



بسم الله الرحمن الرحيم

مقرر هندسة البرمجيات ١٤٩٤

الوحدة الثالثة

تحليل متطلبات البرمجيات Requirements Analysis

إعداد: م. هناء قشطة
الفصل الدراسي الأول
٢٠٢١م - ٢٠٢٢م



أهداف اللقاء

التعرف على أساسيات تحليل متطلبات البرمجيات.

التعرف على طريقة التحليل الهيكلي.

التعرف على طريقة التحليل الكينوني.

١ . مقدمة

تعد مرحلة تحليل متطلبات البرمجيات مرحلة مهمة من مراحل انتاج نظام برمجيات ودورة حياته، وتأتي هذه المرحلة بعد معرفة المتطلبات التي تطلبها الجهة المستفيدة من النظام المفترض انتاجه، كما تعتبر تمهيدا لبدء مرحلة تصميم البرمجيات.

٢. أساسيات تحليل المتطلبات

- عملية تحليل البرمجيات هي الخطوة الأولى من خطوات انجاز مواصفات البرمجيات حيث يتم تحديد المهام التي ستؤديها البرمجية المطلوبة.
- المشكلات الكبيرة والمعقدة تحتاج إلى تقنيات هندسة البرمجيات من أجل إيجاد حلول لها وخصوصاً إن كانت البرمجيات المطلوبة من نوع مستحدث.
- ان تحديد المهام التي سيؤديها النظام المقترح والقيود الموضوعية على تشغيله هي المواصفات المطلوبة ويجب أن تكون بصورة مكتوبة وموثقة وتعد هذه المواصفات الخطوة الأولى في الطريق.

٢. أساسيات تحليل المتطلبات

بعد تحديد موصفات النظام المطلوب و التي تعتبر هي المهام التي سيؤديها النظام، تعتبر عملية تحليل البرمجيات هي الخطوة الأولى من خطوات إنجاز هذه الموصفات.

بعض المؤسسات لا تستطيع تحديد موصفات المسألة بصورة دقيقة، وإنما تطلب فقط بناء نظام برمجي مثل نظام رواتب يقوم بحساب رواتب الموظفين، وفي حالة كهذه لا يمكن تحديد الموصفات بصورة ثابتة للأسباب التالية:

(١) يهدف النظام المقترح إلى تحسين وضع معين في المؤسسة، عن طريق بناء نظام

جديد، أو تحسين نظام سابق، وقد لا يستطيع المرء معرفة محاسن ومساوئ النظام.

(٢) الأنظمة الكبيرة يستخدمها قاعدة واسعة من المستخدمين، الذين قد تختلف متطلبات

عملهم، و لذلك لا يمكن تحديد رغبات الجميع.

(٣) الذين يتعاقدون لبناء النظام وصياغة متطلباته هم غير الذين يستخدمونه، فالمتعاقدون

يكونون من المستوى الإداري، بالتالي يكون النظام لا يحقق رغبات المستخدمين

الفعليين، خاصة مع القيود الموضوعة على كاهل الإدارة كالموازنة والوقت وعوامل

أخرى.

٢. أساسيات تحليل المتطلبات

يفضل أن تقدم المواصفات بمستويات مختلفة تبدأ بالصورة العامة وتتدرج نحو الصور التفصيلية حسب طبيعة الذين يتعاملون مع هذه المواصفات ومستواهم وتقسم هذه المستويات إلي:

الوصف العام للمتطلبات:

- عبارة عن صياغة لغوية تبين المهام التي يجب أن يؤديها النظام المقترح
- يجب ان تكون مفهومة من قبل الادارة والمستخدمين ومصممين النظام.

٢. أساسيات تحليل المتطلبات

الوصف الدقيق للمتطلبات:

هو وصف مكتوب يبين المهام التي يؤديها النظام المقترح بصورة دقيقة ومفصلة بحيث يمكن اعتمادها أساسا للتعاقد يجب أن تكتب بصورة يفهمها الفنيون من كلا الطرفين.

مواصفات البرمجيات (مواصفات التصميم):

عبارة عن وصف مختصر للبرمجيات والتي تعد أساس تصميم البرمجيات وتنفيذها

يجب أن تكون العلاقة بين مواصفات البرمجيات والوصف الدقيق للمتطلبات واضحة وسليمة ولكنها تكتب بلغة خاصة بمهندس تصميم البرمجيات.

