءات رقابية لتوثيق عمليات السلع غير الملموسة	170
6-4-3-1: سياسات وإجراءات توكيد إخطار المتعامل مع الخدمة	170
6-4-3-2: سياسات وإجراءات تأكيد المتعامل (الادارة/الفرع)	171
6-4-3-4: سياسات وإجراءات رقابية لتأكيد عملية إخطار المتعامل	171

6-4-3-4: اعتماد جهة تدقيق تؤكد على جميع السياسات والإجراءات	171
6-4-2: دورة السياسات والإجراءات المتعلقة بالادارة /الفرع المتعامل مع الادارة العامة	172
6-4-2-1: سياسات وإجراءات الحماية الملموسة للزبون	172
6-4-2-2: سياسات وإجراءات الحماية غير الملموسة للادارة/الفرع	173
6-4-2-3: سياسات وإجراءات تحديث ومواكبة التغيرات التكنولوجية	174
6-4-2-4: سياسات وإجراءات رقابية لحماية الادارة/الفرع	174
6-4-2-5: سياسات وإجراءات فصل النظام لتوثيق عمليات النظام الخاصة بالادارة/ الفرع.	175
6-4-2-6: سياسات وإجراءات التوثيق لتتبع العمليات الخاصة بالادارة/الفرع	176
6-4-2-7: سياسات وإجراءات رقابية لتوثيق عمليات لادارة/الفرع	176
6-4-2-8: سياسات وإجراءات توكيد إخطار الادارة/الفرع	176
6-4-2-9: سياسات وإجراءات تأكيد الادارة/الفرع باستلامه لتعليمات الادارة	176
6-4-2-10: سياسات وإجراءات رقابية لتأكيد عملية إخطار الادارة/الفرع	177
6-4-2-11: اعتماد جهة تدقيق تؤكد على جميع السياسات والإجراءات السابقة	177
الباب السابع: النتائج والمناقشة	
7-1: مقدمه	179
7-2:اعادة هندسة البرمجيات	179
7-3: نماذج تطوير البرمجيات	180
7-4: المنهجية المقترحة لاعادة هندسة البرمجيات للبرمجيات غيرr كائنية التوجه	181
7-5: مشروع تطوير نظام الربط بين الادارة العامة للشئون المالية والافرع والادرات	181
7-6: دراسة الحالة	182
7-7: الخلاصة	183
الباب الثامن: الخلاصة والتوصيات والدراسات المستقبليه	
8-1 :الخلاصة	185
8-2 :التوصيات	186
8-3 :الدراسات المستقبلية	186
المراجع ومصادر المعلومات	189
الملاحق	197

قائمة الجداول

الجدول	الصفحة
جدول (4-1) القضايا في نظام مرتبات الضباط : متطلبات مرتبات الضباط الرئسى	67
جدول(4-2): سيناريو يوضح ادخال بيانات ضابط جديد فى نظام مرتبات الضباط	69
جدول(4-3):حالة الاستخدام لادخال ضابط جديد فى نظام مرتبات الضباط	70
جدول(4-4): المتطلبات غير الوظيفية لنظام المرتبات	72
جدول(4-5):مبادى توجيهية لسولة الاستخدام	73
جدول(4-6):تحويل تسلسل الاحداث الى التعاون	77
جدول(4-7):مبادى تصميم واجهة المستخدم	85
جدول (5-1) القضايا في نظام مرتبات الضباط	103
الجدول (5-2) :نظرة عامة عن حالات الاستخدام فى نظام مرتبات الضباط	107
الجدول (5-3) أضافة ضابط جديد(الادخال)	110
الجدول (5-4) أضافة ضابط جديد(المراجعه)	110
الجدول (5-5) أضافة ضابط جديد(الاعتماد)	111
الجدول (5-6) حذف ضابط (الانزال)	112
الجدول (5-7) حذف ضابط (المراجعه)	112
الجدول (5-8) حذف ضابط (الاعتماد)	113
الجدول (5-9) فك مرتب(الانزال)	113
الجدول (5-10) فك مرتب(المراجعة)	114
الجدول (5-11) فك مرتب(الاعتماد)	114
الجدول (5-12) تعديل رقم الحساب والبنك (الانزال)	115
الجدول (5-13) تعديل رقم الحساب والبنك(المراجعة)	115
الجدول (5-14) تعديل رقم الحساب والبنك(الاعتماد)	116
الجدول (5-15) اضافة استحقاق (الانزال)	116
الجدول (5-16) اضافة استحقاق (المراجعة) اضافة استحقاق	117
الجدول (5-17) حذف استحقاق (الانزال)	117
الجدول (5-18) حذف استحقاق (المراجعة)	118
الجدول (5-19) اضافة استقطاع (الانزال)	118
الجدول (5-20) اضافة استقطاع (المراجعة) اضافة استقطاع	119

الجدول (5-21) حذف استقطاع (الانزال)	119
الجدول (5-22) حذف استقطاع (المراجعة)	120
الجدول (5-23) تعديل القطاع والرتبة (الانزال)	12
الجدول (5-24) تعديل القطاع والرتبة (المراجعة)	121
الجدول (5-25) تعديل الحالة الاجتماعيه (الانزال)	121
الجدول (5-26) تعديل الحالة الاجتماعيه (المراجعة)	122
الجدول (5-27) اضافة البنك والفرع	122
الجدول (5-28) اضافة وحدة جديده	123
الجدول (5-29) اضافة اسم استحقاق جديد	123
الجدول (5-30) اضافة اسم استقطاع جديد	124
الجدول (5-31) اضافة اسم مستخدم جديد	124
الجدول (5-32) اعطاء الصلاحيات	125
الجدول (5-33) تغير كلمة المرور	125
الجدول (5-34) المتطلبات الغير وظيفيه	126
الجدول (5-35)التعاون والكائنات في حالات استخدام اضافة ضابط جديد(الانزال)	129
الجدول (5-36)نموذج التعاون والكائنات في حالات استخدام اضافة ضابط جديد (المراجعه)	130
الجدول (5-37)نموذج التعاون والكائنات في حالات استخدام اضافة ضابط جديد (الاعتماد)	131
الجدول (5-38) المتطلبات غير الوظيفيه	135
الجدول (5-39) الاستراتيجية الموضوعة للمتطلبات الغير وظيفية	137

قائمة الاشكال

الشكل	الصفحة
الشكل (2-1): النموذج العام للهندسة العكسية	37
الشكل (2-2): النموذج العام لاعادة الهندسة	38
الشكل (3-1): نموذج الشلال	47
الشكل (3-2): النموذج التكرارى	48
الشكل (3-3): يوضح نموذج V-shaped Model	49
الشكل (3-4): نموذج Spiral Model	50
الشكل (3-5): نموذج Extreme Programming (XP)	52
الشكل (3-6): الأطوار الأربعة لإطار العمل التكراري التزايدي	53
الشكل (3-7): طور البناء و يحتوي على سلسلة من الانحدارات المصغرة	54
الشكل (3-8): هرم التصميم غرضى التوجه	60
الشكل (4-1): المنهجية المطبقة فى البحث	64
الشكل (4-2): الفاعين فى نظام مرتبات الضباط	68
الشكل (4-3):يوضح مخطط حالة الاستخدام لنظام المرتبات	70
الشكل (4-4): تعريف الكائنات من حالة استخدام مرتبات الضباط	76
الشكل (4-5) :التعاون فى نظام مرتبات الضباط	78
الشكل (4-6) :التعاون فى نظام مرتبات الضباط	79
الشكل (4-7) :الانظمة الفرعية فى نظام مرتبات الضباط	81
الشكل (4-8) الاجراءات لتطبيق نمط التصميم	83
الشكل (4-9) مخطط يبين معمارية المزود الزبون	87
الشكل (4-10) مخطط يبين حوار المزود الزبون	88
الشكل (4-11) مخطط يبين معمارية العميل البدين والنحيل	90
الشكل (4-12) مخطط يبين معمارية الثلاثة طبقات	91
الشكل (4-13) اعادة هندسة النظام غير الكئنى المنحى	94
الشكل (5-1) انظمة مرتبات الضباط	98
الشكل (5-2) خدمات نظام المرتبات .	100
الشكل (5-3) نظرة شاملة لتنفيذ وظائف النظام .	101
الشكل (5-4) الجهات الفاعلة فى نظام المرتبات .	106

الشكل (5-5) حالات استخدام الخدمات الرئسية للمرتبات	108
الشكل (5-6) حالات استخدام الخدمات المساعدة .	109
الشكل (5-7) حالة استخدام اضافة ضابط جديد	111
الشكل (5-8) كائنات العناصر	127
الشكل (5-9) حدود الكائنات : واجهة المستخدم	128
الشكل (5-10) كائنات التحكم	128
الشكل (5-11) مخطط التعاون لاضافة ضابط	132
الشكل (5-12) كائنات التحكم	133
الشكل (5-13) كائنات التحكم	134
الشكل (5-14) كائنات التحكم	134
الشكل (5-15) الانظمة الفرعية فى النظام القديم	136
الشكل (5-16) الانظمة الفرعية فى النظام الجديد	137
الشكل (5-17) مخطط الانشطه لاضافة ضابط	138
الشكل (5-18) مخطط الحزمة لنظام المرتبات	139
الشكل (5-19) توزيع انظمة الشئيون المالية	139
الشكل (5-20) مخطط الحزمة لنظام البديل النقدى	141
الشكل (5-21) الشاشة الرئسية لبرنامج المرتبات	142
الشكل (5-19) شاشة تفاصيل الخدمات	142
الشكل (5-20) شاشة تفاصيل ادارة النظام	143
الشكل (5-21) شاشة تفاصيل الخدمات المساعدة	143
الشكل (5-22) شاشة تفاصيل البحث	144
الشكل (5-23) شاشة ادراج ضابط جديد	144
الشكل (5-24) شاشة مراجعة ضابط جديد	145
الشكل (5-25) شاشة اعتماد ضابط جديد	145
الشكل (5-26) شاشة ارجاع ضابط جديد	146
الشكل (5-27) شاشة ارجاع ضابط المراجعة	146
الشكل (5-28) شاشة ارجاع ضابط الاعتماد	146
الشكل (5-29) شاشة قك مرتب ضابط	147
الشكل (5-30) شاشة قك مرتب ضابط المراجعة	147

الشكل (5-31) شاشة قك مرتب ضابط الاعتماد	148
الشكل (5-32) شاشة تعديل القطاع والرتبة.	148
الشكل (5-33) شاشة تعديل القطاع والرتبة المراجعه.	149
الشكل (5-34) شاشة تعديل القطاع والرتبة الاعتماد.	149
الشكل (5-35) حذف/اضافة/استقطاع/استحقاق.	150
الشكل (5-36) حذف/اضافة/استقطاع/استحقاق المراجعه.	150
الشكل (5-37) حذف/اضافة/استقطاع/تعديل القطاع والرتبة.	151
الشكل (5-38) حذف/اضافة/استقطاع/تعديل القطاع والرتبة المراجعه.	151
الشكل (5-39) شاشة بحث البيانات الاساسية.	152
الشكل (5-40) شاشة بحث المرتبات الشهريه.	152
الشكل (5-41) اضافة مستخم جديد.	153
الشكل (5-42) شاشة اعطاء الصلاحيات.	153
الشكل (5-43) شاشة تغير كلمة المرور.	154
الشكل (5-44) شاشةاضافة البنوك وفروعها.	154
الشكل (5-45) شاشةاضافة اسماء الاستحقاقات.	155
الشكل (5-46) شاشةاضافة اسماء الاستقطاعات.	155
الشكل (5-47) شاشة قيم المرتب الاساسى.	156
الشكل (6-1): دورة الاتصال بين الادارة/الفرع والادارة العامة	161
الشكل (6-2) : مشروع نظام الربط	163
الشكل (6-3):دورة السياسات والإجراءات المتعلقة بالخدمة	166
الشكل (6-4):دورة السياسات والإجراءات المتعلقة بالادارة/الفرع المتعامل مع الادارة العامة	172

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

بسم الله الرحمن الرحيم
جمهورية السودان
وزارة التعليم العالي والبحث العلمي
جامعة النيلين
كلية الدراسات العليا

استخدام هندسة البرمجيات فى تطوير وتقويم الانظمة المالية

Use of software engineering in the development And evaluation of financial systems

دراسة حالة/ الإدارة العامة للشئون المالية بوزارة الدفاع

بحث أعد لنيل درجة الدكتوراة فى علوم الحاسوب

اعداد:

ياسر محمد محمد احمد التوم

اشراف:

الدكتور/ سيف الدين فتوح

1436هـ - 2015م

الآيـــــــــــــــة

قال الله تعالى :

(وقُلِ اعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ ورَسُولُهُ والْمُؤْمِنُونَ وسَتُرَدُّونَ إِلَى عَالِمِ الغَيْبِ والشَّهَادَةِ فَيُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ)

صدق الله العظيم

سورة التوبة – الآية (105)

الإهداء

إلى ..... من تحت قدمها تكمن الجنة، إلى أمي الحنون.

إلى ..... من جعل مشواري العلمي ممكنا، إلى أبي الرحيم.

إلى ..... من ساندني وآزرني في دربي، إلى زوجتي الصابرة.

إلى ..... من لأجلهم سرت في الدرب، إلى أبنائي نزار، وطلال، واحمد ومحمد الأعزاء.

إليهم جميعا أهدي جهدي المتواضع هذا راجيا الله الإطالة بأعمارهم ليرو ثمرة جهدهم والرحمة والمغفرة لوالدى.

شكر وعرفان

انطلاقاً من العرفان بالجميل، فإنه ليسرني وليثلج صدري أن أتقدم بالشكر والامتنان إلى أستاذي، ومشرفي الأستاذ الدكتور سيف الدين فتوح الذي مدني من منابع علمه بالكثير، والذي ما توانى يوما عن مد يد المساعدة لي وفي جميع المجالات، وحمدا لله بأن يسره في دربي ويسر به أمري وعسى أن يطيل عمره ليبقى نبراسا متلألئا في نور العلم والعلماء.

واتقدم كذلك بجزيل الشكر إلى كل من ضباط وضباط صف جنود والموظفيين بالادارة العامة للشئون المالية ممثلة بقائدها الفريق محاسب استاذ دكتور محمد عثمان الركابى، واللواء دكتور عصام الدين خالد والعميد عمار وادارى الشئون الماليه بريه ممثلة باللواء محاسب خالد ياسين والعميد محاسب عبد القيوم،و العقيد جهاد الدين والعقيد جعفر محمد الحسن والعقيد بهاء الدين ساتى لكل ما قدموه لي من مساعدة ومسانده مكنتني من المضي بخطى ثابتة في مسيرتي العلمية.
كما أتتقدم بجزيل الشكر إلى أساتذتي أعضاء لجنة النقاش الموقرين على ما تكبدوه من عناء في قراءة رسالتي المتواضعة وإغنائها بمقترحاتهم القيمة.

