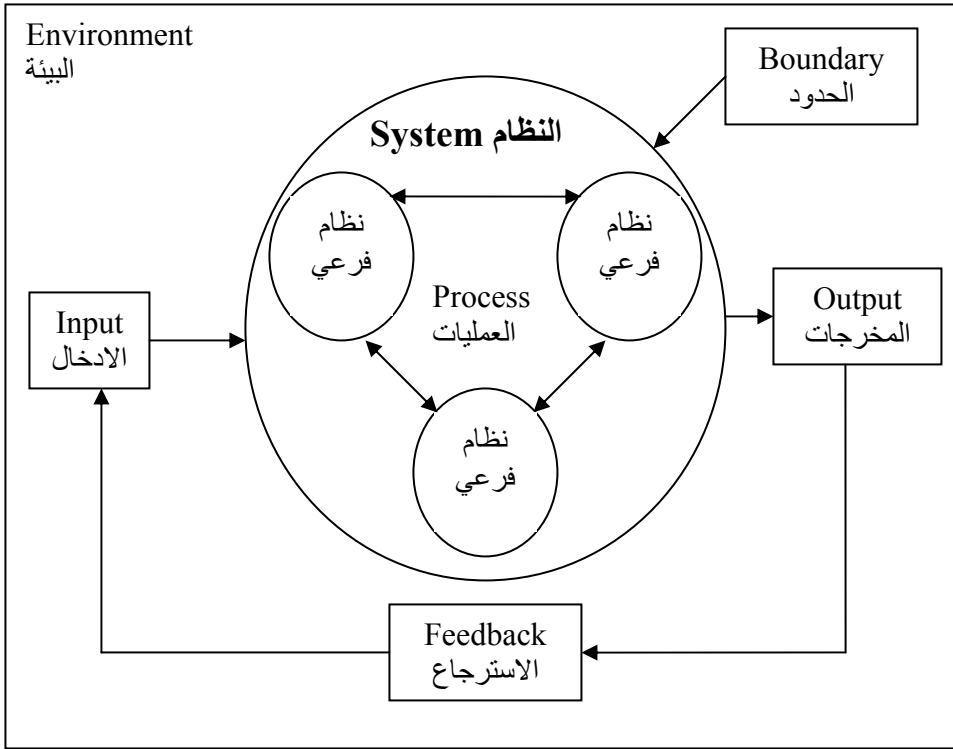


## 1. مقدمة Introduction: 1.1 مفهوم النظام System concept:

النظام هو مجموعة متفاعلة من العناصر بينها مبادئ عامة حاکمة بغض النظر عن طبيعتها والعلاقات الحاکمة بينها. و في تعريف آخر  
النظام هو العملية التي تنفذها مجموعة من العناصر المتحدة في الوظيفة و التشغيل لتحقيق أهداف معينة.  
و في تعريف آخر  
النظام هو مجموعة من الأهداف المرتبطة بعلاقات منظمة لتنفيذ و وظيفة معينة.  
و من الممكن أن يتكون النظام من مجموعة من الأنظمة الفرعية الأصغر من حيث المهام والوظائف التي تم توزيعها على مجموعة من الأنظمة الفرعية لتكون فيما بينها نظام اشمل ليحقق هدف اكبر.  
وكما هو موضح بالشكل 1 فإن المكونات الرئيسية لنظام هي: البيئة Environment, الإدخال Input, العمليات Processes, النظام الفرعي Subsystem, العلاقات Relationships, الإخراج Output, الحدود Boundary و الاسترجاع Feedback. وفي الجزء التالي سنأتي لشرح كل جزء من هذه الأجزاء.



الشكل 1: النظام System.

## 1.2 خصائص النظام System characteristics:

يتكون النظام من العناصر التالية:

### البيئة Environment

بيئة النظام هي مجموعة من عناصر ومكونات بينها علاقات والتي لا تكون عناصرها جزءا من النظام ولكن أي تغيير في أي عنصر من عناصرها ينتج عنه تغيير في حالة النظام. وبذلك فإن بيئة النظام تتألف من جميع المتغيرات التي تؤثر على الحالة، فهي التي تحتوي النظم الفرعية التي ليست جزءا من النظام ولكنها تتأثر به وتؤثر فيه.