٢. أساسيات تحليل المتطلبات

إن المواصفات الدقيقة لأنظمة البرمجيات قد لا يمكن تحديدها سلفاً، إلا بعد الشروع في بناء البرمجيات، و في هذه الحالة يكون اعتماد مفاهيم النموذج التجريبي Prototyping أفضل من نموذج الشلال Water-fall model، **ولكن إذا لم يعتمد استخدام النموذج التجريبي فيمكن اتباع الخطوات التالية:**

• دراسة الجدوى:

تقدير إمكانيات تحقيق رغبات المستخدمين، حسب التقنيات المتوفرة من برمجيات و مكونات مادية، و تقدير فاعلية النظام من النواحي الفنية و الاقتصادية و القيود الموضوعية.

٢. أساسيات تحليل المتطلبات

- تحليل المتطلبات:

هي عملية مراقبة الأنظمة الموجودة و المهام التي تؤديها، أو من خلال عقد اجتماعات مع المستخدمين و الإدارة، و تحديد المهام التي تؤديها المؤسسة، و طبيعة عمل المؤسسة.

- وصف المتطلبات:

وصف نموذج للنظام المقترح، و صياغة الوصف العام للمتطلبات منه. و هذا الوصف يفيد وجهة نظر المستخدم.

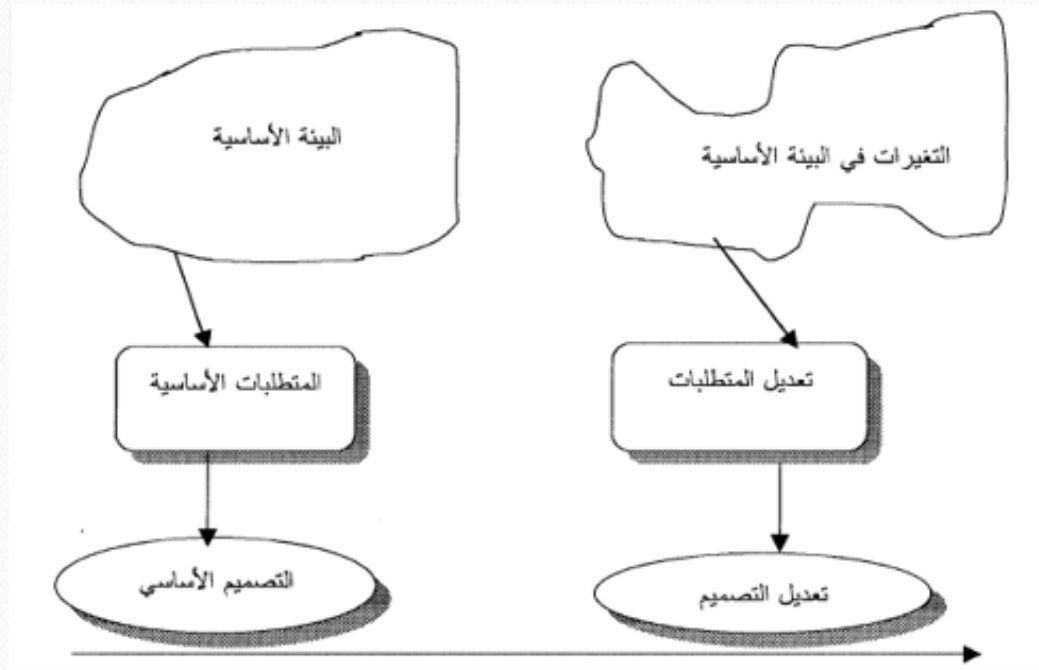
- الوصف الدقيق للمتطلبات:

يتضمن وصفًا دقيقًا للمتطلبات، و قد يتطلب مراجعة التحليل و الوصف العام، للوصول إلى حالة ثابتة، و هي الأساس يمكن التعاقد عليه.

٢. أساسيات تحليل المتطلبات

تطور متطلبات البرمجيات

تتطور متطلبات البرمجيات كلما أصبحت مهام البرمجيات أكثر وضوحاً واستيعاباً، وفي ضوء ذلك التطور يتم تعديل المتطلبات، وبالتالي تعديل التصميم الخاص بالنظام بناءً على المتطلبات الجديدة.