وفي النهاية يسرني أن أتقدم بجزيل الشكر إلى كل من مد لي يد العون في مسيرتي العلمية.

المســـــــتخلص

أذداد الاهتمام بالانظمة المالية بصورة كبيرة لذلك زاد الاهتمام بتطوير هذه الانظمة الى بيئة اكثر مرونه تسمح بردود فعل سريع لاحتياجات العملاء .

ولكن نجد أن هنالك انظمة قديمة موروثة تم تطويرها منذ اكثر من 10 اعوام , ولكنه لاتزال تعمل وموثره فى بيئتها وتقدم الخدمات .المشكلة الرئيسية فى الانظمة الموروثة هو ان هنالك فجوة فى التكنلوجيا بينها وبين النظم المحيطه بها وتقاوم التعديل والتطوير وغالبا الوثائق غير مكتملة وان وجدت تكون خارج نطاق التاريخ الحالى، مما يودى الى صعوبة وتكلفة عالية فى الصيانة .

فنجد أن اعادة هندسة البرمجيات هى التى تقوم بتحسين جودة هذه الانظمة من خلال تقديم تصميم جديد .

لذلك فى هذا البحث تم استكشاف المنهجيات الغرضية التوجه واقتراح منهجية لاعادة هندسة البرمجيات القائمة على غرضية التوجه واقتراح نموذج ربط .

بما أن لغة النمذجة الموحدة (UML) تعتبر لغة قياسية توفر رسوم بيانية لوصف وتوثيق النظام تم استخدامها فى توثيق النظام .

كما تم اقتراح نموذج ربط للربط بين الادارة العامة للشئون المالية وادارتها ووحداتها المختلفة .

كذلك تم تطبيق منهجية اعادة هنسة البرمجيات القائمة على غرضية التوجه المقترحة فى اعادة هندسة برمجيات انظمة الادارة العامة للشئيون المالية بوزارة الدفاع .

وتشير النتائج الى ان تطبيق منهجية أعادة الهندسة المقترح ادى الى المساعدة على تحسين مرونة النظام وسهولة الاستخدام والصيانة والتوسع ، ونموذج الربط أعطى موثوقية ومرونة .

وتشير النتائج على أن النظام الموجود كان به انظمة فرعية ولكنه غير مكتملة والهياكل غير معرفة بطريقة صحيحة .

كذلك تمت اضافة امكانيات جديدة فى النظام الجديد باستخدام قواعد بيانات ORACLE11G و developer6i .

Abstract

The Financial system interest increased significantly so increased interest in the development of these systems to a more flexible environment that allows quick reaction responses to customer needs.

But we find that there is a Legacy system has been developed for more than 10 years, but it still works and effect of in their environment and providing services. The main problems of legacy systems are the enlarging technology gaps between them and their surrounding systems and their resistance to modification and evolution. And Documentation is usually incomplete and out-of-date. Leading to difficulty and high cost of maintenance.

Re-engineering a legacy system mainly focuses on improving quality and maintainability of the existing system by introducing new designs.

In this research was to explore object-oriented methodologies and propose a methodology for re-engineering the software-based object-oriented model and propose a link.

The Unified Modeling Language (UML) is a standard language provides graphs to describe and document the system used in the system documentation.

It was also suggested linking model to link the financial systems between the units and the various departments and branches of public financial management.

In the case study, the proposed methodologies are successfully applied to Re-engineer the legacy system. Results indicate that objected-oriented Methodologies help improve system flexibility, usability, maintainability and extendibility.

Also results indicate that the existing system was its sub-systems, but it is incomplete and the structures are not defined properly.

Also it has added new possibilities in the new system using ORACLE11G databases and developer6i.

المحتويات Contents

الموضوع	الصفحة
الآية	أ
الإهداء	ب
شكر وعرفان	ج
المستخلص	د
Abstract	هـ
قائمة المحتويات	و
قائمة الجداول	ل
قائمة الاشكال	ن
الباب الاول: مقدمة	
1-1 مقدمـــــة	2
1-2 مشكلة البحث	4
1-3 اهداف البحث	6
1-4 محددات البحث	7
1-5 أهمية البحث	7
1-6 حدود البحث	7
1-7 الدراسات السابقة	8
1-8 التعليق على الدراسات السابقة	13
1-9 منهجية البحث	14
1-10 هيكل البحث	14
الباب الثانى: تطوير البرمحيات واعادة هندسة البرمجيات	
2-1 الانظمة القديمة الموروثة Legacy system	16
2-1-1 دورة حياة نموذج الشلال الاصلي water fall	16
2-1-2 تغير الاعمال Business change	17
2-1-3 أرتقاء البرمجيات software Evolution	17
2-2 هندسة البرمجيات للتطوير computer System Evolution	19
2-2-1 تطور الحاسوب computer System Evolution	19
2-2-2 خصائص نظم البرمجيات الكبيرة	20

2-2-3 استعراض حالات الممارسة في نموذج المتطلبات	21
2-3 هندسة البرمجيات software Engineering	21
2-4 جودة البرمجيات software Quality	25
2-5 البرمجيات في اي مكان Software everywhere	26
2-6 صيانة البرمجيات Software maintenance	28
2-6-1 تطوير البرمجيات	32
2-7 اعادة هندسة البرمجيات للتطوير Software Reengineering For Evaluation	33
2-7-1 دورة أعادة هندسة البرمجيات Software reengineering cycle	34
2-7-2 تصنيف اعادة هندسة البرمجيات Taxonomy of software reengineering	34
2-7-3 الهندسة العكسية Reverse engineering	39
2-7-4 الوضع الحالي للطرق الرسمية في اعادة الهندسة	41
2-8 التلخيص	42
الباب الثالث: نماذج عمليات البرمجيات	
3-1 المقدمة	44
3-2 عمليات البرمجيات Software Processes	44
3-3 نماذج عمليات البرمجيات	46
3-3-1 نموذج الشلال	46
3-3-2 :النموذج التكرارى	47
3-3-3 V-shaped Model	48
3-3-4: النموذج الحلزونى Spiral Model	50
3-3-5: Extreme Programming (XP) Model	51
3-4 العمل التكراري التزايدي Iterative, Incremental Framework	52
3-5 العملية الموحّدة من أي بي ام The Rational Unified Process	55
3-6 مفاهيم المنهج الغرضي التوجه في هندسة البرمجيات	58
3-6-1 التحليل الغرضى التوجه	59
3-6-2 التصميم الغرضى التوجه	60
3-6-3 الاختبارات الغرضى التوجه	61
الباب الرابع: المنهجية	
4-1 تقيم جودة النظام	65

2-4 استخراج المتطلبات Requirements Elicitation	66
3-4 تعريف الفاعلين Identifying Actors	67
4-4 تعريف السيناريو Identify Scenarios	68
1-4-4 تعريف حالات الاستخدام Identifying use case	69
2-4-4 النتقاط المتطلبات غير الوظيفية	71
2-4-4 سهولة الاستخدام usability	72
5-4 التحليل الغرضى التوجه OO Analysis	73
1-5-4 تحديد العناصرو الحدود والتحكم للكائنات	74
2-5-4 نموذج التعاونModeling Collaboration	76
3-5-4 التعميم والتخصيص Generalization and specialization	79
6-4 تصميم المنحني الكائنى(OO Design)	79
1-6-4 التقاط اهداف التصميم(Capture Design Goals)	80
2-6-4 معمارية النظام System Architecture	80
1-2-6-4 تصميم النظام الفرعي Subsystem Design	81
2-2-6-4 تصميم الكائن Object Design	82
3-2-6-4 انماط التصميم Design Patterns	82
3-6-4 سهولة الاستخدام	83
1-3-6-4 تصميم واجهة المستخدم	84
2-3-6-4 التصميم المتمركز على المستخدم	84
3-3-6-4 مبادى تصميم واجهات المستخدم	84
4-3-6-4 تفاعل المستخدم والنظام	84
5-3-6-4 دعم المستخدم	84
6-3-6-4 رسائل الخطا	85
1-3-6-4 توثيق المستخدم	85
7-4 الانظمة الموزعة	86
1-7-4 دوافع توزيع الانظمة	86
2-7-4 معمارية الخادم العميل	87
3-7-4 طبقات التطبيق	88
4-7-4 العميل النحيل والعميل البدين	89

4-7-5 معمارية الثلاثة طبقات	91
4-8 اعادة تنظيم النظام غير الكائني Non – OO System Reorganization	92
الباب الخامس: دراسة حالة	
5-1 دراسة حالة (case study)	96
5-2 الاهداف	96
5-3 مقدمة عن النظام	96
5-3-1 انظمة الادارة العامة للشئون المالية	97
5-3-1-1انظمة مرتبات الضباط	98
5-3-2 نظام مرتبات الضباط	98
5-3-2 -1 نظرة الي نموذج الوظائف	98
5-4 تقيم نظام المرتبات	102
5-4-1 تقييم المتطلبات الوظيفية	102
5-4-2 تقيمم المتطلبات الغير وظيفية	102
5-4-2-1 التكرار	103
5-4-2-2 مكافحة النمط	104
5-4-2-3 وظائف /واوجهة المستخدم للمنظمة	104
5-4-2-4 معالجة الاستثناءات	105
5-5 التقاط المتطلبات فى نظام مرتبات الضباط	105
5-5-1 الفاعلوب (اللاعب)	106
5-5-2 السيناريو	106
5-5-3 حالات الاستخدام	106
5-5-4 المتطلبات الغير وظيفية :Non-functional requirements	126
5-6 التحليل الكائني OO Analysis	126
5-6-1 تعريف كائنات الحدود ، كائنات التحكم كائنات العناصر كائنات الاجسام	127
5-6-2 نموذج التعاون modeling collaboration	128
5-6-3 التعميم والتخصيص	132
5-7 التصميم الكائني OO Design	135
5-7-1 اهداف تصميم النظام	135
5-7-2 اعادة تصميم الانظمة الفرعية	136

5-7-3 تصميم الكائنات	137
5-7-4 مخطط الأنشطه :Activity Diagram	138
5-7-5 مخطط التحزيم	138
5-7-6 أعادة تصميم واجهة المستخدم	139
5-7-6-1 دعم المستخدم	140
5-7-6-2 توثيق المستخدم	140
5-7-7 تصميم قواعد البيانات	140
5-8 توزيع الانظمة للادارة العامة	141
5-9 التحقق وأثبات الصحة	156
5-9-1 التيقن	156
الباب السادس: نموذج الربط المقترح	
6-1 :المقدمة	160
6-2: مشروع تطوير نظام الربط بين الادارة العامة للشئون المالية والافرع والادرات	161
6-2-1: دورة الاتصال بين الزبون/العميل	161
6-3 :الأسس التي يعتمد عليها مشروع النظام المقترح	162
6-3-1: دورة السياسات والإجراءات المتعلقة بالخدمة	164
6-3-2: دورة السياسات والإجراءات المتعلقة بالادارة/ الفرع والتي ينبع منها الدورات الفرعية	164
6-4 : شرح النموذج بالتفصيل	165
6-4-1: دورة السياسات والإجراءات المتعلقة بالخدمة	165
6-4-1-1: سياسات وإجراءات الحماية الملموسة للخدمة	167
6-4-1-2: سياسات وإجراءات الحماية غير الملموسة للخدمة	167
6-4-1-3: سياسات وإجراءات تحديث ومواكبة التغيرات التكنولوجية	168
6-4-1-4: سياسات وإجراءات رقابية لحماية الخدمة	168
6-4-2-1: سياسات وإجراءات فصل النظام لتوثيق عمليات النظام الخاصة الخدمة	169
6-4-2-2: سياسات وإجراءات التوثيق لتتبع العمليات الخاصة بالخدمة	170
6-4-2-3: سياسات وإجراءات رقابية لتوثيق عمليات السلع غير الملموسة	170
6-4-3-1: سياسات وإجراءات توكيد إخطار المتعامل مع الخدمة	170
6-4-3-2: سياسات وإجراءات تأكيد المتعامل (الادارة/الفرع)	171
6-4-3-4: سياسات وإجراءات رقابية لتأكيد عملية إخطار المتعامل	171

6-4-3-4: اعتماد جهة تدقيق تؤكد على جميع السياسات والإجراءات	171
6-4-2: دورة السياسات والإجراءات المتعلقة بالادارة /الفرع المتعامل مع الادارة العامة	172
6-4-2-1: سياسات وإجراءات الحماية الملموسة للزبون	172
6-4-2-2: سياسات وإجراءات الحماية غير الملموسة للادارة/الفرع	173
6-4-2-3: سياسات وإجراءات تحديث ومواكبة التغيرات التكنولوجية	174
6-4-2-4: سياسات وإجراءات رقابية لحماية الادارة/الفرع	174
6-4-2-5: سياسات وإجراءات فصل النظام لتوثيق عمليات النظام الخاصة بالادارة/ الفرع.	175
6-4-2-6: سياسات وإجراءات التوثيق لتتبع العمليات الخاصة بالادارة/الفرع	176
6-4-2-7: سياسات وإجراءات رقابية لتوثيق عمليات لادارة/الفرع	176
6-4-2-8: سياسات وإجراءات توكيد إخطار الادارة/الفرع	176
6-4-2-9: سياسات وإجراءات تأكيد الادارة/الفرع باستلامه لتعليمات الادارة	176
6-4-2-10: سياسات وإجراءات رقابية لتأكيد عملية إخطار الادارة/الفرع	177
6-4-2-11: اعتماد جهة تدقيق تؤكد على جميع السياسات والإجراءات السابقة	177
الباب السابع: النتائج والمناقشة	
7-1: مقدمه	179
7-2:اعادة هندسة البرمجيات	179
7-3: نماذج تطوير البرمجيات	180
7-4: المنهجية المقترحة لاعادة هندسة البرمجيات للبرمجيات غيرr كائنية التوجه	181
7-5: مشروع تطوير نظام الربط بين الادارة العامة للشئون المالية والافرع والادرات	181
7-6: دراسة الحالة	182
7-7: الخلاصة	183
الباب الثامن: الخلاصة والتوصيات والدراسات المستقبليه	
8-1 :الخلاصة	185
8-2 :التوصيات	186
8-3 :الدراسات المستقبلية	186
المراجع ومصادر المعلومات	189
الملاحق	197