### الحدود Boundary:

هي نطاق العمل المطلوب مثلا قد تحوي المنظومة الجديدة على معلومات عن الزبون والمبيعات ولكن دون أن تشمل على المخزون.

### الإدخال Input:

هي مجموعة من المتطلبات (بيانات) والأوامر التي يتم إدخالها من خارج النظام و يحتاجها لتنفيذ العمليات داخل النظام. وتتم عمليات الإدخال إما بواسطة مستخدم النظام أو نظام فرعي خارج النظام.

### العمليات Processes:

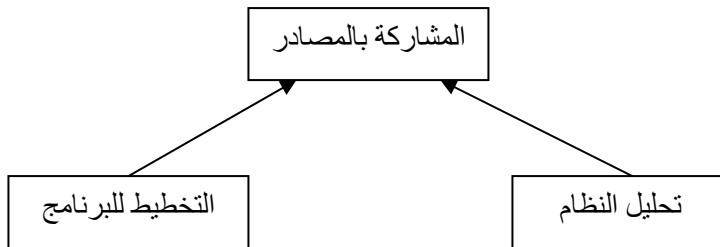
هي مجموعة من الاجراءات التي تتحكم فيها علاقات محددة لتنفيذ مجموعة مهام والتي تحقق الهدف العام للنظام. هذه المهام او الوظائف التي يقوم بتنفيذها النظام لانجاز هدف محدد يعتمد في تنفيذ على المدخلات (بيانات) لمعالجتها تحت عملية تحكم معينة لكي يضمن الدقة في الوصول الى معلومات التي سيستفاد منها في اتخاذ القرار السليم وتنظيم العمل.

**النظام الفرعي Subsystem:** هو عبارة عن نظام ويحمل مواصفات النظام العادي لكنه ينفذ مهمة جزء من عملية كاملة لنظام شامل أو اكبر.

### العلاقات Relationships:

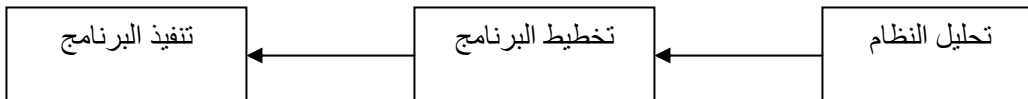
هي الصلات التي تربط بين النظم الفرعية للنظام أو البيئة حيث لا يمكن لأي نظام فرعي أن يؤدي وظيفته بمعزل عن غيره لأنه لابد من اعتماده على بيانات الواردة إليه من نظم فرعية أخرى حتى يستطيع تأدية وظيفته، كما أن ما ينتجه من معلومات هي الأخرى تعتبر مدخلات أي بيانات لنظم فرعية أخرى. وللعلاقات أنواع مختلفة تحدد المراد منها وهي:

(أ) العلاقات المتبادلة المجمعّة: ويراد بها استخدام أفراد أو إدارات النظم الفرعية لمجموعة من المصادر المشتركة.



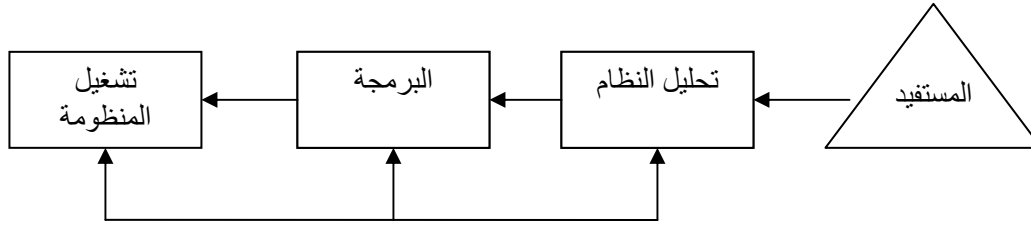
شكل 2: العلاقات المتبادلة المجمعّة.