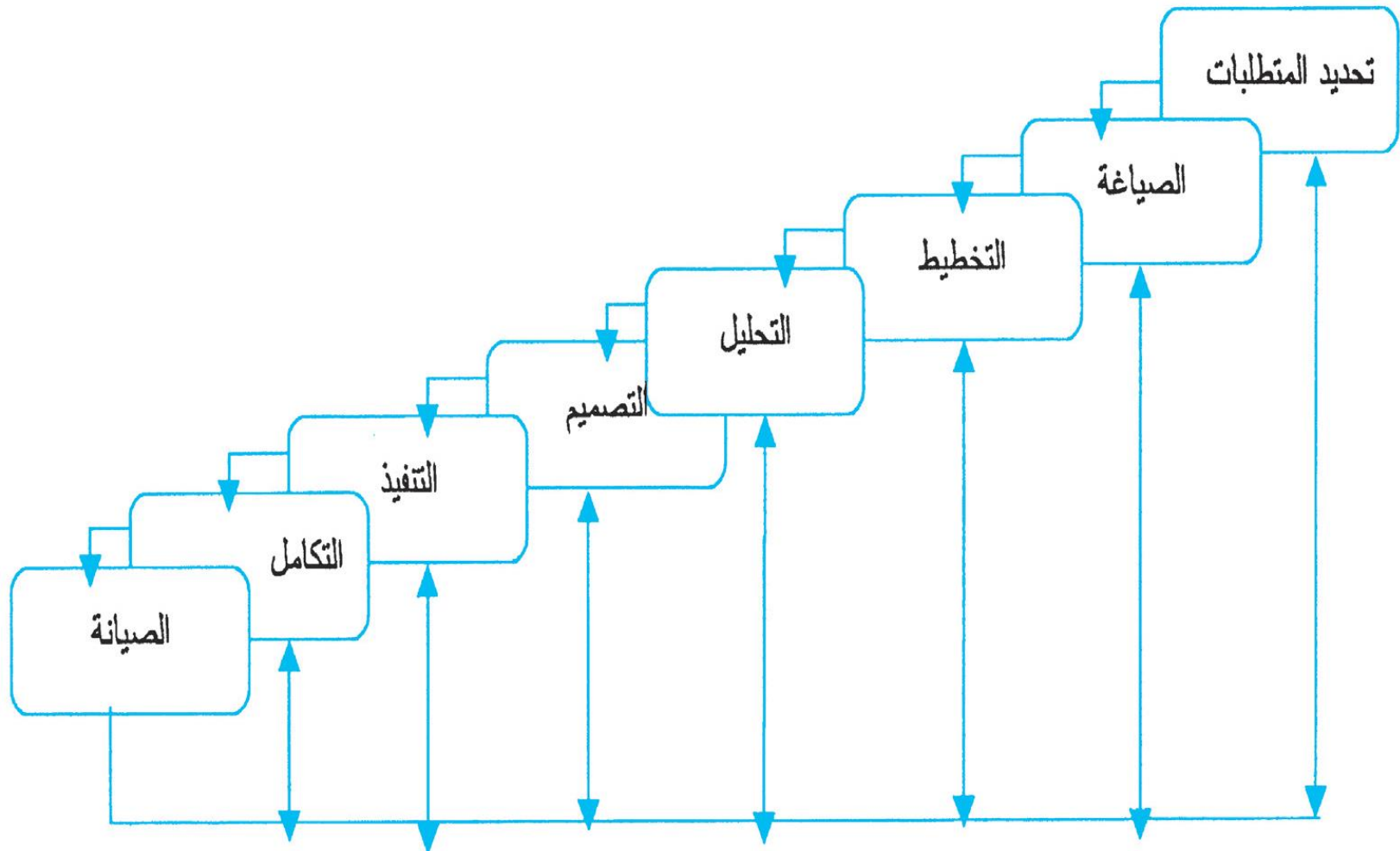


٢. أساسيات تحليل المتطلبات

نموذج الشلال أو النموذج السلمي Water-Fall Model

- هو نموذج يستخدم في تحليل المتطلبات، ويستخدم عادة في الأنظمة التي تحتاج إلى تحليل تفصيلي و دقيق، وإضافة ضوابط لكل مرحلة من مراحل بناء النظام.
- فهو نموذج لتطوير البرمجيات.
 - يتكون من عدة مراحل متسلسلة و متتالية و منفصلة عن بعضها البعض.
 - لا يمكن لمرحلة أن تبدأ قبل انتهاء المرحلة السابقة لها، ولا يمكن للمراحل أن تتداخل فيما بينها.
 - فيكون التقدم في سير العمل خلال المراحل على شكل خطي متدفق من أعلى إلى أسفل (مثل الشلال)، ويكون المخرج من المرحلة الحالية هو المدخل للمرحلة التالية.