قائمة الجداول

الجدول	الصفحة
جدول (4-1) القضايا في نظام مرتبات الضباط : متطلبات مرتبات الضباط الرئسى	67
جدول(4-2): سيناريو يوضح ادخال بيانات ضابط جديد فى نظام مرتبات الضباط	69
جدول(4-3):حالة الاستخدام لادخال ضابط جديد فى نظام مرتبات الضباط	70
جدول(4-4): المتطلبات غير الوظيفية لنظام المرتبات	72
جدول(4-5):مبادى توجيهية لسولة الاستخدام	73
جدول(4-6):تحويل تسلسل الاحداث الى التعاون	77
جدول(4-7):مبادى تصميم واجهة المستخدم	85
جدول (5-1) القضايا في نظام مرتبات الضباط	103
الجدول (5-2) :نظرة عامة عن حالات الاستخدام فى نظام مرتبات الضباط	107
الجدول (5-3) أضافة ضابط جديد(الادخال)	110
الجدول (5-4) أضافة ضابط جديد(المراجعه)	110
الجدول (5-5) أضافة ضابط جديد(الاعتماد)	111
الجدول (5-6) حذف ضابط (الانزال)	112
الجدول (5-7) حذف ضابط (المراجعه)	112
الجدول (5-8) حذف ضابط (الاعتماد)	113
الجدول (5-9) فك مرتب(الانزال)	113
الجدول (5-10) فك مرتب(المراجعة)	114
الجدول (5-11) فك مرتب(الاعتماد)	114
الجدول (5-12) تعديل رقم الحساب والبنك (الانزال)	115
الجدول (5-13) تعديل رقم الحساب والبنك(المراجعة)	115
الجدول (5-14) تعديل رقم الحساب والبنك(الاعتماد)	116
الجدول (5-15) اضافة استحقاق (الانزال)	116
الجدول (5-16) اضافة استحقاق (المراجعة) اضافة استحقاق	117
الجدول (5-17) حذف استحقاق (الانزال)	117
الجدول (5-18) حذف استحقاق (المراجعة)	118
الجدول (5-19) اضافة استقطاع (الانزال)	118
الجدول (5-20) اضافة استقطاع (المراجعة) اضافة استقطاع	119

الجدول (5-21) حذف استقطاع (الانزال)	119
الجدول (5-22) حذف استقطاع (المراجعة)	120
الجدول (5-23) تعديل القطاع والرتبة (الانزال)	12
الجدول (5-24) تعديل القطاع والرتبة (المراجعة)	121
الجدول (5-25) تعديل الحالة الاجتماعيه (الانزال)	121
الجدول (5-26) تعديل الحالة الاجتماعيه (المراجعة)	122
الجدول (5-27) اضافة البنك والفرع	122
الجدول (5-28) اضافة وحدة جديده	123
الجدول (5-29) اضافة اسم استحقاق جديد	123
الجدول (5-30) اضافة اسم استقطاع جديد	124
الجدول (5-31) اضافة اسم مستخدم جديد	124
الجدول (5-32) اعطاء الصلاحيات	125
الجدول (5-33) تغير كلمة المرور	125
الجدول (5-34) المتطلبات الغير وظيفيه	126
الجدول (5-35)التعاون والكائنات في حالات استخدام اضافة ضابط جديد(الانزال)	129
الجدول (5-36)نموذج التعاون والكائنات في حالات استخدام اضافة ضابط جديد (المراجعه)	130
الجدول (5-37)نموذج التعاون والكائنات في حالات استخدام اضافة ضابط جديد (الاعتماد)	131
الجدول (5-38) المتطلبات غير الوظيفيه	135
الجدول (5-39) الاستراتيجية الموضوعة للمتطلبات الغير وظيفية	137

قائمة الاشكال

الشكل	الصفحة
الشكل (2-1): النموذج العام للهندسة العكسية	37
الشكل (2-2): النموذج العام لاعادة الهندسة	38
الشكل (3-1): نموذج الشلال	47
الشكل (3-2): النموذج التكرارى	48
الشكل (3-3): يوضح نموذج V-shaped Model	49
الشكل (3-4): نموذج Spiral Model	50
الشكل (3-5): نموذج Extreme Programming (XP)	52
الشكل (3-6): الأطوار الأربعة لإطار العمل التكراري التزايدي	53
الشكل (3-7): طور البناء و يحتوي على سلسلة من الانحدارات المصغرة	54
الشكل (3-8): هرم التصميم غرضى التوجه	60
الشكل (4-1): المنهجية المطبقة فى البحث	64
الشكل (4-2): الفاعين فى نظام مرتبات الضباط	68
الشكل (4-3):يوضح مخطط حالة الاستخدام لنظام المرتبات	70
الشكل (4-4): تعريف الكائنات من حالة استخدام مرتبات الضباط	76
الشكل (4-5) :التعاون فى نظام مرتبات الضباط	78
الشكل (4-6) :التعاون فى نظام مرتبات الضباط	79
الشكل (4-7) :الانظمة الفرعية فى نظام مرتبات الضباط	81
الشكل (4-8) الاجراءات لتطبيق نمط التصميم	83
الشكل (4-9) مخطط يبين معمارية المزود الزبون	87
الشكل (4-10) مخطط يبين حوار المزود الزبون	88
الشكل (4-11) مخطط يبين معمارية العميل البدين والنحيل	90
الشكل (4-12) مخطط يبين معمارية الثلاثة طبقات	91
الشكل (4-13) اعادة هندسة النظام غير الكئنى المنحى	94
الشكل (5-1) انظمة مرتبات الضباط	98
الشكل (5-2) خدمات نظام المرتبات .	100
الشكل (5-3) نظرة شاملة لتنفيذ وظائف النظام .	101
الشكل (5-4) الجهات الفاعلة فى نظام المرتبات .	106

الشكل (5-5) حالات استخدام الخدمات الرئسية للمرتبات	108
الشكل (5-6) حالات استخدام الخدمات المساعدة .	109
الشكل (5-7) حالة استخدام اضافة ضابط جديد	111
الشكل (5-8) كائنات العناصر	127
الشكل (5-9) حدود الكائنات : واجهة المستخدم	128
الشكل (5-10) كائنات التحكم	128
الشكل (5-11) مخطط التعاون لاضافة ضابط	132
الشكل (5-12) كائنات التحكم	133
الشكل (5-13) كائنات التحكم	134
الشكل (5-14) كائنات التحكم	134
الشكل (5-15) الانظمة الفرعية فى النظام القديم	136
الشكل (5-16) الانظمة الفرعية فى النظام الجديد	137
الشكل (5-17) مخطط الانشطه لاضافة ضابط	138
الشكل (5-18) مخطط الحزمة لنظام المرتبات	139
الشكل (5-19) توزيع انظمة الشئيون المالية	139
الشكل (5-20) مخطط الحزمة لنظام البديل النقدى	141
الشكل (5-21) الشاشة الرئسية لبرنامج المرتبات	142
الشكل (5-19) شاشة تفاصيل الخدمات	142
الشكل (5-20) شاشة تفاصيل ادارة النظام	143
الشكل (5-21) شاشة تفاصيل الخدمات المساعدة	143
الشكل (5-22) شاشة تفاصيل البحث	144
الشكل (5-23) شاشة ادراج ضابط جديد	144
الشكل (5-24) شاشة مراجعة ضابط جديد	145
الشكل (5-25) شاشة اعتماد ضابط جديد	145
الشكل (5-26) شاشة ارجاع ضابط جديد	146
الشكل (5-27) شاشة ارجاع ضابط المراجعة	146
الشكل (5-28) شاشة ارجاع ضابط الاعتماد	146
الشكل (5-29) شاشة قك مرتب ضابط	147
الشكل (5-30) شاشة قك مرتب ضابط المراجعة	147

الشكل (5-31) شاشة قك مرتب ضابط الاعتماد	148
الشكل (5-32) شاشة تعديل القطاع والرتبة.	148
الشكل (5-33) شاشة تعديل القطاع والرتبة المراجعه.	149
الشكل (5-34) شاشة تعديل القطاع والرتبة الاعتماد.	149
الشكل (5-35) حذف/اضافة/استقطاع/استحقاق.	150
الشكل (5-36) حذف/اضافة/استقطاع/استحقاق المراجعه.	150
الشكل (5-37) حذف/اضافة/استقطاع/تعديل القطاع والرتبة.	151
الشكل (5-38) حذف/اضافة/استقطاع/تعديل القطاع والرتبة المراجعه.	151
الشكل (5-39) شاشة بحث البيانات الاساسية.	152
الشكل (5-40) شاشة بحث المرتبات الشهريه.	152
الشكل (5-41) اضافة مستخم جديد.	153
الشكل (5-42) شاشة اعطاء الصلاحيات.	153
الشكل (5-43) شاشة تغير كلمة المرور.	154
الشكل (5-44) شاشةاضافة البنوك وفروعها.	154
الشكل (5-45) شاشةاضافة اسماء الاستحقاقات.	155
الشكل (5-46) شاشةاضافة اسماء الاستقطاعات.	155
الشكل (5-47) شاشة قيم المرتب الاساسى.	156
الشكل (6-1): دورة الاتصال بين الادارة/الفرع والادارة العامة	161
الشكل (6-2) : مشروع نظام الربط	163
الشكل (6-3):دورة السياسات والإجراءات المتعلقة بالخدمة	166
الشكل (6-4):دورة السياسات والإجراءات المتعلقة بالادارة/الفرع المتعامل مع الادارة العامة	172

الباب الأول

المقـــــدمـــــــــة

1-1 مقدمـــــة :

أستخدمت هندسة البرمجيات كمفهوم نظرى من حين لاخر فى اوخر الخمسينات من القرن الماضى اما الاستخدام الاول لهذا المصطلح فكان فى موتمر عقد من قبل اللجنة العلمية فى منظمة حلف شمال الاطلسى 1968 حول البرمجيات وقد اخذ هذا المصطلح بالانتشار من ذلك الحين ولاقى اهتماما متزيدا فى نواحى مختلفة .

وقد ظهرت هندسة البرمجيات بسبب عدم استخدام منهجية فى التفكير عند بناء البرمجيات مما ادى الى ظهور اخطاء كثيرة خلال بناء وصيانة البرمجيات بالتالى اصبحت تحتاج الى وقت كبير لتطويرها ولصيانتها وكلفة مالية عالية .

لذلك ظهرت هندسة البرمجيات لوضع اسس وقواعد تهدف الى تصميم وتطوير البرامج بجودة عالية تلبى احتياجات المستخدمين وامكانية وضع اسس لتميز البرامج الجيدة .

فالتغيرات التكنلوجيا والتقنية المستخدمة فى الانظمة المالية تحتاج فى الغالب الى تطوير الانظمة القديمة الى بيئة اكثر مرونة تسمح بردود فعل سريع لاحتياجات العملاء لذكك ينظر الى الجانب التقنى حيث هندسة البرمجيات تتعامل مع المصطلحات المالية والادارية المتخصصة بلادوار والاحتياجات وتودى الى تطوير الانظمة وفق لاسس وقواعد مدروسة وتقيم تلك البرمجيات وفقا لتلك الاسس ماخذة فى الاعتبار القيود المالية والبرمجية .

ان التطوير المستمر قد اصبح اهم المواصفات التى يجب توافرها فى البرامج المالية فالذى لا يعد نفسه اعداد كافيا لمقابلة تلك التغيرات التى تحدث بشكل دائم ومستمر فى سبيل التحسين يصبح مثل سيارة عتيقة وسط سباق تنافس به احدث السيارات .

لقد بدأت الادارة العامة للشئون المالية بوزارة الدفاع فى مواكبة هذا التطور فى عام 1997 م وذلك بحوسبة مجموعة كبيرة من انظمة العمل المالى واعداد البنيات التحتيه له والكوادر الموهلة لذلك العمل .

ففى العام 1997 م كانت البداية بحوسبة الفصل الثانى وحققت التجربة نجاحا مقدرا وكبيرا فى تنفيذ الاهداف المخطط لها والمتمثلة فى سرعة الاجراءات وتامين المعلومات وسلامتها واستخراج البيانات المطلوبة بسهولة ويسر .

وكانت التجربة الثانية فى العام 1999 م بانشاء مكتب الرصد والمتابعة لادخال بيانات الفصل الاول مع سجل الافراد وحقق البرامج اهدافه فى ضبط المرتبات ومحاصرة المخالفات المالية وحصر القوة .

المرحلة الثالثة كانت مرتبات الضباط ثم بعد برامج الميزانية ثم برامج المعاشات ثم بعد ذلك تم حوسبت معظم العمل المالى واخره هو الدفع الالترونى لمرتبات ضباط الصف والجنود .

لكن التغير المستمر فى هيكلة وتطوير القوات المسلحة الى جانب التطوير المستمر من وزارة المالية فى جانب تحسين الاداء المالى فى الدولة جعل لزاما على الادارة العامة للشئون المالية مواكبة التغيرات وتطوير الانظمة المالية له وتحسين اداء الرقابة المالية.

ولكن وبالرغم من إيجابيات التغير فى الهيكلة انه هناك تحديات و محددات جديدة على نظام المعلومات المحاسبي والتي تؤثر على فاعليته وكفاءته ومن أهم هذه المحددات ما يلي:

1. دخول معايير جديدة لتقيم البرمجيات لتحقيق الامان والتوكيدية والموثوقية.
2. ترابط البرامج المالية الحالية عبر الشبكة بعد ان حولها الهيكل الجديد الى عدد من البرامج التى يتم ترابطها عبر الشبكة .
3. تطوير الانظمة الموروثه .

يعرف النظام القديم بانه نظام البرمجيات التي تستخدم تقنيات ولغات كمبيوتر سابقة في حين لايزال في الخدمة وحساس في العمليات اليومية في المنظمة ' التعامل مع مثل هذا النظام يواجه العديد من التحديات علي سبيل المثال وليس الحصر الكلفه الماليه العاليه في تطوير النظام الاول وعدم كفاية الوثائق للصيانة وعدم وجود برامج وادوات دعم.

معظم النظم القديمة تلعب دور هاماً في بيئات الانتاج وتحقق النتائج المتوقعة بنجاح ومع ذلك من الصعب صيانة الانظمة الموروثه. النظام الموروث في كثير من الأحيان قد شهد الكثير من حالات إصلاح الخلل ' والتحدثيات والتعديلات علي طول دورة الحياة . هيكل النظام تدريجياً قد تعرض للتشوية مقارنة مع التصميم الأصلي وبالتالي أصبحت الوثائق الأصلية التي عفا عليها الزمن . كما الفجوة بين الكود والوثائق التي لم يتم تحديثها فتصبح الصيانة من الصعب ومكلفة [9] علي نحو متزايد .

النظم الموروثه التي تم دمجها مع أنظمة أخري والصيانة له تحدث عقبات ايضاً بسبب التوافق والتواصل مع الانظمة الأخري علي الرغم من أن الانظمة الوسيطة(middleware) المتطورة

للمساعدة في التكيف مع الانظمة القديمة مع البيئة الجديدة. لانه يزيد من تعقيد النظام ويتطلب بذل جهد أضافي ليتم صيانتها.

اعادة هندسة البرمجيات هو المكون الرئيسي في تطوير البرمجيات يهدف الي تعزيز القدرة لانظمة البرمجيات على التكيف والصيانة ويشير الىى عملية تصميم وتنفيذ الهياكل الجديدة لنظام البرامج الموجودة . لحل قضايا التكيف والصيانة دون أهمية لتعديل الوظائف الداخلية.