(ب) العلاقات المتبادلة المتعاقبة: وتعني ان مخرجات احد النظم هي مدخلات لنظام اخرى.



شكل 3: العلاقات المتبادلة المتعاقبة.

ت) العلاقات المتبادلة التبادلية: حيث ان تنفيذ عمل نظام فرعي يؤثر بالتبادل على عمل نظام اخر.



شكل 4: العلاقات المتبادلة التبادلية.

### المخرجات Output:

هي المعلومات المتحصل عليها من المنظومة بعد معالجة المدخلات, وقد تأخذ هذه المخرجات شكل تقارير او استعلامات.

### الاسترجاع Feedback:

بعض الأنظمة تتمتع بعامل التحكم وذلك عن طريق التغذية الاسترجاعية والتي تتم بإعادة ادخال المخرجات كمدخلات جديدة للأعادة المعالجة و انتاج مخرجات جديدة , لغرض تحسين او تغيير المخرجات حسب قياسات معينة. في تحليل النظم, يتم الاستفادة من التغذية الاسترجاعية لتوضيح أو تنقيح متطلبات المستخدم.

### 1.3 أنواع الانظمة Systems classifications:

وتأسيسا على مفهوم البيئة وعلاقتها بالنظام تم تقسيم النظم الى نظم مغلقة (Closed System) ونظم مفتوحة (Open System).

#### النظم المغلقة Closed System:

النظام المغلق هو النظام الذي ليس له علاقة من اي نوع مع البيئة المحيطة به ويعمل بمعزل عنها. بمعنى لا يحتاج النظام الى مدخلات او عملية اخراج من و الى البيئة لكي يعمل. من أمثلة الأنظمة المغلقة البيوت الزجاجية, الساعة, الكتاب, المحركات التي تعمل بالبطاريات, التفاعلات الكيميائية التي لا تحتاج إلى مؤثر خارجي.

#### النظم المفتوحة Open System:

النظام المفتوحة عكس النظام المغلق فهو يتفاعل مع بيئته ويعتمد عليها في اتمام عملياته عن طريق المدخلات التي تتم معالجتها ويقوم بإخراج المعلومات للبيئة كنتائج عمل معين, كذلك يتأثر بكل العوامل الخارجية المحيطة به. الأمثلة على هذا النوع من الأنظمة عديدة مثل جسم الإنسان , نظام إداري , نظام الحاسوب...الخ.

### 1.4 أمثله على الانظمة:

نظام الحاسوب: هو مجموعة من الأجهزة والبرمجيات تحت نوع معين من التحكم لمعالجة بيانات وإنتاج معلومات.

نظام تسوق: مجموعة من الناس والبضائع والمعدات والإجراءات تعد أو تنتج وتوزع بضاعة او خدمات للمستخدم.

نظام الخطوط الجوية: ويتكون هذا النظام من موظفين وطائرات ورحلات ومسافرين بالإضافة لعمليات شحن, و يعمل هذا النظام على تنظيم رحلات للمسافرين وتقديم خدمات لهم خلال الرحلات بالإضافة الى تقديم خدمات شحن جوي.