٢. أساسيات تحليل المتطلبات



٢. أساسيات تحليل المتطلبات

يعتبر نموذج الشلال مفيد جدًا وملائم للكثير من أنواع البرمجيات مثل برمجيات المركبات الفضائية، والسيطرة الجوية، ومصافي النفط والغاز، وذلك يعود إلى أن هذه الأنواع من البرمجيات تتطلب تحليل دقيق للمسألة، وصياغة المواصفات بشكل مفصل (قبل الشروع في بناء النظام المطلوب)، وهذا ما يفرضه علينا نموذج الشلال خلال مرحلته.

٢. أساسيات تحليل المتطلبات

مساوئ نموذج الشلال كما يلي:

- تكاليف بناء البرمجيات عالية، و تستغرق وقتًا أطول.
- غير ملائم للبرمجيات التي تتسم بكثرة التعديلات و القرارات، فالنموذج السلمي لا يسمح بوجود الغموض، **حيث:**
 - يمكن أن تكون القرارات مطاطية فتؤدي إلى تغير المتطلبات بصورة مستمرة.
 - يمكن أن يكون متخذو القرار لا يستطيعون تحديد حاجاتهم في المرحلة الأولى، فيكونون بحاجة إلى تجريب النظام، و بناء عليه لاحقًا يتم تحديد القرارات.

٢. أساسيات تحليل المتطلبات

النموذج التجريبي Prototyping

هو نموذج يستخدم لتحليل متطلبات نظام برمجي سريع و غير مكلف، يسلم للمستخدم لإجراء تجارب عليه، و تزويد فريق التصميم بملاحظات الناتجة عن هذه التجارب، ليتم تحديد شروطه.

- يعد النظام التجريبي نسخة عملية من النظام المطلوب، أو جزءًا منه، و من خلال استخدام النظام التجريبي، تتضح الصورة أكثر للمستخدم، و يستطيع تحديد احتياجاته بصورة أدق.

- بعد ذلك يتم تحويل النظام التجريبي، أي إعادة بنائه، بما يتلاءم و احتياجات المطلوبة.
- إن عملية بناء تصميم أولي و تشغيله تحت التجربة، و إعادة تصحيحه و تعديله تسمى العملية التكرارية لتطوري البرمجيات Iterative.

٢. أساسيات تحليل المتطلبات

خطوات بناء النموذج التجريبي:

١. تحديد المتطلبات الأساسية للنظام

يقوم مصمم النظام بمناقشة المشروع أو النظام المقترح مع المستخدم من النظام، و تدوين المواصفات المطلوبة و الوظائف المتوقع أن يؤديها النظام.

٢. تطوير نظام تجريبي

بناء نظام يؤدي معظم المهام المطلوبة بأسرع وقت ممكن، باستخدام الإمكانيات و البرمجيات المتاحة. بعبارة أخرى تجهيز نظام يؤدي المهام بصورة مرحلية دون كتابة التفاصيل الكثيرة.