جميع الادوات المستخدمة في مجالات هندسة البرمجيات والاكثر شيوعاً تطبق النمذجة الكائنيه (object – oriented modeling) "OOM" واكثر الاطر شعبيه مثل تصميم المكونات الاساسية للJ22EE و .Net علي اساس OOM أصلاً انشئت لتزيل العقبات في البرمجة الهيكلية والنمذجة في نطاق المشكلة واعادة الاستخدام.

نموذج الكائنى المنحى يحاكي سلوك الانسان في العالم الحقيقي بالتالي يتم نمذجة المشكلة بكفاءة مثل مشكلة الانسان. هذا يفترض أن مجال المشكلة يحتوي علي عدد من الكائنات وكل كائن له سلوكيات وصفات مختلفة [10] علي سبيل المثال أفترض أن الكمبيوتر المحول هو كائن السمات يمكن أن تكون العلامة التجارية وتعريفات الجهاز والسلوكيات للكمبيوتر المحمول يمكن أن تكون يحسب ويعمل.

من اجل تحقيق مرونة أفضل والتوافق مع التكنولوجيا الناشئة وغالباً ماتحتاج النظم القديمة الي تحويله الي انظمة الكائنى المنحى تقليدياً الهندسة العكسية و نموذج الكائنى المنحى والتقنيات الرئيسية المستخدمة في اعادة هندسة الانظمة القديمة .

وقد تمت دراسات علي نطاق واسع لتكنولوجيا الكائنى المنحى موخراً من قبل المهندسين للمساعدة في اعادة بناء OOD بناء علي OOA الموجودة [11] ومع ذلك نادراً ماتم اختبار تقنية علي نظم مالية محددة غير كائنية هذه الرسالة تستكشف تطبيق تقنيات الكائنى المنحى لاعادة بناء نظام غير كائنى ونناقش نتائج التجريب .

2-1 مشكلة البحث:

من المعروف ان التغير فى هيكلة الانظمة وتعديلها وتطوير منهجيات تقيمها يتطلب تطوير لهذه الانظمة لذلك التطور المتسارع فى البرمجيات والانظمة المالية وتغير هيكلة القوات المسلحة ووضع اسس جديدة لتحسين الاداء المالى من وزارة المالية افرز مجموعة من المشاكل تركز

هذه الدراسة علي اعادة هندسة النظم القديمة غير القائمة علي الغرضي التوجه لتحسين جودة النظام والتحديات التي تواجه هذه الدراسة هي :

1/ تكنولجيا الارتقاء بها Technology upgrading

2/ هجرة الوظائف functionality migration

3/ اعادة التقاط المتطلبات requirement re-capture

4/ تحسين الهيكل الغير كائني improving the non-oo structure

تطوير التكنولوجيا يشير الي اعتماد التقنيات الاساسية المتطورة للنظام القديم من اجل توفير اطار اكثر من حيث الفهم حيث يمكن للمهندسين التواصل والفهم وصيانة النظام باعلي كفاءة ومع ذلك في الوقت الحاضر التطوير سريع من اي وقت مضي في نتائج صناعة البرمجيات في وجود فجوة كبيرة بين التقنيات ا والتكنولوجيا القديمة والجديدة الغير موجودة في وثائق النظام القديم فى كل جانب من جوانب البرمجيات بما في ذلك جمع الاحتياجات وطرق التصميم ولغات البرمجة واجراء الاختبارات وتنسيق البيانات والوثائق والتعامل مع هذه الفجوة يتطلب فهم تكنلوجيا النظام القديم والجديدعلي حدا سواء . التقنيات القديمة تستخدم من قبل النظام الموروث وتحتاج الي جهد كبير تشير هجرة الوظائف الي تنفيذ نفس الوظائف التي يقدمها النظام القديم القائم علي تكنولوجيا منصة جديدة ' وتتطلب هذه العملية دقة وفهم الوظائف القائمة ولكن نظراً الي أن الوثائق غير موجودة أو غير كافية قديمة ' القناة الرئيسية لفهم النظام القديم هو عن طريق مراجعة الكود وتشغيل النظام يدوياً .

الفشل في تلبية متطلبات معينه هي واحدة من الاسباب لاعادة تصميم النظام الموروث علي سبيل المثال تنسيق البيانات القديم يجعل النظام غير قادر علي تلبية متطلبات التواصل مع الانظمة التي تستخدم تنسيق البيانات الجديد هذا الفشل يرتفع في الغالب من المتطلبات غير الوظيفية بدلاً من المتطلبات الوظيفية لان الاول هو ضمنى.

خلفية المستخدم ذات الصلة والتغير من الحين للاخر المتطلبات الوظيفيه مستقره نسبيا لالتقاط المتطلبات غير الوظيفية بكفاءة مطلوب من المطوريين مراقبة النظام من وجهة نظر المستخدم ' التي هي اكثر صعوبة كمايبدو ايضاً لاتوجد وسيلة منهجية للمساعدة والاستدلال الا بالاساليب البحثية.

يسبب الدعم المحدود لاعادة الكود اللاكائني فمن الصعب تحقيق نفس المرونة . التي يتم توفيرها من قبل لغات الكائنى المنحنى معظمها يبتبع مواصفات الكائنى المنحنى بالرغم من اختلاف بناء الجمل فى حين تتبع اللغات غير الكائنى المنحنى تختلف نماذج التصميم وليس هنالك طريقة منهجية وضعت النظام غير الكائني لمساعدة التحليل والتصميم في معظم الحالات الاداة الرئيسيه التي يتم استخدامها في اعادة هندسة النظام غير اغرضي التوجه هو تجارب المطوريين ومراجعة الكود.

3-1 أهداف البحث

الهدف الرئيسي من هذا البحث هو

1. اعادة هندسة انظمة البرمجيات القديمة .
2. ايجاد نموذج ربط من اجل تحسين الموثوقية والمرونة والصيانة

ولتحقيق هذا الهدف يتم تأسيس اربعة اهداف .

أ. تحليل النظام القديم

تحليل النظام القديم هو اكتشاف نشاة النظام القديم والتكنولوجيا ذات الصلة' والقضايا القائمة وودورها في سياق النظام . الفهم الجيد لتلك العوامل يمكن أن تساعد في اختيار المنهجية والادوات المناسبة في عملية اعادة الهندسة.

ب. استكشاف منهجيات الهندسة الكائنية

استكشاف معظم اجراءات هندسة البرمجيات المشتركة والمنهجيات لفهم استخدامها والسيناريوهات وتعطي الدراسة شاملة RUP واطار تطوير البرمجيات المتكررة .

ج. اقتراح منهجية مناسبة

اختيار منهجيات خاصة تستخدم منهجيات المنحى الكائنى المناسبة للمشاريع الصغيرة ' والانظمة المالية وبناء خطوات تطبيق لتحديد الهياكل الفرعية في غير الكائنى المنحنى الـــ RUP و نموذج الكائنى المنحنى تتكون من المنهجيات المختلفة التي لاتكون مناسبة لجميع انواع المشاريع ينبغي أن تكون مصممة خصيصاً لتناسب حجم ومتطلبات المشاريع لتحقيق لتجربه جيده للمشروع ' ويجب تطبيق التقنيات التي تمثل مجال المشكلة

4. اجراء دراسة حالة لتطبيق منهجيات اعادة هندسة النظام القديم " الانظمة المالية لادارة الشون المالية" واختباره لاظهار النتائج .

تحليل النظام القديم وسيتم تطبيق المنهجيات للتعرف علي قضايا نظام القديم وانشاء الحل وتنفيذ عملية اعادة الهندسة لتحسين جودة كل النظام .

5.بناء نموذج ربط لربط الانظمة عبر الوحدات والادارات المختلفه.

1-4 محددات البحث

يعتقد الباحث أنه من أهم محدداتها ما يلي:

1. حداثة الموضوع.
2. قلة البحوث المتعلقة بالموضوع.

1-5 أهمية البحث

تنبع أهمية الدراسة وبشكل رئيسي من أهمية الادارة العامة للشئون المالية بوزارة الدفاع اذ به كل ميزانيات القوات المسلحة والتى تعتبر نسبة كبيرة من ميزانية الدولة فلابد من التطوير واستخدام كل ادوات التكنلوجيا الحديثه والمتطوره واستخدام ادوات ربط امنه وموثوقة.

كما ان وجود نماذج واليات لتقييم البرمجيات المالية يجعله تواكب التطورات الحديثه فى مجال البرمجيات . ومن ثم الحفاظ على المال العام وبذلك المحافظة على الاقتصاد الوطنى .

1-6 **حدود البحث**

1. تتمثل الحدود المكانية للبحث فى وحدات الادارة العامة للشئون المالية بوزارة الدفاع
2. اما الحدود الزمانية فتتمثل فى حوسبة العمل المالى فى الفترة من 2012_ 2015م.

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

المراجع والمصادر العلميه

المراجع

المراجع العربية

[1]. د عوض حاج على الطرق العلمية فى التصميم وكفاءة النظم الالية.مطبعة جامعة الخرطوم 1991.

[2] مهندس عبد الحميد بسيوني،" لغة النمذجة الموحدة"، دار الكتب العلمية للنشر والتوزيع، 2005م .

[3] مهندس عبد الحميد بسيوني،" أساسيات هندسة البرمجيات"، دار الكتب العلمية للنشر والتوزيع، 2005م .

]4[د.درغام ميخائيل،" هندسة البرمجيات باستخدام لغة (ADA) دار الرضا للنشر، الطبعة الأولى 2000 م

[5] د .حسين صالح سلامة،" تصميم النظم المحوسبة ومدخل إلى هندسة البرام ج"، مكتبة المجتمع العربي للنشر،الطبعة الأولى، 2004 م.

[6]المهندس زاهر الحاج حسين،" هندسة البرمجيات ثنائية الهندسة والإدارة"، دار شعاع للنشر والعلوم، الطبعةالأولى، 2006 م.

[7] روجر بريسمان ، "هندسة البرمجيات" ، الدار العربية للعلوم ، مركز التعريب والبرمجة ، الطبعة الأولى ، 1425هـ - 2004م .

http://www.w3.org

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

http://www.c4arab.com/

المراجع الاجنبيه

[9]. Boehm, B. (2006). A View of 20th and 21st Century Software Engineering. *the 28th International conference on Software engineering* (pp. 12-29). New York, New York,

[10]. Rumbaugh, J., Blaha, M. R., Lorensen, W., & Edd, F. (1990). *Object-oriented*

modeling and design. Prentice-Hall.

[11]. Lung, C.-H. (1998). Software Architecture Recovery and Restructuring. *Proc. of the 3rd International Software Architecture Workshop (ISAW)*, (pp. 101-104).

[12] Yang, H., and K. H. Bennett, ‘‘Acquiring Entity-Relationship Attribute Diagrams from Code and Data Through Program Transformation,’’ IEEE International Conference on Software Maintenance (ISCM ’95), Nice, France, October 1995.

[13] Royce, W. W., ‘‘Managing the Development of Large Software Systems ,’’ IEEE WESCON, August 1970.

[14] Pressman, R. S., Software Engineering—A Practitioner’s Approach, New York:-Hill, 1987.

[15] Brooks, F. P., ‘‘No Silver Bullet,’’ IEEE Computer, Vol. 20, No. 4, April 1987,pp. 10–19.

[16] Parnas, D. L., ‘‘Designing Software,’’ IEEE Trans. on Software Engineering, Vol. 5,No. 2, March 1979.

[17] Lubers, M., C. Potts, and C. Richter, ‘‘A Review of the State of the Practice in Requirements Modeling,’’ Proc. International Requirements Engineering Symposium,Los Alamitos, California, 1993.

[18] ‘‘Report on the NATO Software Engineering Conference, Garmisch, 1968’’in Software Engineering Concepts and Techniques, P. Naur and B. R. Randell (eds.),Petrocelli/Charter, 1976.

[19] Carey, J. M., ‘‘Prototyping: Alternative Systems Development Methodology,’’Information and Software Technology, Vol. 32, No. 2, 1990.

[20] Fisher, A. S., CASE: Using Software Development Tools, New York: John Wiley and Sons, 1988.

[21] McCall, J., P. Richards, and G. Walters, ‘‘Factors in Software Quality,’’ presented at NTIS, November 1977.

[22] Wasserman, A. I., ‘‘Software Engineering Environments,’’ Advances in Computers, Vol. 22, 1983.

[23] Ramamoorthy, C. V., et al., ‘‘Software Engineering: Problems and Perspectives,’’ IEEE Computer, October 1984.

[24] Ratcliff, B., Software Engineering Principles and Methods, Oxford, England: Blackwell Scientific Publications, 1987.

[25] Bennett, K. H., ‘‘An Overview of Maintenance and Reverse Engineering,’’ in The REDO Compendium, Chichester, England: John Wiley and Sons, 1993.

[26] Lehman, M. M., ‘‘Programs, Life Cycles, and Laws of Software Evolution,’’ Proc. IEEE, Vol. 19, No. 9, 1980, pp.

[27] ANSI standard 729, IEEE Standard Glossary of Software Engineering Terminology,1983.

[28] Swanson, E. B., ‘‘The Dimension of Maintenance,’’ in Second International Conference on Software Engineering, Los Alamitos, California: IEEE Computer Society, 1976.

[29] Bennett, K. H., ‘‘The Software Maintenance of Large Software Systems: Management Method and Tools,’’ Durham University, Technical Report, 1989.

[30] Yang, H., ‘‘Is Year 2000 Problem a Paper Tiger or a Real Tiger,’’ presented at the IEEE International Computer Software and Application Conference (CompSac’97),Washington, D.C., August 1987.

[31] Bennett, K. H., J. Denier, and J. Estublier, ‘‘Environments for Software Maintenance,’’ Durham University, Technical Report, 1989.

[32] Standard FIPS PUB 106, ‘‘Guidelines on Software Maintenance,’’ U.S.Department of Commerce/National Bureau of Standards, June 1984.

[33] Robson, D. J., et al., ‘‘Approaches to Program Comprehension,’’ Journal of Systems Software, Vol. 14, February 1991, pp. 79–84.

[34] Kopetz, H., Software Reliability, Berlin: Springer-Verlag, 1979.

[35] Gilb, T., ‘‘A Comment on the Definition of Reliability,’’ ACM Software Engineering Notes, Vol. 4, No. 3, July 1979.

[36] Hayes, I. J., Specification Case Studies, Englewood Cliffs, NJ: Prentice Hall, 1987.

[37] Jones, C. B., Systematic Software Development Using VDM, Englewood Cliffs, NJ:Prentice Hall, 1986.

[38] Dijkstra, E. W., A Discipline of Programming, Englewood Cliffs, NJ: Prentice Hall,1976.

[39] Morgan, C. C., Programming from Specifications, Second Edition, Englewood Cliffs,NJ: Prentice Hall, 1994.