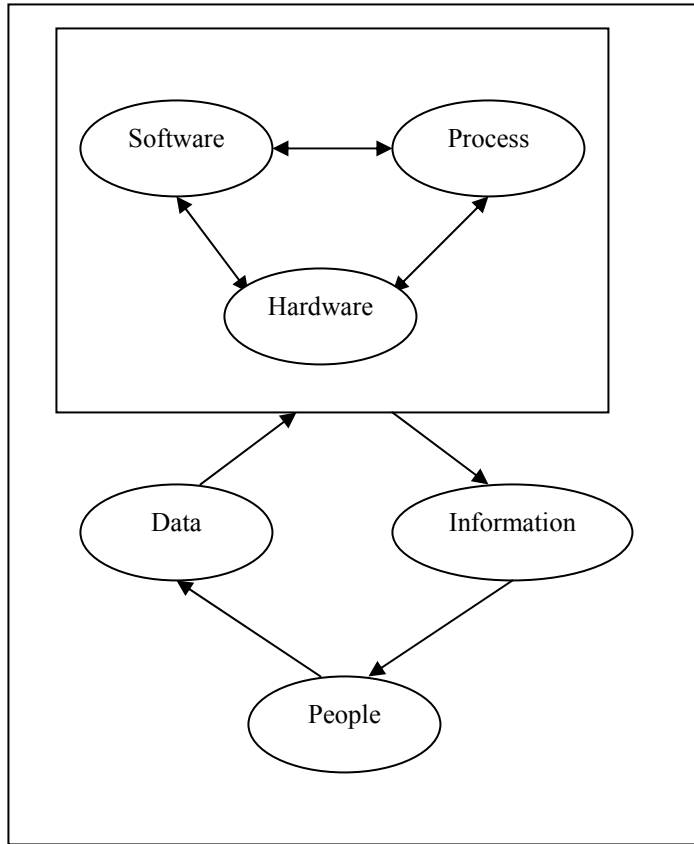
النظام المصرفي: يتكون من موظفين , اجراءات و عملة و يقوم النظام على اساس تقديم خدمات ايداع و صرف

وتقديم قروض الى الزبائن بحيث تحكم هذه المعاملة قوانين ولوائح محددة لتسيير السياسة المالية للمصرف.

## 2. أنظمة المعلومات Information Systems:

تعرف أنظمة المعلومات بأنها أنظمة تتكون من بيانات وبرمجيات وأجهزة وأشخاص وعمليات معالجة. وكما هو مبين في شكل 5 تتكون من الأجزاء التالية:

1. البرمجيات Software: وتشمل البرامج والبيانات و التوثيق وهي تنقسم الى نوعين  
أ- برمجيات تطبيقية Application software: مثل برمجيات الرواتب ومنظومات ادارة المخازن.  
ب- برمجيات النظم System software: مثل أنظمة التشغيل والمترجمات وبرامج النسخ الاحتياطي وبرامج الحماية من الفيروسات.
2. العتاد او الاجهزة Hardware: و هي تشمل الحاسوب و الاجهزة الملاحقة له مثل الطابعة والماسحة , بالإضافة الى اجهزة الاتصال لربط الشبكات.
3. الأشخاص People: هم المستخدمين للمنظومات والموظفين او المستفيدين مثل الزبائن والموردين والمصدرين.
4. البيانات Data: هي سبل من الحقائق في صورتها الأولية قبل معالجتها وتحويلها إلى معلومات مفيدة مثل الأسماء , العناوين , الاسعار , الكميات , اسماء المواد والدرجات.
5. المعالجة Process: هي مجموعة الاجراءات التي تتم على البيانات لتحويلها الى معلومة مفيدة.



شكل 5: أنظمة المعلومات Information System.

وتنقسم نظم المعلومات حسب وظائفها كالتالي:

#### منظومة معالجة المعاملات Transaction Processing Systems:

وتسمى ايضا منظومات العمليات مثل منظومة طلبية مبيعات ومنظومة جباية الفواتير. ومهمة هذه المنظومات معالجة البيانات واجراء حسابات وتصنيفات معينه عليها لاصدار فواتير وطلبات.

#### منظومة إدارة المعلومات Management Information System:

هذا النوع من المنظومات يساعد المدراء في اتخاذ قرارات أفضل, فمثلا باستخدام بيانات المبيعات المأخوذة من منظومة العمليات يمكن لمنظومة إدارة المعلومات ان تشير إلي الأصناف الأكثر رواجاً والأصناف الأقل مبيعاً. وبهذه الطريقة يستطيع متخذ القرار أو المسؤول ان يأخذ القرار الصحيح بجلب الأصناف الأكثر مبيعاً وارباح الى المخزن قبل نفاذها.