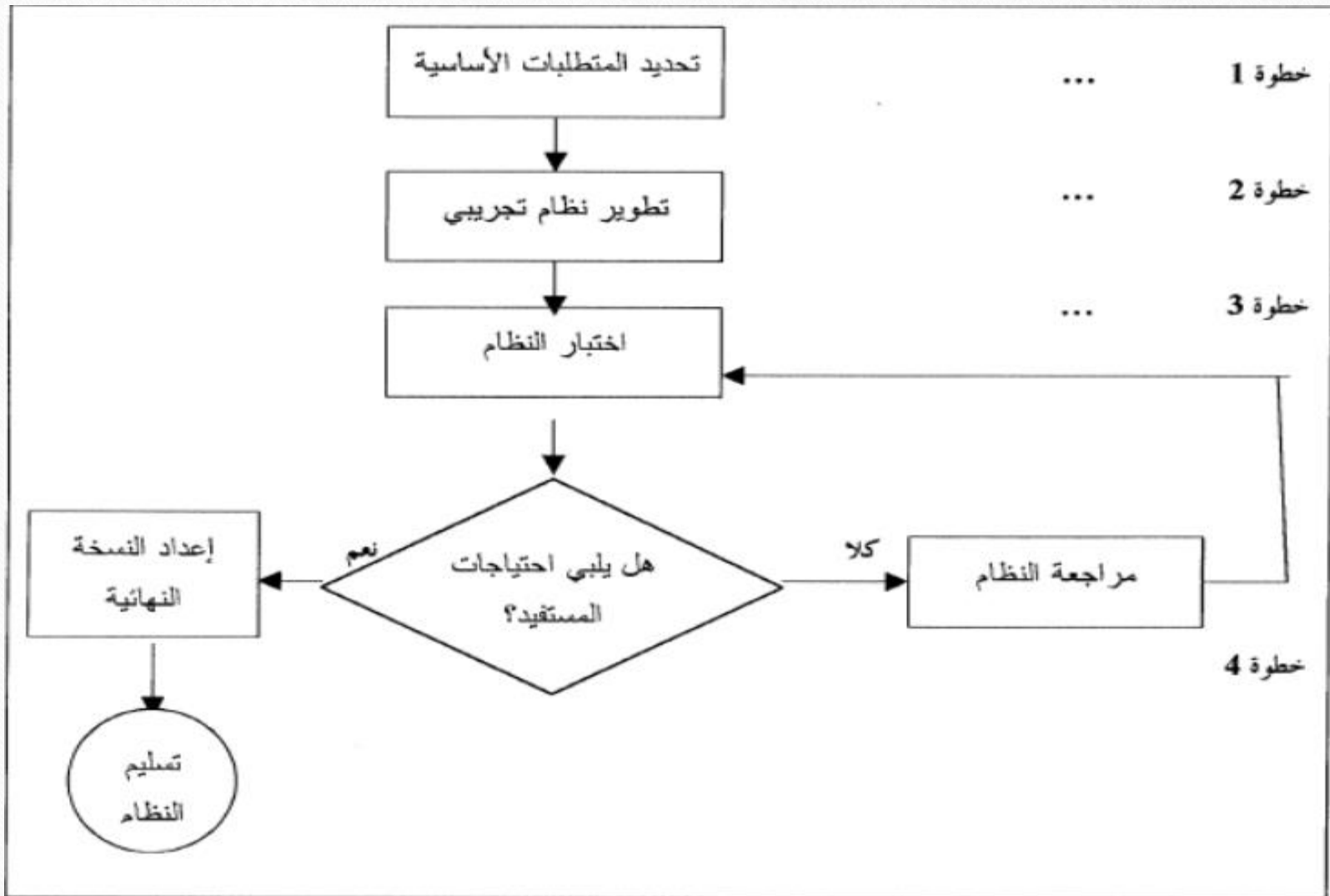
٢. أساسيات تحليل المتطلبات

٣. اختبار النظام و تسليمه إلى المستخدم لتنفيذ التجارب

هنا يستطيع المستخدم تقييم قوة النظام وتحديد مساوئه، واقتراح التعديلات المطلوبة، و اتخاذ القرار المناسب للمواصفات المطلوبة.

٤. مراجعة النظام و تعزيز بنائه

- بعد اختبار النظام والحصول على الملاحظات والتقييمات من المستخدم، تتم مراجعة النظام البرمجي وتعديله ليوافق احتياجات المستخدم.
- تتكرر الخطوتان ٣ و ٤ حتى يصبح النظام خاليًا من المشكلات، ويلبي مستلزمات الجهة المستفيدة. يبين الشكل التالي خطوات بناء النموذج التجريبي.



خطوات بناء النموذج التجريبي

٢. أساسيات تحليل المتطلبات

مزايا النموذج التجريبي:

- يفيد في بناء البرمجيات التي يكتنفها الغموض وصعوبة تحديد المتطلبات أو البرمجيات المتغيرة باستمرار او عندما يكون المستفيد غير مدرك ماذا يريد.
 - يفيد في بناء البرمجيات التي يصعب فيها تحديد المتطلبات.
 - يفيد في بناء البرمجيات المتغيرة باستمرار.
 - يفيد عندما يكون المستخدم غير مدرك ماذا يريد.
- من الميزات الأخرى حضور الجهة المستفيدة في كافة مراحل تطوير النظام ودورها الفعال في تحديد الاتجاهات المطلوبة، حيث يفيد ذلك في ضمان عدم إنتاج برمجيات ليست هي المقصودة.

٢. أساسيات تحليل المتطلبات

من مساوئ النموذج التجريبي:

- تنقصه الدقة في معالجة القضايا الجوهرية، مثل الدراسات الدقيقة و الشاملة لمرحلة التحليل و التوثيق المصاحب لكل مرحلة من مراحل التطوير.
- غير مناسب للبرمجيات المعقدة التي تحتاج إلى عمليات حسابية كبيرة.
- عدم توافقه مع البنيان الهيكلي للبرمجة، حيث لا يمكن تقسيم المشروع إلى أقسام و مستويات دون تحليل النظام بصورة شافية و دقيقة.