[40] Ward, M., ‘‘Derivation of a Sorting Algorithm,’’ Durham University, Durham,U.K., Technical Report, 1990 (http://www.dur.ac.uk/~dcs0mpw/martin/papers/sorting-t.ps.gz).

[41] Morgan, C. C., ‘‘The Specification Statement,’’ Trans. Programming Lang. Syst.,Vol. 10, 1988, pp. 403–419.

[7] Morgan, C. C., and K. Robinson, ‘‘Specification Statements and Refinement

[8] Morgan, C. C., and T. Vickers, On the Refinement Calculus, Berlin: Springer-Verlag, 1993.

[42] Ward, M., ‘‘Foundations for a Practical Theory of Program Refinement and

Transformation,'' Durham University, Durham, U.K., Technical Report, 1994 (http://www.dur.ac.uk/~dcs0mpw/martin/papers/foundation2-t.ps.gz).

[43] Karp, C. R., Languages with Expressions of Infinite Length, Amsterdam: North-Holland, 1964.

[44] Scott, D., ''Logic with Denumerably Long Formulas and Finite Strings of Quantifiers,'' in Symposium on the Theory of Models, J. Addison, L. Henkin, and A. Tarski (eds.), Amsterdam: North-Holland, 1965, pp. 329–341.

[45] Engeler, E., Formal Languages: Automata and Structures, Chicago: Markham, 1968.

[46] Back, R. J. R., Correctness Preserving Program Refinements (Mathematical Centre Tracts), Vol. 131, Amsterdam, the Netherlands, Mathematisch Centrum, 1980.

[47] Back, R. J. R., ''A Calculus of Refinements for Program Derivations,'' Acta Informatica, Vol. 25, 1988, pp. 593–624.

[48] Morgan, C. C., Programming from Specifications, Second Edition, Englewood Cliffs, NJ: Prentice Hall, 1994, p. 180.

[49] Back, R. J. R., and J. von Wright, ''Refinement Concepts Formalized in Higher-Order Logic,'' Formal Aspects of Computing, Vol. 2, 1990,

pp. 247–272.

[50] Ward, M., ‘‘A Definition of Abstraction,’’ J. Software Maintenance: Research and
Practice, Vol. 7, No. 6, November 1995, pp. 443–450 (http://www.dur.ac.uk/
~dcs0mpw/martin/papers/abstraction-t.ps.gz).

[51] Bull, T., ‘‘An Introduction to the WSL Program Transformer,’’ Conference on
Software Maintenance, San Diego, CA, November 26–29, 1990.

[52] Ward, M., and K. H. Bennett, ‘‘A Practical Program Transformation System for
Reverse Engineering,’’ Working Conference on Reverse Engineering, Baltimore,
MD, May 21–23, 1993 (http://www.dur.ac.uk/~dcs0mpw/martin/papers/
icse.ps.gz).

[53] Ward, M., F. W. Calliss, and M. Munro, ‘‘The Maintainer’s Assistant,’’
Conference on Software Maintenance, Miami, Florida, October 16–19, 1989
(http://www.dur.ac.uk/~dcs0mpw/martin/papers/MA-89.ps.gz).

[54] Ward, M., Proving Program Refinements and Transformations, Oxford University,
Oxford, U.K., Ph.D. thesis, 1989.

[55] Ward, M., ‘‘A Recursion Removal Theorem—Proof and Applications,’’

Durham University, Durham, U.K., Technical Report, 1991 (http://www.dur.-ac.uk/~dcs0mpw/martin/papers/rec-proof-t.ps.gz).

[56] Ward, M., ‘‘A Recursion Removal Theorem,’’ New York-Heidelberg-Berlin: Springer-Verlag, Proc. of the 5th Refinement Workshop, London, January 8–11, 1992 (http://www.dur.ac.uk/~dcs0mpw/martin/papers/ref-ws-5/ps.gz).

[57] Ward, M., ‘‘Recursion Removal/Introduction by Formal Transformation: An Aid to Program Development and Program Comprehension,’’ Comput. J., Vol. 42, No. 8, 1999, pp. 650–673.

[58] Ward, M., ‘‘Derivation of Data-Intensive Algorithms by Formal Transformation,’’ IEEE Trans. on Software Engineering, Vol. 22, No. 9, September 1996, pp. 665–686 (http://www.dur.ac.uk/~dcs0mpw/martin/papers/sw-alg.ps.gz).

[59] Arsac, J., ‘‘Transformation of Recursive Procedures, in Tools and Notations for Program Construction, D. Noel (ed.), Cambridge, U.K.: Cambridge University Press, 1982, pp. 211–265.

[60] Arsac, J., ‘‘Syntactic Source-to-Source Program Transformations and Program Manipulation,’’ Comm. ACM, Vol. 22, No. 1, January 1982, pp. 43–54.

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الملاحق

==========جدول المستخدمين==========

```
CREATE TABLE USERS (
  US_ID        NUMBER (3)    NOT NULL,
  US_NAME      VARCHAR2 (100),
  US_PASS      VARCHAR2 (6),
  US_SECTION   VARCHAR2 (15),
  US_PRV       NUMBER (2),
  CONSTRAINT USER_PK
  PRIMARY KEY ( US_ID ) ) ;
```

----------جدول الصلاحيات-2----------

```
CREATE TABLE USER_PRIV (
  US_ID                 NUMBER (3)    NOT NULL,
  US_PRV                NUMBER (1),
  CHANG_PASS            NUMBER (1),
  ADD_USER              NUMBER (1),
  ADD_PRIV              NUMBER (1),
  UNIT_IN               NUMBER (1),
  UNIT_UP               NUMBER (1),
  UNIT_DEL              NUMBER (1),
  ADD_OFF_IN            NUMBER (1),
  ADD_OFF_UP            NUMBER (1),
  ADD_OFF_DEL           NUMBER (1),
  ADD_OFF_REV           NUMBER (1),
  ADD_OFF_AATMAD        NUMBER (1),
  SECT_MDFIC_IN         NUMBER (1),
  SECT_MDFIC_UP         NUMBER (1),
  SECT_MDFIC_DEL        NUMBER (1),
  SECT_MDFIC_REV        NUMBER (1),
  SECT_MDFIC_AATMAD     NUMBER (1),
  BANK_ACC_IN           NUMBER (1),
  BANK_ACC_UP           NUMBER (1),
  BANK_ACC_DEL          NUMBER (1),
  BASIC_DATA_IN         NUMBER (1),
  BASIC_DATA_UP         NUMBER (1),
  BASIC_DATA_DEL        NUMBER (1),
  BASIC_DATA_AATMAD     NUMBER (1),
```

```
SOCIAL_IN               NUMBER (1),
SOCIAL_UP               NUMBER (1),
SOCIAL_DEL              NUMBER (1),
SOCIAL_REV              NUMBER (1),
SOCIAL_AATMAD           NUMBER (1),
SAL_IN                  NUMBER (1),
SAL_UP                  NUMBER (1),
SAL_DEL                 NUMBER (1),
SAL_REV                 NUMBER (1),
SAL_AATMAD              NUMBER (1),
SEARCH                  NUMBER (1),
SEARCH_DATA             NUMBER (1),
SEARCH_SAL              NUMBER (1),
REP_TSWIA               NUMBER (1),
REP_SAL                 NUMBER (1),
REP_MOZANA              NUMBER (1),
REP_REDCUTION           NUMBER (1),
REP_LITERS              NUMBER (1),
REP_REFRESH             NUMBER (1),
REP_GARIMASH            NUMBER (1),
BANK_IN                 NUMBER (1),
BANK_UP                   NUMBER (1),
BANK_DEL                  NUMBER (1),
MERIT_IN                  NUMBER (1),
MERIT_UP                NUMBER (1),
MERIT_DEL                 NUMBER (1),
RED_IN                    NUMBER (1),
RED_UP                    NUMBER (1),
RED_DEL                   NUMBER (1),
RED_MERIT_IN              NUMBER (1),
RED_MERIT_UP              NUMBER (1),
RED_MERIT_DEL             NUMBER (1),
DEF_IN                    NUMBER (1),
DEF_UP                    NUMBER (1),
DEF_DEL                   NUMBER (1),
DEF_REP                   NUMBER (1),
DEF_REV                 NUMBER (1),
DEF_AATMAD              NUMBER (1),
MOV                     NUMBER (1),
```

```
  DEL_OFF_IN            NUMBER (1),
  DEL_OFF_REV           NUMBER (1),
  DEL_OFF_DEL_AATMAD    NUMBER (1),
  ADD2_OFF_IN           NUMBER (1),
  ADD2_OFF_REV          NUMBER (1),
  ADD2_OFF_DEL_AATMAD   NUMBER (1),
  ADD2_WORK_IN          NUMBER (1),
  ADD2_WORK_REV         NUMBER (1),
  ADD2_WORK_DEL_AATMAD  NUMBER (1),
  TOILA_OFF_IN          NUMBER (1),
  TOILA_OFF_REV         NUMBER (1),
  TOILA_OFF_DEL_AATMAD  NUMBER (1),
  AGZ_OFF_IN            NUMBER (1),
  AGZ_OFF_REV           NUMBER (1),
  AGZ_OFF_DEL_AATMAD    NUMBER (1),
  M9_OFF_IN             NUMBER (1),
  M9_OFF_REV            NUMBER (1),
  M9_OFF_DEL_AATMAD     NUMBER (1),
  RED_OFF_IN            NUMBER (1),
  RED_OFF_REV           NUMBER (1),
  RED_OFF_DEL_AATMAD    NUMBER (1),
  BDIAL_IN              NUMBER (1),
  BDIAL_REV             NUMBER (1),
  BDIAL_AATMAD          NUMBER (1),
  BDIAL_DOC_IN          NUMBER (1),
  BDIAL_DOC_REV         NUMBER (1),
  BDIAL_DOC_AATMAD      NUMBER (1),
  BDIAL_SEARCH          NUMBER (1),
  BDIAL_SEARCH2         NUMBER (1),
  BDIAL_REP1            NUMBER (1),
  BDIAL_REP2            NUMBER (1),
  BDIAL_REP3            NUMBER (1) ) ;

ALTER TABLE USER_PRIV ADD  CONSTRAINT USER_PRV_FK
 FOREIGN KEY (US_ID)
  REFERENCES SAL2015.USERS (US_ID) ;
```

----------جدول البيانات الاساسية---3------

```
CREATE TABLE TBL_SAL_OFFICER_BANK (
  MILITARY_NO     NUMBER (10)   NOT NULL,
  NAME            VARCHAR2 (70),
  RANKER_NO       NUMBER (10)   NOT NULL,
  ACC_NO          NUMBER (15),
  BANK_NO         NUMBER (10)   NOT NULL,
  CENTER_NO       NUMBER (10),
  SOCIAL_STATE    VARCHAR2 (30),
  OFFICER_STATUS  VARCHAR2 (30),
  SECTOR_NO         NUMBER (10)   NOT NULL,
  BANK_B_NO         NUMBER (10)   NOT NULL,
  TAX               NUMBER (15,3),
  CON_SAL           NUMBER (15,3),
  ADNAME            NUMBER (12),
  AD_NAME           VARCHAR2 (30),
  TYPE              VARCHAR2 (20),
  TKHSAS            VARCHAR2 (20),
  START_DATE        DATE,
  QUALTY            VARCHAR2 (20),
  ADD_PERIOD        NUMBER (4),
  TOLAP             NUMBER (4),
  RAF               NUMBER (5),
  US_ID             NUMBER (3),
  REVEW             NUMBER (3),
  INPUT_DATE        DATE,
  REMARK            VARCHAR2 (100),
  COL               VARCHAR2 (5),
  SPEC              VARCHAR2 (10),
  SEX               VARCHAR2 (5),
  SOL_DATE          DATE,
  TKIND             VARCHAR2 (8),
  SRIA              VARCHAR2 (25),
  ST_NO             NUMBER (2),
  PR_NO             NUMBER (2),
  LOC               VARCHAR2 (15),
  TVIL              VARCHAR2 (15),
  HOUSE           VARCHAR2 (10),
```

```
  ADDRESS          VARCHAR2 (50),
  PHONE            CHAR (15),
  JOB_ID           NUMBER (2),
  ARCHIF_SER       NUMBER (4),
  ARCHIF_CHAR      CHAR (1),
  SALA             NUMBER (2),
  MILITARY_NO1     NUMBER (10),
  TYP_NO           NUMBER (2),
  GROUP_NO         NUMBER (2),
  JOP_NO           NUMBER (2),
  CONSTRAINT TBL_SAL_OFFICER_BANK_UNIG
    UNIQUE (MILITARY_NO1),
  CONSTRAINT TBL_SAL_OFFICER_BANK_PK
  PRIMARY KEY ( MILITARY_NO ) ) ;
```

----------4-جدول البيانات الاساسية----------

```
CREATE TABLE TBL_SAL_OFFICER_BANK (
  MILITARY_NO      NUMBER (10)   NOT NULL,
  NAME             VARCHAR2 (70),
  RANKER_NO        NUMBER (10)   NOT NULL,
  ACC_NO           NUMBER (15),
  BANK_NO          NUMBER (10)   NOT NULL,
  CENTER_NO        NUMBER (10),
  SOCIAL_STATE     VARCHAR2 (30),
  OFFICER_STATUS   VARCHAR2 (30),
  SECTOR_NO        NUMBER (10)   NOT NULL,
  BANK_B_NO        NUMBER (10)   NOT NULL,
  TAX              NUMBER (15,3),
  CON_SAL          NUMBER (15,3),
  ADNAME           NUMBER (12),
  AD_NAME          VARCHAR2 (30),
  TYPE             VARCHAR2 (20),
  TKHSAS           VARCHAR2 (20),
  START_DATE       DATE,
  QUALTY           VARCHAR2 (20),
  ADD_PERIOD       NUMBER (4),
  TOLAP           NUMBER (4),
```

```
  RAF             NUMBER (5),
  US_ID           NUMBER (3),
  REVEW           NUMBER (3),
  INPUT_DATE      DATE,
  REMARK          VARCHAR2 (100),
  COL             VARCHAR2 (5),
  SPEC            VARCHAR2 (10),
  SEX             VARCHAR2 (5),
  SOL_DATE          DATE,
  TKIND             VARCHAR2 (8),
  SRIA              VARCHAR2 (25),
  ST_NO             NUMBER (2),
  PR_NO             NUMBER (2),
  LOC               VARCHAR2 (15),
  TVIL              VARCHAR2 (15),
  HOUSE             VARCHAR2 (10),
  ADDRESS           VARCHAR2 (50),
  PHONE             CHAR (15),
  JOB_ID            NUMBER (2),
  ARCHIF_SER        NUMBER (4),
  ARCHIF_CHAR       CHAR (1),
  SALA              NUMBER (2),
  MILITARY_NO1      NUMBER (10),
  TYP_NO            NUMBER (2),
  GROUP_NO          NUMBER (2),
  JOP_NO            NUMBER (2),
  CONSTRAINT TBL_SAL_OFFICER_BANK_UNIG
    UNIQUE (MILITARY_NO1),
  CONSTRAINT TBL_SAL_OFFICER_BANK_PK
  PRIMARY KEY ( MILITARY_NO ) ) ;
```