#### المنظومات الخبيرة Expert Systems:

هي المنظومات التي تستخدم تقنية الذكاء الاصطناعي سواء كان استعمالها في التحكم في آلات أو أنظمة معلومات مثل التجارة الإلكترونية أو أنظمة طبية معينة. أصبحت المنظومات الخبيرة يشار إليها باسم منظومات إدارة المعرفة Knowledge Management Systems بعد أن استخدمت في الإنترنت لإدارة كم هائل من المعلومات على الشبكة للبحث عن المعرفة والوصول إلى المعلومة المطلوبة. تملك منظومة ادارة المعرفة قاعدة بيانات كبيرة لتكون تحت سيطرة المستخدم للوصول الى المعلومة التي يحتاجها باستخدام محركات بحث داخل المنظومة.

#### منظومات تحسين إنتاجية المستخدم Employee Productivity Enhancement Systems:

وتسمى ايضاً منظومة ميكنة المكاتب وتتميز بسهولة استخدامها وهي منظومات يستخدمها الموظف لمساعدته في انجاز اعماله الروتينية (مثل الطباعة) بكفاءة عالية ومن أمثلة هذه المنظومات:

نظم البريد الإلكتروني Email system.

معالجة النصوص Word processing.

الجدول الإلكتروني Spreadsheets.

قواعد البيانات Databases.

### 3. دورة حياة وإعداد النظام :System Development Life Cycle

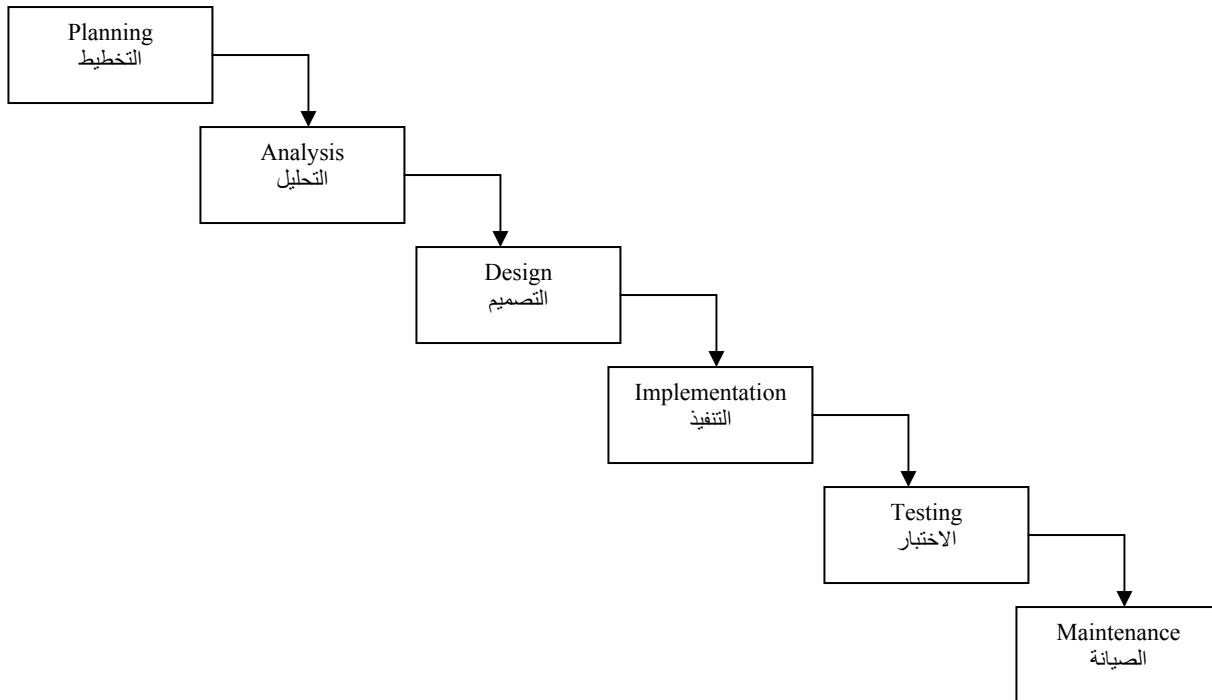
إن من الأهمية بمكان أن يتم التخطيط و الإعداد الجيد لأي منظومة حتى لا تحدث مشاكل و أخطاء قد تكلف الكثير بسبب سوء الإعداد و التخطيط. لذلك فإن المنظومات يجب أن يتم إعدادها بطريقة منظمة باستخدام منهجية (طريقة) سليمة ويوجد ثلاثة منهجيات تستخدم في تحليل وتصميم نظم المعلومات هي:

1- الطريقة غير الهيكلية Unstructured Methodology:  
هي طريقة قديمة في إعداد المنظومات بدأت في الستينات مستخدمة المخططات الانسيابية كوسيلة للتحليل والتصميم. وهي لا تستخدم حالياً إلا نادراً نظراً لتعدد المنظومات المستخدمة.

2- الطريقة الهيكلية Structured Methodology:  
بدأ استخدام هذه الطريقة في السبعينات وتتميز بالأدوات والتقنيات الهيكلية المنظمة والسهولة الاستعمال. هذا النوع من الطرق تناسب إعداد المنظومات الكبيرة والمعقدة نظراً لأنها تقسم المنظومة إلى أجزاء صغيرة لتبسيطها. كمثال على هذه الطريقة (مخطط انسياب البيانات و المخطط الهيكلية).

3- الطريقة الشيئية (الكائنية) Object Oriented Methodology:  
الطريقة الشيئية (الكائنية) تستعمل تقنيات متقدمة مثل إعادة استعمال الأجزاء البرمجية لتقليص وقت إعداد المنظومات. و بعض الأدوات في هذه الطريقة هي مخطط حالة الاستعمال (Use Case Diagram) و مخطط الفصيلة (Class Diagram).....الخ.

إن أبسط نموذج لدورة حياة إعداد منظومة هو ما يعرف بالنموذج التتابعي أو النموذج التدفقي الذي يعني أننا نقوم بإعداد المنظومة في تسلسل أي مرحلة بعد مرحلة. يتكون هذا النموذج كما هو مبين بالشكل 6 من عدد من المراحل المتعاقبة بحيث تعرف كل مرحلة بأنها عبارة عن عدد من الأنشطة يجب أن تنجز في فترة زمنية وقبل الانتقال إلى المرحلة التي تليها.



شكل 6: دورة حياة وإعداد النظام System Development Life Cycle

ويمكن توضيح هذا النموذج بشيء من التفصيل كما هو مبين بالجدول التالي:

Phase المرحلة	النشاطات Activities
Planning التخطيط	1- طالب المستخدم User request 2- دراسة الجدوى Feasibility Study 3- خطة المشروع Project Plan 4- مقترح المشروع Project Proposal
Analysis التحليل	1. إيجاد الحقائق Fact find 2. تحليل المتطلبات Requirement analysis
Design التصميم	1. التصميم المبدئي Initial Design 2. التصميم المفصل Detailed Design
Implementation التنفيذ	1. التشفير Coding 2. اكتشاف الأخطاء Debugging 3. اختبار الوحدة Unit test
Testing الاختبار	1. اختبار التكامل Integration test 2. اختبار النظام System test 3. اختبار القبول Acceptance test
Maintenance الصيانة	1. التحسين Enhancement 2. التكيف Adaptation 3. التصحيح Correction 4. إعادة الهندسة Re-engineering

جدول 1: نشاطات مراحل دورة حياة النظام

عند البدء في دورة حياة تطوير النظام يجب ملاحظة التالي:

1. كل مرحلة تحتاج إلى مستلزمات (برمجيات, أجهزة و بشر) لإكمال كل عملياتها.
2. كل مرحلة تحتوي على مدخلات ومعالجة ومخرجات.
3. يجب اجراء مراجعة في نهاية كل مرحلة.
4. يجب إعداد وثيقة المواصفات ومراجعتها في نهاية كل مرحلة.
5. يجب التحقق من اكتمال وصحة ووضوح وثيقة المواصفات في كل مرحلة.