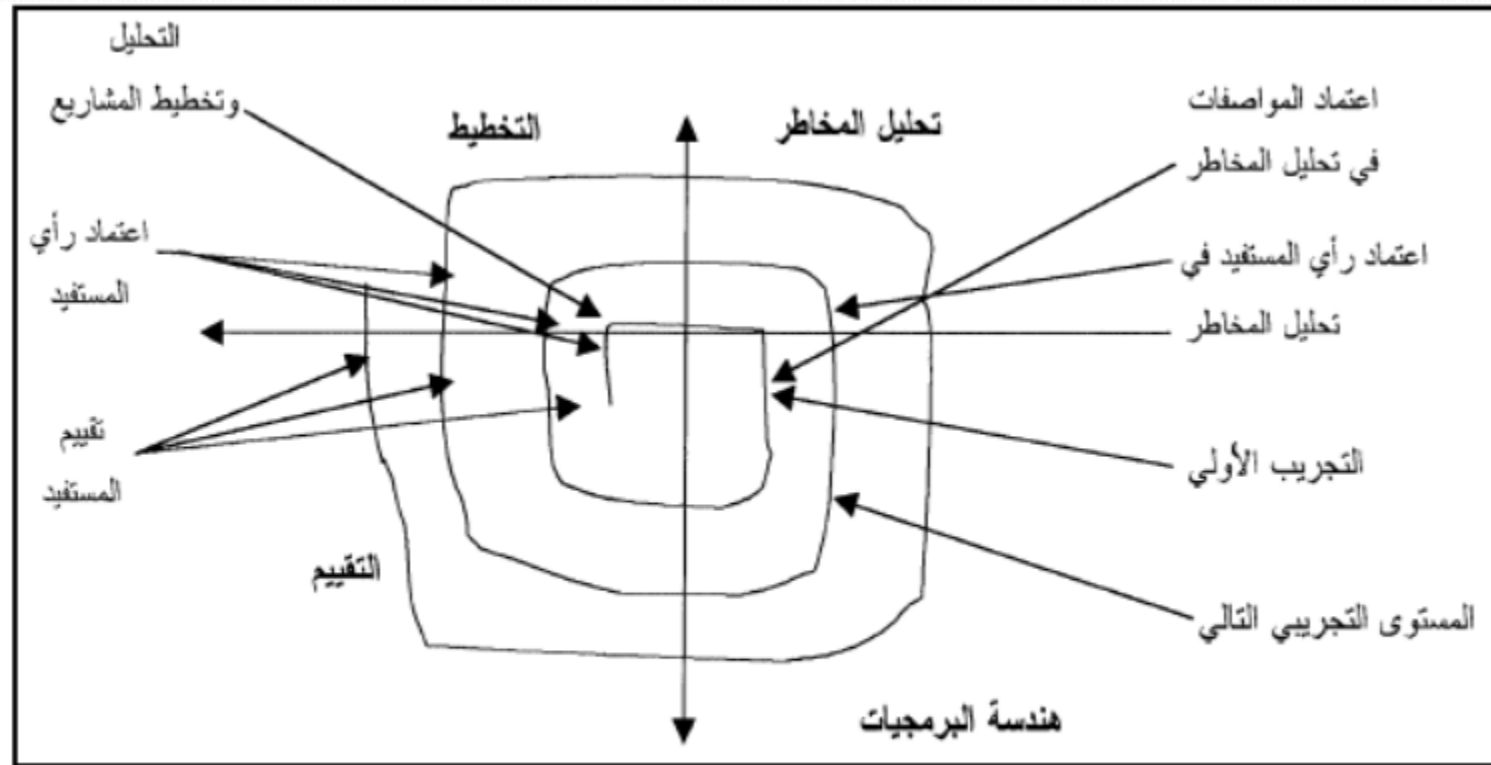
٢. أساسيات تحليل المتطلبات

النموذج الحلزوني The Spiral Model

يجمع هذا النموذج بين كل من خصائص النموذج التجريبي والشلال اضافة الى استخدام خاصية تحليل المخاطر Risk Analysis.
يتكون النموذج الحلزوني من أربع مقاطع:

- التخطيط: تحديد الأهداف و القيود.
- تحليل المخاطر: تحديد المخاطر المحتملة، حصرها و إزالتها.
- هندسة البرمجيات: تطوير المستوى اللاحق من المنتج.
- التقييم: تقييم المنتج البرمجي، من قبل المستخدم، و استقبال مقترحاته و تعديلاته.