----------جدول بيانات المرتبات الشهرية---5------

```
CREATE TABLE TBL_SAL_OFFICER (
  MILITARY_NO  NUMBER (10)   NOT NULL,
  MONTH        NUMBER (2)    NOT NULL,
  YEAR         NUMBER (4)    NOT NULL,
  SAL_MONTH    NUMBER (32,2),
```

```
  RANKER_NO     NUMBER (10),
  TOTAL         NUMBER (12,2),
  REMMARK       VARCHAR2 (50),
  TAX           NUMBER (15,3),
  CON_SAL       NUMBER (15,3),
  SECTOR_NO     NUMBER (10),
  MERIT         NUMBER,
  REDUC         NUMBER,
  BANK_NO       NUMBER (10),
  BANK_B_NO     NUMBER (10),
  REV           NUMBER (1),
  INC_REASON    VARCHAR2 (100),
  ACC_NO        NUMBER (15),
  SOCIAL        NUMBER (1),
  SAL_KIND      NUMBER (1),
  TYPE          VARCHAR2 (20),
  TYP_NO        NUMBER (2),
  GROUP_NO      NUMBER (2),
  JOP_NO        NUMBER (2),
  TOTAL2        NUMBER (12,2),
  NO1           NUMBER (10),
  CENTER_NO     NUMBER (10),
  CONSTRAINT TBL_SAL_OFFICER_PK
  PRIMARY KEY ( MILITARY_NO, MONTH, YEAR ) ) ;

ALTER TABLE TBL_SAL_OFFICER ADD  CONSTRAINT TBL_SAL_OFFICERWORK
 FOREIGN KEY (TYP_NO, GROUP_NO)
  REFERENCES SAL2015.MIL_GROUP (TYP_NO, GROUP_NO) ;

ALTER TABLE TBL_SAL_OFFICER ADD  CONSTRAINT TBL_SAL_OFFICER_FK
 FOREIGN KEY (MILITARY_NO)
  REFERENCES SAL2015.TBL_SAL_OFFICER_BANK (MILITARY_NO) ;
```

----------جدول بيانات الاستحقاقات الشهرية---6------

```
CREATE TABLE TBL_MERIT (
  MERIT_NO      NUMBER (10)   NOT NULL,
  YEAR          NUMBER (4)    NOT NULL,
```

```
  MONTH         NUMBER (2)     NOT NULL,
  MILITARY_NO   NUMBER (10)    NOT NULL,
  MERIT_VALUE   NUMBER (22,2),
  RANKER_NO     NUMBER (10),
  MR_V          NUMBER (20),
  CONSTRAINT TBL_MERIT_PK
  PRIMARY KEY ( MERIT_NO, YEAR, MONTH, MILITARY_NO ) ) ;

ALTER TABLE TBL_MERIT ADD  CONSTRAINT TBL_MERIT_FK
 FOREIGN KEY (MILITARY_NO, MONTH, YEAR)
  REFERENCES SAL2015.TBL_SAL_OFFICER (MILITARY_NO, MONTH, YEAR) ;
```

----------جدول بيانات الاستقطاعات الشهرية---7------

```
CREATE TABLE TBL_REDCUTION (
  REDC_NO       NUMBER (10)    NOT NULL,
  YEAR          NUMBER (4)     NOT NULL,
  MONTH          NUMBER (2)     NOT NULL,
  MILITARY_NO   NUMBER (10)    NOT NULL,
  REDC_NALUE    VARCHAR2 (30),
  RANKER_NO     NUMBER (10),
  REDC_VALUE    NUMBER (32,2),
  CONSTRAINT TBL_REDCUTION_PK
  PRIMARY KEY ( REDC_NO, YEAR, MONTH, MILITARY_NO ) ) ;

ALTER TABLE TBL_REDCUTION ADD  CONSTRAINT TBL_REDCUTION_FK
 FOREIGN KEY (MILITARY_NO, MONTH, YEAR)
  REFERENCES SAL2015.TBL_SAL_OFFICER (MILITARY_NO, MONTH, YEAR) ;
```

----------جدول اسماء الاستحقاقات--8-------

```
CREATE TABLE TLK_MERIT1 (
  MERIT_NO      NUMBER (10),
  MERIT_NAME    VARCHAR2 (50),
  MERIT_VALUE   NUMBER (12,2),
  RANKER_NO     NUMBER (10),
  MERIT_NO1     NUMBER (10) ) ;

ALTER TABLE TLK_MERIT1 ADD  CONSTRAINT TLK_MERIT1_FK
```

```
 FOREIGN KEY (MERIT_NO1)
  REFERENCES SAL2015.MERIT (MERIT_NO) ;
```

----------جدول اسماء الاستقطاعات---9-----

```
CREATE TABLE TLK_REDCUTION1 (
  REDC_NO      NUMBER (10),
  REDC_NAME    VARCHAR2 (50),
  REDC_VALUE   NUMBER (12,2),
  RANKER_NO    NUMBER (10),
  REDUCNO1     NUMBER (10),
  REDC_VALUE2  NUMBER (12,2) ) ;

ALTER TABLE TLK_REDCUTION1 ADD  CONSTRAINT TLK__REDCUTION1_FK
 FOREIGN KEY (REDUCNO1)
  REFERENCES SAL2015.REDCUTION (REDC_NO) ;
```

----------جدول اسماء الوحدات---10-----

```
CREATE TABLE TLK_CENTERS (
  CENTER_NO    NUMBER (10)   NOT NULL,
  CENTER_NAME  VARCHAR2 (50),
  CONSTRAINT TLK_CENTERS_PK
  PRIMARY KEY ( CENTER_NO ) ) ;
```

----------جدول اسماء القطاعات---11------

```
CREATE TABLE TLK_SECTOR (
  SECTOR_NO    NUMBER (10)   NOT NULL,
  RANKER_NO    NUMBER (10)   NOT NULL,
  BASIC_SAL    NUMBER (12,2),
  SECTOR_NAME  VARCHAR2 (50),
  FRG1         NUMBER (12,2),
  CONSTRAINT TLK_SECTOR_PK
  PRIMARY KEY ( SECTOR_NO, RANKER_NO ) ) ;
```

----------جدول اسماءالرتب--12------

```
CREATE TABLE TLK_RANKER1 (
  RANKER_NO    NUMBER (10),
  RANKER_NAME  VARCHAR2 (50),
  INST_TABIA   NUMBER,
  INST_SACAN1  NUMBER,
```

```
  INST_SACAN2  NUMBER,
  TAHSEEN      NUMBER,
  TASKEEL      NUMBER,
  YEAR         NUMBER (10),
  TEH          NUMBER (12,2) ) ;
```

----------جدول اسماء البنوك--13------

```
CREATE TABLE TLK_BANKS (
  BANK_NO    NUMBER (10)   NOT NULL,
  BANK_NAME  VARCHAR2 (50),
  CONSTRAINT TLK_BANK_PK
  PRIMARY KEY ( BANK_NO ) ) ;
```

----------جدول اسماء الافرع---14-----

```
CREATE TABLE TLK_BANK_BRANCH (
  BANK_NO      NUMBER (10)   NOT NULL,
  BANK_B_NO    NUMBER (10)   NOT NULL,
  BANK_B_NAME  VARCHAR2 (30),
  LOC          NUMBER (1),
  CONSTRAINT TLK_BANK_BRANCH_PK
  PRIMARY KEY ( BANK_NO, BANK_B_NO ) ) ;

ALTER TABLE TLK_BANK_BRANCH ADD  CONSTRAINT TLK_BANK_BRANCH_FK
 FOREIGN KEY (BANK_NO)
  REFERENCES SAL2015.TLK_BANKS (BANK_NO) ;
```

----------جدول اسماء اوقيم بدل سكن وبدل ترحيل---15------

```
CREATE TABLE TLK_RANKER (
  RANKER_NO    NUMBER (10)   NOT NULL,
  RANKER_NAME  VARCHAR2 (50),
  INST_TABIA   NUMBER,
  INST_SACAN1  NUMBER,
  INST_SACAN2  NUMBER,
  TAHSEEN      NUMBER,
  TASKEEL      NUMBER,
  YEAR         NUMBER (10)   NOT NULL,
  TEH          NUMBER (12,2),
```

```
  CONSTRAINT PK_TLK_RANKER
  PRIMARY KEY ( YEAR, RANKER_NO ) ) ;
```

--------------جدول اضافة ضابط جديد-----16---

```
CREATE TABLE TBL_SAL_OFFICER_ADD (
  MILITARY_NO  NUMBER (10)   NOT NULL,
  MONTH        NUMBER (2),
  YEAR         NUMBER (4),
  SAL_KIND     NUMBER (1),
  BANK_NO      NUMBER (10),
  BANK_B_NO    NUMBER (10),
  ACC_NO       NUMBER (15),
  RANKER_NO    NUMBER (10),
  SECTOR_NO    NUMBER (10),
  SOCIAL       NUMBER (1),
  ADD_DATE     DATE,
  INPUT_DATE   DATE,
  US_ID        NUMBER (3),
  NAME          VARCHAR2 (50),
  CENTER_NO    NUMBER (10),
  TYPE         VARCHAR2 (20),
  REVIEW       NUMBER (3),
  TYP_NO       NUMBER (2),
  GROUP_NO     NUMBER (2),
  JOP_NO       NUMBER (2),
  REVIEW2      NUMBER (3),
  NO1          NUMBER (10),
  CASE_NOW     NUMBER (1)    NOT NULL,
  CONSTRAINT TBL_SAL_OFFICER_ADD_PK
  PRIMARY KEY ( MILITARY_NO ) ) ;
```

--------------جدول الحركات اليومية---17-----

```
CREATE TABLE TBL_SAL_OFFICER1 (
  MILITARY_NO   NUMBER (10),
  MONTH         NUMBER (2),
  YEAR          NUMBER (4),
  OBJECT        VARCHAR2 (30),
  BIAN          VARCHAR2 (150),
  USER_NAME     VARCHAR2 (40),
  REV           VARCHAR2 (40),
  ATMAD         VARCHAR2 (40),
  REM           VARCHAR2 (200),
```

```
  REP           NUMBER (1),
  TRANS_DATE    DATE,
  AMOUNT1       NUMBER (12,2),
  AMOUNT2       NUMBER (12,2),
  MILITARY_NO1  NUMBER (10) ) ;
```

--------------جدول خطاب استقطاعات صناديق الزمالة---18----

```
CREATE TABLE TBL_REDCUTION_DOC (
  REDC_NO      NUMBER (10)   NOT NULL,
  BANK_NO      NUMBER (10),
  BANK_B_NO    NUMBER (10),
  YEAR         NUMBER (4)    NOT NULL,
  MONTH        NUMBER (2)    NOT NULL,
  REDC_VALUE   NUMBER (32,2),
  ACC_NO       NUMBER (15),
  DOC_NO       NUMBER (3),
  CH           NUMBER (1),
  US_ID        NUMBER (3),
  REV          NUMBER (3),
  REDC_VALUE2  NUMBER (32,2),
  REDC_VALUE3  NUMBER (32,2),
  COUNTT       NUMBER (3),
  CONSTRAINT TBL_REDCUTION_DOC_PK
  PRIMARY KEY ( YEAR, MONTH, REDC_NO ) ) ;
```

--------------جدول اضافة/حذف استحقاق/استقطاع---19-----

```
CREATE TABLE TBL_SAL_OFFICER_ADD_DEL_MERT (
  MILITARY_NO  NUMBER (10),
  MONTH        NUMBER (2),
  RANKER_NO    NUMBER (10),
  MERIT_NO     NUMBER (10),
  YEAR         NUMBER (4),
  CHANG_DATE   DATE,
  INPUT_DATE   DATE,
  OPERATION    NUMBER (1),
  OP_TYPE      NUMBER (1),
  CH           NUMBER (1),
  REVIEW       NUMBER (3),
  US_ID        NUMBER (3),
  REVIEW2      NUMBER (3),
  T            NUMBER (12,2),
  NO1          NUMBER (10),
```

```
  CASE_NOW     NUMBER (1)    NOT NULL ) ;
--------------جدول حذف ضابط------20-
CREATE TABLE TBL_SAL_OFFICER_DEL (
  MILITARY_NO  NUMBER (10),
  MONTH        NUMBER (2)    NOT NULL,
  YEAR         NUMBER (4)    NOT NULL,
  DEL_DATE     DATE,
  INPUT_DATE   DATE,
  US_ID        NUMBER (3),
  ASRA         VARCHAR2 (40),
  ASHRA_DATE   DATE,
  REASON       VARCHAR2 (50),
  WHO          VARCHAR2 (50),
  NOTE         VARCHAR2 (100),
  REVIEW       NUMBER (3),
  REVIEW1      NUMBER (3),
  REVIEW2      NUMBER (3),
  NO1          NUMBER (10)   NOT NULL,
  CASE_NOW     NUMBER (1)    NOT NULL,
  TYPES        NUMBER (1),
  MONY_TAK     NUMBER (12,2),
  YEAR1        NUMBER (4),
  MONTH1       NUMBER (2),
  CONSTRAINT OFF_DEL_PK
  PRIMARY KEY ( NO1, MONTH, YEAR ) ) ;
--------------جدول الاوسمة-----21-
CREATE TABLE TBL_SAL_OFFICER_OSMA (
  MILITARY_NO  NUMBER (10)   NOT NULL,
  OSMA_NO      NUMBER (3)    NOT NULL,
  OSMA_NAME    VARCHAR2 (50),
  OSMA_DATE    DATE,
  US_ID        NUMBER (3),
  REVEW        NUMBER (3),
  INPUT_DATE   DATE,
  CONSTRAINT PK_BASIC_OFFICER_OSMA
  PRIMARY KEY ( MILITARY_NO, OSMA_NO ) ) ;

ALTER TABLE TBL_SAL_OFFICER_OSMA ADD  CONSTRAINT FK_BASIC_OFFICER_OSMA
 FOREIGN KEY (MILITARY_NO)
  REFERENCES SAL2015.TBL_SAL_OFFICER_BANK (MILITARY_NO) ;
```

```
--------------جدول فك المرتب-----22---
CREATE TABLE TBL_SAL_OFFICER_RTURN (
  MILITARY_NO  NUMBER (10)   NOT NULL,
  MONTH        NUMBER (2)    NOT NULL,
  YEAR         NUMBER (4)    NOT NULL,
  RETURN_DATE  DATE,
  INPUT_DATE   DATE,
  US_ID        NUMBER (3),
  TYP_NO       NUMBER (2),
  GROUP_NO     NUMBER (2),
  ASHAR_NO     NUMBER (10),
  ASHAR_DATE   DATE,
  SOCIAL       NUMBER (1),
  JOP_NO       NUMBER (2),
  REVIEW       NUMBER (3),
  REVIEW2      NUMBER (3),
  NO1          NUMBER (10)   NOT NULL,
  CASE_NOW     NUMBER (1)    NOT NULL,
  CONSTRAINT OFFICER_RTURN_PK
  PRIMARY KEY ( NO1, MONTH, YEAR ) ) ;
--------------جدول اضافة استحقاق لاكثر من شهر---23-----
CREATE TABLE ADD_REDUC (
  MILITARY_NO  NUMBER (10),
  MONTH        NUMBER (2)    NOT NULL,
  YEAR         NUMBER (4)    NOT NULL,
  ADD_DATE     DATE,
  INPUT_DATE   DATE,
  MONTH1       NUMBER (2)    NOT NULL,
  YEAR1        NUMBER (4)    NOT NULL,
  AMOUNT       NUMBER (14,2),
  ASH_NO       VARCHAR2 (40),
  ASH_DATE     DATE,
  REASON       VARCHAR2 (40),
  WHO          VARCHAR2 (40),
  AATMAD       NUMBER (3),
  REDC_NO      NUMBER        NOT NULL,
  US_ID        NUMBER (3),
  NO1          NUMBER (10),
  CASE_NOW     NUMBER (1)    NOT NULL,
  CENTER_NO    NUMBER (10),
```

```
  CH            NUMBER (1) ) ;
--------------جدول المستندات------24--
CREATE TABLE BASIC_OFFICER_DOC (
  MILITARY_NO       NUMBER (10)    NOT NULL,
  DOC_NO            NUMBER (3)     NOT NULL,
  DOC_NAME          VARCHAR2 (50),
  DOC_DATE          DATE,
  DATE_IN_ARSHIF    DATE,
  US_ID_IN_ARSHIF   NUMBER (3),
  US_ID             NUMBER (3),
  REVEW             NUMBER (3),
  INPUT_DATE        DATE,
  DOC2_NO           VARCHAR2 (20),
  DOC2_NAME         VARCHAR2 (30),
  DOC2_DATE         DATE,
  CONSTRAINT PK_BASIC_OFFICER_DOC
  PRIMARY KEY ( MILITARY_NO, DOC_NO ) ) ;

ALTER TABLE BASIC_OFFICER_DOC ADD  CONSTRAINT FK_BASIC_OFFICER_DOC
 FOREIGN KEY (MILITARY_NO)
  REFERENCES SAL2015.TBL_SAL_OFFICER_BANK (MILITARY_NO) ;
--------------جدول تعديل القطاع----25----
CREATE TABLE CHANG_RED (
  REDC_NO      NUMBER (10),
  YEAR         NUMBER (4),
  MONTH        NUMBER (2),
  YEAR_NEW     NUMBER (4),
  MONTH_NEW    NUMBER (2),
  US_ID        NUMBER (3),
  REV_ID       NUMBER (3),
  RED_COUNT    NUMBER (2),
  RED_VALUE    NUMBER (32,2),
  OLD_VALUE    NUMBER (32,2),
  NEW_VALUE    NUMBER (32,2),
  CASE_NOW     NUMBER (1)     NOT NULL,
  CHANG_DATE   DATE ) ;
--------------جدول انواع القوات----26---
CREATE TABLE MIL_GROUP (
  TYP_NO       NUMBER (2)     NOT NULL,
  GROUP_NO     NUMBER (2)     NOT NULL,
```

```
  GROUP_NAME  VARCHAR2 (150),
  CONSTRAINT MIL_GROUP_PK
  PRIMARY KEY ( TYP_NO, GROUP_NO ) ) ;

ALTER TABLE MIL_GROUP ADD  CONSTRAINT MIL_GROUP_FK
 FOREIGN KEY (TYP_NO)
  REFERENCES SAL2015.MIL_TYPE (TYP_NO) ;
```

----------جدول تعديل القطاع والرتبة--27------

```
CREATE TABLE RANK_SECT_MDFIC (
  MILITARY_NO    NUMBER (10),
  MONTH          NUMBER (2),
  YEAR           NUMBER (4),
  RANKER_NO      NUMBER (2)    NOT NULL,
  SECTOR_NO      NUMBER (10)   NOT NULL,
  CHANG_DATE     DATE,
  INPUT_DATE     DATE,
  US_ID          NUMBER (3),
  CH             NUMBER (1),
  REVIEW         NUMBER (3),
  REVIEW2        NUMBER (3),
  NO1            NUMBER (10)   NOT NULL,
  CASE_NOW       NUMBER (1)    NOT NULL,
  RANKER_NO_OLD  NUMBER (2),
  SECTOR_NO_OLD  NUMBER (10),
  CONSTRAINT RANK_SECT_MDFIC_PK
  PRIMARY KEY ( NO1, RANKER_NO, SECTOR_NO ) ) ;
```

----------جدول الاستقطاع لعدة شهور----28----

```
CREATE TABLE ADD_REDUC_UPDATE (
  MILITARY_NO  NUMBER (10),
  MONTH        NUMBER (2)    NOT NULL,
  YEAR         NUMBER (4)    NOT NULL,
  ADD_DATE     DATE,
  INPUT_DATE   DATE,
  MONTH1       NUMBER (2)    NOT NULL,
  YEAR1        NUMBER (4)    NOT NULL,
  AMOUNT       NUMBER (14,2),
  ASH_NO       VARCHAR2 (40),
  ASH_DATE     DATE,
  REASON       VARCHAR2 (40),
```

```
  WHO           VARCHAR2 (40),
  AATMAD        NUMBER (3),
  REDC_NO       NUMBER        NOT NULL,
  US_ID         NUMBER (3),
  NO1           NUMBER (10),
  CASE_NOW      NUMBER (1)    NOT NULL,
  CENTER_NO     NUMBER (10),
  MONTH2        NUMBER (2),
  YEAR2         NUMBER (5),
  AMOUNT2       NUMBER (14,2),
  MONTH3        NUMBER (2),
  YEAR3         NUMBER (2) ) ;
```

----------جدول تغير نوع القوات---29------

```
CREATE TABLE CHANG_MIL_TYPE (
  MILITARY_NO   NUMBER (10)   NOT NULL,
  MONTH         NUMBER (2)    NOT NULL,
  YEAR          NUMBER (4)    NOT NULL,
  JOP_NO        NUMBER (2),
  CHANGE_DATE   DATE,
  INPUT_DATE    DATE,
  US_ID         NUMBER (3),
  REVIEW        NUMBER (3),
  TYP_NO        NUMBER (2)    NOT NULL,
  GROUP_NO      NUMBER (2)    NOT NULL,
  MERIT_NO      NUMBER (10),
  RANKER_NO     NUMBER (10),
  NO1           NUMBER (10),
  CASE_NOW      NUMBER (1)    NOT NULL,
  TYP_NO_OLD    NUMBER (2),
  GROUP_NO_OLD  NUMBER (2),
  CONSTRAINT CHANG_MIL_TYPE_PK
  PRIMARY KEY ( MILITARY_NO, YEAR, MONTH, TYP_NO, GROUP_NO ) ) ;
```

----------جدوطبيعة عمل----30-----

```
CREATE TABLE MIL_WORK (
  TYP_NO     NUMBER (2)    NOT NULL,
  GROUP_NO   NUMBER (2)    NOT NULL,
  MERIT_NO   NUMBER (10)   NOT NULL,
  RANKER_NO  NUMBER (10)   NOT NULL,
  AMOUNT     NUMBER (12,2),
  FRG        NUMBER (12,2),
```

```
  CONSTRAINT MIL_WORK_PK
  PRIMARY KEY ( TYP_NO, GROUP_NO, MERIT_NO, RANKER_NO ) ) ;

ALTER TABLE MIL_WORK ADD  CONSTRAINT MIL_WORK_FK2
 FOREIGN KEY (MERIT_NO)
  REFERENCES SAL2015.MERIT (MERIT_NO) ;

ALTER TABLE MIL_WORK ADD  CONSTRAINT MIL_WORK_FK3
 FOREIGN KEY (TYP_NO, GROUP_NO)
  REFERENCES SAL2015.MIL_GROUP (TYP_NO, GROUP_NO) ;

ALTER TABLE MIL_WORK ADD  CONSTRAINT MIL_WORK_FK1
 FOREIGN KEY (RANKER_NO)
  REFERENCES SAL2015.RANKER (RANKER_NO) ;


--------------جدول اعداد المرتبات----31----
CREATE TABLE MAS_SALNEW (
  DOC_NO      NUMBER (6),
  DOC_DATE    DATE,
  MONTH       NUMBER (2)    NOT NULL,
  YEAR        NUMBER (4)    NOT NULL,
  BANK_NO     NUMBER (10)   NOT NULL,
  BANK_NO_B   NUMBER (10),
  LOC         NUMBER (1),
  AMOUNT      NUMBER (12,2),
  REMARK      VARCHAR2 (150),
  USERID      NUMBER (3),
  AATMAD      NUMBER (3),
  REVEW       NUMBER (3),
  TYPE        VARCHAR2 (20)  NOT NULL,
  CONSTRAINT MAS_SALNEW_PK
  PRIMARY KEY ( YEAR, MONTH, BANK_NO, TYPE ) ) ;


--------------جدول الملفات------32--
CREATE TABLE MOV_FILE (
  MILITARY_NO        NUMBER (10),
  MOV_TO_SECTION     VARCHAR2 (20),
  MOV_FROM_SECTION   VARCHAR2 (20),
  MOV_DATE           DATE,
```

```
  TAKE_TO           VARCHAR2 (30),
  TAKE_FROM         VARCHAR2 (30),
  REASON            VARCHAR2 (50),
  US_ID             NUMBER (3),
  REVEW             NUMBER (3) ) ;

ALTER TABLE MOV_FILE ADD  CONSTRAINT FK_MOV_FILE
 FOREIGN KEY (MILITARY_NO)
  REFERENCES SAL2015.TBL_SAL_OFFICER_BANK (MILITARY_NO)  DISABLE ;
```

--------------جدول الرتب--------33

```
CREATE TABLE RANKER (
  RANKER_NO    NUMBER (10)   NOT NULL,
  RANKER_NAME  VARCHAR2 (50),
  CONSTRAINT RANKER_PK
  PRIMARY KEY ( RANKER_NO ) ) ;
```

--------------جدول المرتبات للبنوك--34------

```
CREATE TABLE SAL_BANK1 (
  DOC_NO     NUMBER (6),
  MONTH      NUMBER (2)    NOT NULL,
  YEAR       NUMBER (4)    NOT NULL,
  BANK_NO    NUMBER (10)   NOT NULL,
  BANK_NO_B  NUMBER (10)   NOT NULL,
  AMOUNT     NUMBER (12,2),
  REMARK     VARCHAR2 (150),
  TYPE       VARCHAR2 (20)  NOT NULL,
  CH_NO      NUMBER (15),
  CONSTRAINT SAL_BANK1_PK
  PRIMARY KEY ( YEAR, MONTH, BANK_NO, TYPE, BANK_NO_B ) ) ;

ALTER TABLE SAL_BANK1 ADD  CONSTRAINT SAL_BANK1_FK
 FOREIGN KEY (YEAR, MONTH, BANK_NO, TYPE)
  REFERENCES SAL2015.MAS_SALNEW (YEAR, MONTH, BANK_NO, TYPE) ;
```

--------------جدول تعديل الحالة الاجتماعية--35------

```
CREATE TABLE SOCIAL_MDFIC (
  MILITARY_NO  NUMBER (10),
  MONTH        NUMBER (2)    NOT NULL,
  YEAR         NUMBER (4)    NOT NULL,
  SOCIAL       NUMBER (1)    NOT NULL,
```

```
  CHANG_DATE    DATE,
  INPUT_DATE    DATE,
  CH            NUMBER (1),
  REVIEW        NUMBER (3),
  US_ID         NUMBER (3),
  REVIEW2       NUMBER (3),
  OLD           NUMBER (10),
  NO1           NUMBER (10)    NOT NULL,
  CASE_NOW      NUMBER (1)     NOT NULL,
  CH1_NAME      VARCHAR2 (30),
  CH2_NAME      VARCHAR2 (30),
  CH3_NAME      VARCHAR2 (30),
  DATE_CH1      DATE,
  DATE_CH2      DATE,
  DATE_CH3      DATE,
  CONSTRAINT SOCIAL_MDFIC_PK
  PRIMARY KEY ( NO1, SOCIAL, YEAR, MONTH ) ) ;


--------------جدولتعديل الاسم--36------

CREATE TABLE CHANGE_NAME (
  MILITARY_NO   NUMBER (10),
  OLD_NAME      VARCHAR2 (70),
  NEW_NAME      VARCHAR2 (70),
  MONTH         NUMBER (2),
  YEAR          NUMBER (4),
  CASE_NOW      NUMBER (1),
  US_ID         NUMBER (3),
  REVIEW        NUMBER (3),
  REVIEW2       NUMBER (3),
  IN_DATEA      DATE,
  RE_DATEA      DATE,
  REM           VARCHAR2 (100) ) ;

ALTER TABLE CHANGE_NAME ADD  CONSTRAINT CHANGE_NAME_FK
 FOREIGN KEY (MILITARY_NO)
  REFERENCES SAL2015.TBL_SAL_OFFICER_BANK (MILITARY_NO) ;

--------------جدول تغير رقم الحساب والبنك---37----

CREATE TABLE BANK_ACC_MDFIC (
  MILITARY_NO   NUMBER (10),
  MONTH         NUMBER (2)     NOT NULL,
```

```
  YEAR          NUMBER (4)    NOT NULL,
  BANK_NO       NUMBER (10)   NOT NULL,
  BANK_B_NO     NUMBER (10)   NOT NULL,
  ACC_NO        NUMBER (15)   NOT NULL,
  CHANG_DATE    DATE,
  INPUT_DATE    DATE,
  US_ID         NUMBER (3),
  CH            NUMBER (1),
  REVIEW        NUMBER (3),
  REVIEW2       NUMBER (3),
  NO1           NUMBER (10)   NOT NULL,
  CASE_NOW      NUMBER (1)    NOT NULL,
  BANK_NO1      NUMBER (4),
  BANK_B_NO1    NUMBER (10),
  ACC_NO1       NUMBER (15),
  CONSTRAINT BANK_ACC_MDFIC_PK
  PRIMARY KEY ( NO1, YEAR, MONTH, BANK_NO, BANK_B_NO, ACC_NO ) ) ;
```

----------جدول تغير الوحدات-----38--

```
CREATE TABLE CHANG_UNIT (
  MILITARY_NO    NUMBER (10),
  MONTH          NUMBER (2)    NOT NULL,
  YEAR           NUMBER (4)    NOT NULL,
  CENTER_NO_OLD  NUMBER (10),
  CENTER_NO_NEW  NUMBER (10),
  CHANG_DATE     DATE,
  REVIEW         NUMBER (3),
  US_ID          NUMBER (3),
  CASE_NOW       NUMBER (1) ) ;
```