٢. أساسيات تحليل المتطلبات



الشكل (4) النموذج الحلزوني

٢. أساسيات تحليل المتطلبات

- بناءً على مقترحات وتعديلات المستخدم، تبدأ المرحلة التالية لإنتاج منتج المرحلة التالية (التخطيط، تحليل المخاطر (بعد التعديلات)، هندسة البرمجيات، التقييم).
- عند تنفيذ هندسة البرمجيات في القسم الخاص بها لتطوير النموذج التجريبي الخالص بالدورة الحالية، يتم تنفيذ هندسة البرمجيات باستخدام النموذج السلمي، أو النموذج التجريبي.

٢. أساسيات تحليل المتطلبات

يعد النموذج الحزوني أفضل النماذج المتوفرة:

- لمرونته وسهولته في التحديث والتعديل.
- تقييم المخاطر ومراقبتها.
- توفير أرضية للتعاون بين المستخدم والمبرمج.

من مساوئ النموذج الحزوني:

- حاجته إلى خبراء في تحليل المخاطر وتقييمها.
- صعوبة إقناع المستفيد بأهميته.
- قلة مستخدميه.

٣. التحليل الهيكلي Structured Analysis

التحليل الهيكلي وتقنية التصميم (Structured Analysis and Design Technique SADT):
هما من الطرق المتبعة في استخدام التمثيل بالرسم أثناء الإعداد لبناء النظام، و حتى للتخطيط لمرحلة تصميم النظام.

- لوصف نشاط ما من أنشطة النظام، يستخدم شكل له ثلاث مداخل، و جزء للتحكم، و مخرج واحد.
- ص ٧٦ الشكل (٥) مثال على وصف نشاط في نظام.
- و بتكرار الشكل لكل الأنشطة المطلوبة، و تحديد الربط المناسب بينها، نخرج بتمثيل مناسب لتحليل النظام المطلوب.Z.
- ص ٧٦ مثال (١).
- أيضًا يوجد لغات خاصة تستخدم لوضع وصف رسمي للمتطلبات بأسلوب رياضي، يمكن أن يوفر لنا البرهنة الرياضية لهذا الوصف، أو الآلية، و من هذه اللغات لغة تحمل اسم Z.

٣. التحليل الهيكلي Structured Analysis

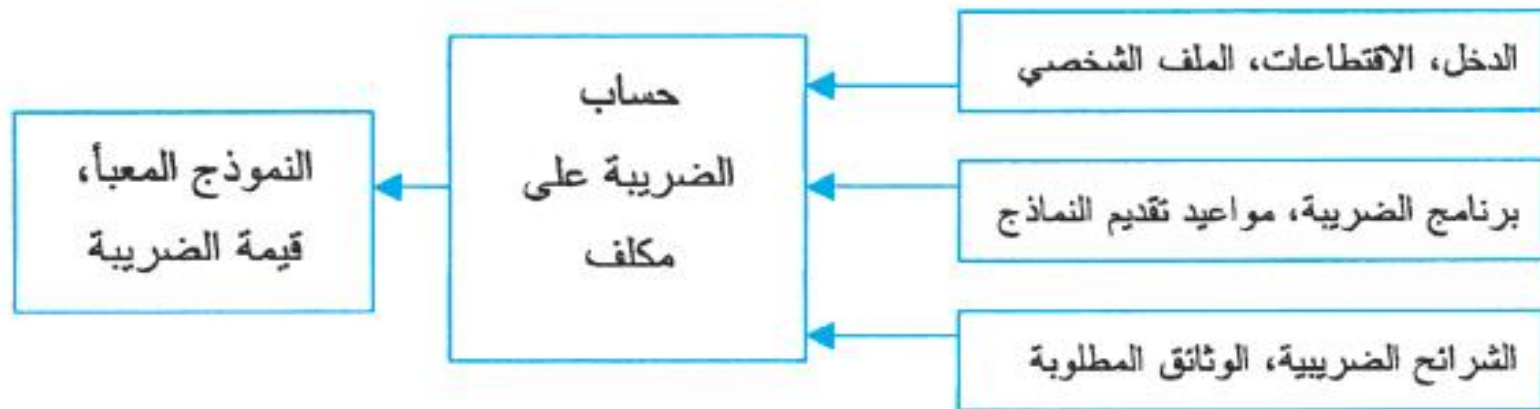


الشكل (5) : وصف نشاط في النظام

٣. التحليل الهيكلي Structured Analysis

مثال/

إذا أردت زيارة دائرة ضريبة الدخل، وجمعت بعض المعلومات التي تتعلق بحساب الضريبة على مكلف، لوجدت أن حساب الضريبة يمكن أن يمثل بالشكل ٦ بمداخله الثلاثة وبعض التفاصيل لها، وبمخرجه الوحيد وبعض التفاصيل له.