----------جدولطبيعة عمل المجموعات--39------

```
CREATE TABLE MIL_GROUP_JOB (
  TYP_NO    NUMBER (2)    NOT NULL,
  GROUP_NO  NUMBER (2)    NOT NULL,
  JOP_NO    NUMBER (2)    NOT NULL,
  JOB_NAME  VARCHAR2 (50),
  CONSTRAINT MIL_GROUP_JOB_PK
  PRIMARY KEY ( TYP_NO, GROUP_NO, JOP_NO ) ) ;

ALTER TABLE MIL_GROUP_JOB ADD  CONSTRAINT MIL_GROUP_JOB_FK
 FOREIGN KEY (TYP_NO, GROUP_NO)
```

```
  REFERENCES SAL2015.MIL_GROUP (TYP_NO, GROUP_NO) ;
```

--------------جدول الفروقات----40----

```
CREATE TABLE TBL_DIFFERENCE (
  YEAR           NUMBER        NOT NULL,
  MONTH          NUMBER        NOT NULL,
  MILITARY_NO    NUMBER        NOT NULL,
  PAY_DATE       DATE,
  AMOUNT         NUMBER (32,2) NOT NULL,
  REMARK         VARCHAR2 (200),
  US_ID          NUMBER (3),
  REV            NUMBER (3),
  ATMMAD         NUMBER (3),
  DOC_NO         NUMBER (4),
  DOC_DATE       DATE,
  BANK_NO        NUMBER (10),
  BANK_B_NO      NUMBER (10),
  ACC_NO         NUMBER (15),
  CH             NUMBER (1),
  REV_R          NUMBER (3),
  ATMMAD_R       NUMBER (3),
  REV_REASON     VARCHAR2 (50),
  ATMMAD_REASON  VARCHAR2 (50),
  NO1            NUMBER (10),
  CASE_NOW       NUMBER (1)    NOT NULL,
  CH_NO          NUMBER (20),
  CONSTRAINT TBL_DIFFERENCE_PK
  PRIMARY KEY ( YEAR, MONTH, MILITARY_NO ) ) ;
```

--------------جدول تفاصيل الفروقات----41---

```
CREATE TABLE TBL_DIFFERENCE_DET (
  YEAR         NUMBER        NOT NULL,
  MONTH        NUMBER        NOT NULL,
  MILITARY_NO  NUMBER        NOT NULL,
  BIAN         VARCHAR2 (30)  NOT NULL,
  AMOUNT       NUMBER (12,2),
  CONSTRAINT TBL_DIFFERENCE_DET_PK
  PRIMARY KEY ( YEAR, MONTH, MILITARY_NO, BIAN ) ) ;

ALTER TABLE TBL_DIFFERENCE_DET ADD  CONSTRAINT TBL_DIFFERENCE_DET_FK
 FOREIGN KEY (YEAR, MONTH, MILITARY_NO)
```

```
  REFERENCES SAL2015.TBL_DIFFERENCE (YEAR, MONTH, MILITARY_NO) ;


--------------جدول تسعة شهور-----42---
CREATE TABLE TBL_PEN_MERIT (
  MERIT_NO     NUMBER (10)   NOT NULL,
  INST_TYPE    NUMBER        NOT NULL,
  MILITARY_NO  NUMBER (10)   NOT NULL,
  MERIT_VALUE  NUMBER (22,2),
  YEAR         NUMBER (4)    NOT NULL,
  CONSTRAINT TBL_PEN_MERIT_PK
  PRIMARY KEY ( MERIT_NO, MILITARY_NO, INST_TYPE, YEAR ) ) ;
--------------جدول الخدمة الطويلة الممتازة----43----
CREATE TABLE TBL_SAL_OFFICER1 (
  MILITARY_NO   NUMBER (10),
  MONTH         NUMBER (2),
  YEAR          NUMBER (4),
  OBJECT        VARCHAR2 (30),
  BIAN          VARCHAR2 (150),
  USER_NAME     VARCHAR2 (40),
  REV           VARCHAR2 (40),
  ATMAD         VARCHAR2 (40),
  REM           VARCHAR2 (200),
  REP           NUMBER (1),
  TRANS_DATE    DATE,
  AMOUNT1       NUMBER (12,2),
  AMOUNT2       NUMBER (12,2),
  MILITARY_NO1  NUMBER (10) ) ;
--------------جدول البديل النقدى----44----
CREATE TABLE TBL_PENSION_INST_OFFICER (
  RECIVRE_VALUE  NUMBER (12,2),
  INST_TYPE      NUMBER        NOT NULL,
  REDC_COUNT     NUMBER (10),
  MILITARY_NO    NUMBER (10)   NOT NULL,
  INST_AMOUNT    NUMBER,
  YEAR           NUMBER (4)    NOT NULL,
  TYPE           NUMBER (2),
  DATE_MONEY     DATE,
  SAFE_MONEY     NUMBER (12,2),
  SAL_MONTH      NUMBER,
```

```
  TAH_AMOUNT      NUMBER,
  FINISH_AMOUNT   NUMBER,
  MONTH_COUNT     NUMBER,
  RANKER_NO       NUMBER,
  SEC_NO          NUMBER,
  SOCIAL_STATE    NUMBER,
  BANK_NO         NUMBER (10),
  BANK_B_NO       NUMBER (10),
  US_ID           NUMBER (3),
  REVEW           NUMBER (3),
  INPUT_DATE      DATE,
  CH              NUMBER (1),
  OLOYA           NUMBER (1),
  REMARK          VARCHAR2 (100),
  REASON          VARCHAR2 (100),
  ACC_NO          NUMBER (15),
  NO1             NUMBER (10)   NOT NULL,
  CASE_NOW        NUMBER (1),
  SAFE_MONEY2     NUMBER (12,2),
  CONSTRAINT TBL_PENSION_INST_OFFICER_PK
  PRIMARY KEY ( INST_TYPE, MILITARY_NO, YEAR ) ) ;
```

----------جدول البديل النقدى الذين صرفوا----45----

```
CREATE TABLE TBL_PENSION_INST_OFFICER_TAKE (
  RECIVRE_VALUE   NUMBER (12,2),
  MILITARY_NO     NUMBER (10)   NOT NULL,
  YEAR            NUMBER (4)    NOT NULL,
  BANK_NO         NUMBER (10),
  BANK_B_NO       NUMBER (10),
  DOC_NO          NUMBER (5),
  RANKER_NO       NUMBER (10),
  FRG             NUMBER (10),
  DOC2            NUMBER (5),
  REV             NUMBER (3),
  REV_DATE        DATE,
  MORAGA          NUMBER (3),
  CASE_NOW        NUMBER (1),
  NO1             NUMBER (10),
  DOC_DATE        DATE,
  CONSTRAINT OFFICER_TAKE_PK
  PRIMARY KEY ( MILITARY_NO, YEAR ) ) ;
```

---------- فروقات جدول البديل النقدى ----46----

```
CREATE TABLE TBL_PENSION_INST_OFFICER2 (
  RECIVRE_VALUE  NUMBER (12,2),
  INST_TYPE      NUMBER        NOT NULL,
  REDC_COUNT     NUMBER (10),
  MILITARY_NO    NUMBER (10)   NOT NULL,
  INST_AMOUNT    NUMBER,
  YEAR           NUMBER (4)    NOT NULL,
  TYPE           NUMBER (2),
  DATE_MONEY     DATE,
  SAFE_MONEY     NUMBER (12,2),
  SAL_MONTH      NUMBER,
  TAH_AMOUNT     NUMBER,
  FINISH_AMOUNT  NUMBER,
  MONTH_COUNT    NUMBER,
  RANKER_NO      NUMBER,
  SEC_NO         NUMBER,
  SOCIAL_STATE   NUMBER,
  BANK_NO        NUMBER (10),
  BANK_B_NO      NUMBER (10),
  US_ID          NUMBER (3),
  REVEW          NUMBER (3),
  INPUT_DATE     DATE,
  CH             NUMBER (1),
  OLOYA          NUMBER (1),
  REMARK         VARCHAR2 (100),
  REASON         VARCHAR2 (100),
  GRAR_NO        NUMBER (10),
  GRAR_DATE      DATE,
  INST_TABIA     NUMBER,
  INST_SACAN     NUMBER,
  AHLA_DATE      DATE,
  LASTSAL        NUMBER (12,2),
  DAYS           NUMBER (2),
  BDIAL_NOW      NUMBER (12,2),
  DOC_NO         NUMBER (10),
  DOC_DATE       DATE,
  TOTAL          NUMBER (10,2),
  STUTES         NUMBER (10),
  CONSTRAINT TBL_PENSION_INST_OFFICER2_PK
```

```
  PRIMARY KEY ( INST_TYPE, MILITARY_NO, YEAR ) ) ;
```

----------جدولفروقات البديل النقدى الذين صرفوا----47----

```
CREATE TABLE TBL_PENSION_INST_OFFICER_TAKE2 (
  RECIVRE_VALUE  NUMBER (12,2),
  MILITARY_NO    NUMBER (10)   NOT NULL,
  YEAR           NUMBER (4)    NOT NULL,
  BANK_NO        NUMBER (10),
  BANK_B_NO      NUMBER (10),
  DOC_NO         NUMBER (5),
  RANKER_NO      NUMBER,
  FRG            NUMBER (10),
  DOC2           NUMBER (5),
  REV            NUMBER (3),
  REV_DATE       DATE,
  MORAGA         NUMBER (3),
  CASE_NOW       NUMBER (1),
  NO1            NUMBER (10),
  DOC_DATE       DATE,
  TYPES          NUMBER        NOT NULL,
  CONSTRAINT TBL_PENSION_TAKE2_PK
  PRIMARY KEY ( MILITARY_NO, TYPES, YEAR ) ) ;
```

----------جدول تقارير الفروقات متابعه----48----

```
CREATE TABLE DOC_REP (
  DOC_NO    NUMBER (5),
  REP_DATE  DATE,
  N         NUMBER,
  YEAR      NUMBER ) ;
```

----------كودتحويل الرقم الى حرفى-----

```
PROCEDURE HDATE_YEAR (H_DATE IN VARCHAR2,H_YEAR OUT NUMBER) IS

BEGIN SELECT TO_NUMBER(DECODE(LENGTH(DECODE(LENGTH(SUBSTR(H_DATE,
DECODE(INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0,
```

```
INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1),

0, INSTR(H_DATE, '-', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-
'),

INSTR(H_DATE, '/'))+1),

INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),
INSTR(H_DATE, '/'))+1))+1,

4)),1, '0'||SUBSTR(H_DATE, DECODE(INSTR(H_DATE, '/',
DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),

INSTR(H_DATE, '/'))+1),

0, INSTR(H_DATE, '-', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-
'),

INSTR(H_DATE, '/'))+1),

INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),

INSTR(H_DATE, '/'))+1))+1, 4),

SUBSTR(H_DATE, DECODE(INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0,

INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1),

0, INSTR(H_DATE, '-', DECODE(INSTR(H_DATE, '/'), 0,

INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1),

INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0,

INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1))+1, 4))),

4, DECODE(LENGTH(SUBSTR(H_DATE,

DECODE(INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE,
'-'), INSTR(H_DATE, '/'))+1),
```

```
0,INSTR(H_DATE, '-', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),
INSTR(H_DATE, '/'))+1),

INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),
INSTR(H_DATE, '/'))+1))+1, 4)),

1, '0'||SUBSTR(H_DATE, DECODE(INSTR(H_DATE, '/', DECODE(INSTR(H_DATE,
'/'), 0, INSTR(H_DATE, '-'),

INSTR(H_DATE, '/'))+1),

0, INSTR(H_DATE, '-', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-
'),

INSTR(H_DATE, '/'))+1),

INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),

INSTR(H_DATE, '/'))+1))+1, 4),

SUBSTR(H_DATE, DECODE(INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0,

INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1),

0, INSTR(H_DATE, '-', DECODE(INSTR(H_DATE, '/'), 0,

INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1),

INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0,

INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1))+1, 4)),

2, DECODE(LENGTH(SUBSTR(H_DATE, 1,

DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))-
1)), 1,

'0'||SUBSTR(H_DATE, 1,

DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))-1),
```

```
2, SUBSTR(H_DATE, 1,

DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))-1),

SUBSTR(H_DATE, 1,

DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))-1)),

DECODE(LENGTH(SUBSTR(H_DATE,

DECODE(INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE,
'-'), INSTR(H_DATE, '/'))+1),

0,INSTR(H_DATE, '-', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),
INSTR(H_DATE, '/'))+1),

INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),
INSTR(H_DATE, '/'))+1))+1, 4)),

1, '0'||SUBSTR(H_DATE, DECODE(INSTR(H_DATE, '/', DECODE(INSTR(H_DATE,
'/'), 0, INSTR(H_DATE, '-'),

INSTR(H_DATE, '/'))+1),

0, INSTR(H_DATE, '-', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),

INSTR(H_DATE, '/'))+1),

INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'), 0, INSTR(H_DATE, '-'),

INSTR(H_DATE, '/'))+1))+1, 4),

SUBSTR(H_DATE, DECODE(INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'),0,

INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1),

0, INSTR(H_DATE, '-', DECODE(INSTR(H_DATE, '/'),0,

INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1),

INSTR(H_DATE, '/', DECODE(INSTR(H_DATE, '/'),0,
```

```
INSTR(H_DATE, '-'), INSTR(H_DATE, '/'))+1))+1,

4)))) INTO H_YEAR FROM DUAL; END;
```

دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام هندسة البرمجيات في تطوير وتقويم الأنظمة المالية : دراسة حالة الإدارة العامة للشئون المالية بوزارة الدفاع
المؤلف الرئيسي:	التوم، ياسر محمد محمد أحمد
مؤلفين آخرين:	فتوح، سيف الدين(مشرف)
التاريخ الميلادي:	2015
موقع:	الخرطوم
الصفحات:	1 - 226
رقم MD:	831704
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة دكتوراه
الجامعة:	جامعة النيلين
الكلية:	كلية الدراسات العليا
الدولة:	السودان
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات
رابط:	https://search.mandumah.com/Record/831704

© 2019 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

بسم الله الرحمن الرحيم
جمهورية السودان
وزارة التعليم العالي والبحث العلمي
جامعة النيلين
كلية الدراسات العليا

استخدام هندسة البرمجيات فى تطوير وتقويم الانظمة المالية

Use of software engineering in the development And evaluation of financial systems

دراسة حالة/ الإدارة العامة للشئون المالية بوزارة الدفاع

بحث أعد لنيل درجة الدكتوراة فى علوم الحاسوب

اعداد:

ياسر محمد محمد احمد التوم

اشراف:

الدكتور/ سيف الدين فتوح

1436هـ - 2015م