الشكل (6) : تمثيل حساب الضريبة بطريقة SADT

٤. التحليل الكينوني Object-Oriented Analysis

يمكن تطبيق فكرة الكينونات في التحليل، حيث أن هذا أسلوب جديد يختلف عن أساليب التحليل المعتمدة على الوظائف (Functions) و البيانات (Data)، مثل SADT و SA/SD، قتلک الوظائف تفصل بين الوظائف و البيانات، و تعامل كلاً منهما على انفراد، أما أسلوب التحليل الكينوني، فيكامل بينهما و يتعامل معهما من خلال ترابط مفيد.

يحتوي التحليل الكينوني على مجموعة من الخطوات:

١. إيجاد الكون و تحديده

من خلال استعراض المسألة و مفرداتها يمكن استنتاج أي المسميات تعتبر أساسية، و تعتمد عليها الكثير من الإجراءات في النظام المتوقع.

يجب استخدام المسميات التي تتصف بالثبات و الديمومة في مراحل النظام كلها (مثلاً موظف، عميل، حساب...)، فمثل هذه المسميات هي التي نريد تحديدها في هذه الخطوة.

٤. التحليل الكينوني Object-Oriented Analysis

٢. تصنيف الكينونات

بعض الكينونات يعتبر جزءًا من الآخر ويعتمد عليه.

بعضها تربطه علاقات مع البعض الآخر، ويعملان ويؤثران معًا.

بعضها يرث إجراءات وعمليات من البعض الآخر.

استنادًا إلى مثل هذه التصنيفات يمكن تصنيف الكينونات أصناف، بحيث توجد بين كينونات الصنف الواحد صفات مشتركة.

٣. تفاعل الكينونات مع بعضها البعض

يجب معرفة كيفية العلاقة بين كل كينونين، بمعرفة كيفية استفادة كينون من كينون آخر، وكيفية تأثير كل كينونين على بعضهما، حتى يتم ربط أجزاء النظام مع بعضها.

فيجب معرفة ما هي النتائج التي يوفرها كينون ما وتنتظرها الكينونات الأخرى لتعمل عليها، وهذه الخاصية تسمى بالتفاعلية بين الكينونات.

٤. التحليل الكينوني Object-Oriented Analysis

٤. العمليات على الكينونات

حتى تؤدي الكينونات عملها، لابد من وجود عمليات يتم إجراؤها من خلالها وعليها، من هذه العمليات ما يكون:

- بسيطاً يتم إجراؤه من خلال كينون واحد (مثل: عملية الإضافة أو السحب).
 - مركبة يتدخل في إنجازها أكثر من كينون (مثل طباعة تقرير طلاب كلية معينة، فاعتمد على كينون يمثل الكلية، وكينونات تمثل طلابها).
- و ينصح بالابتعاد عن الغوص في العمليات المركبة بعمق كبير.

٥. التنفيذ من خلال الكينونات

يجب تعريف كل كينون بشكل تام، فيجب تعريف خصائصه، عملياته، علاقاته مع الكينونات الأخرى، فكل العمل القادم على النظام يدور حول هذه الكينونات المكونة للنظام و كيفية ارتباطها مع بعضها.

٤. التحليل الكينوني Object-Oriented Analysis

في النهاية نتوصل إلى أن التحليل الكينوني يؤدي إلى:

- فهم أشمل للنظام المطلوب.

- بناء منطقي للنظام.

فالكينونات يجب تحديد:

- البيانات الخاصة بكل منها.

- العمليات التي تؤدي عليها.

- العمليات التي يتشارك فيها أكثر من كينون.

لتحديد كيفية عمل النظام و ترابط أجزائه لإنجاز المهمة المطلوبة.

و كلما حرصنا أثناء تحليل النظام أن تكون التغيرات في نطاقات لا تؤثر إلا في عدد قليل من

الكينونات، فسوف نحصل بذلك على نظام يتمتع بالثبات و الاعتمادية.

Questions or Comments?