في الجزء التالي سنبدأ في شرح تفصيلي لكل مرحلة من مراحل تطوير واعداد النظام.



### 3.1 مرحلة التخطيط Planning Phase

في هذه المرحلة يوجد اربعة أنشطة رئيسية يقوم بها المحلل وهي:

- طلب المستخدم User request
- دراسة الجدوى Feasibility study
- خطة المشروع Project plan
- مقترح المشروع Project proposal

#### طلب المستخدم User request

الهدف من هذه الخطوة هي معرفة احتياجات المستخدم والمشاكل التي يواجهها في نظامه القائم. تستخدم نماذج خاصة بملاها المستخدم لغرض تحديد المشاكل الموجودة في النظام الحالي وتعريف الحدود والأهداف المرجوة من النظام المقترح. و في هذا النموذج , يمكن أن يطلب المستخدم واحداً من الاختيارات التالية:

- المطلوب نظام جديد.
- المطلوب تحسين النظام الحالي.
- المطلوب تصحيح أخطاء المنظومة الحالية.

في المثال التالي نموذج مبسط لطلب المستخدم. هذا النموذج تسلمه لأدارة الشركة لاتخاذ القرار بخصوص البدء في التحليل وذلك بتكليف محلل نظم للقيام بالمهمة.

من: رئيس قسم المبيعات

إلى: المدير العام لشركة

الموضوع: نظام المبيعات

يقوم قسم المبيعات ببيع الاثاث. يدفع الزبون إما نقداً أو بالدين. يستقبل القسم أكثر من 50 زبون في اليوم الواحد منهم 50% تقريباً الدفع نقداً. ويتوقع القسم زيادة في المبيعات نظراً لجودة المنتجات التي تقدم للزبائن. ومع ذلك أود ان افيدكم بأننا في حاجة لتحسين أداء قسم المبيعات. فالعمليات في القسم تجرى يدوياً. والنظام الحالي يعاني بعض المشاكل في الحصول على المعلومة عند الحاجة إليها. ونحن نود أن نطور خدماتنا و التفوق على المنافسين في السوق. ويمكن أداء ذلك بربط جميع فروع المبيعات و بهذه الطريقة نحصل على رضا الزبائن الحاليين وجلب زبائن جدد.

#### نموذج طلب المستخدم

#### دراسة الجدوى Feasibility study :

هذه الدراسة يجب ان تتم لأغلب المشاريع المتوسطة والكبيرة الحجم وهي تتكون من:

- الجدوى الاقتصادية
- الجدوى الفنية
- الجدوى التشغيلية

الجدوى الاقتصادية: هي تحليل التكاليف والمزايا لمعرفة ما إذا كانت المزايا تفوق التكاليف المتوقعة. لتحقيق هذا الهدف يقوم المحلل بإعداد نموذج يذكر فيه المزايا المتوقعة مثل : تحسين الخدمات , تقليص التأخير , تحسين رضا الزبائن , اتخاذ قرارات أفضل , تحسين صورة الشركة , إعداد موقع لشركة على شبكة الانترنت.

ثانياً يقوم المحلل بحساب التكلفة التقديرية للمشروع كالتالي:

تكاليف العتاد والمعدات (-----د.ل.)

تكاليف البرمجيات (-----د.ل.)

تكاليف القوة العاملة (-----د.ل.)

تكاليف برامج التدريب (-----د.ل.)

تكاليف التجهيزات (-----د.ل.)

تكاليف الاستشارات (-----د.ل.)

مصاريف أخرى (-----د.ل.)

المجموع (-----د.ل.)

بالإضافة الى ذلك يجب دراسة الفترة الزمنية وتقدير الوقت اللازم لكي يبدأ في الحصول على عائد من المشروع يفوق التكاليف. تتم العملية بمقارنة التكاليف المتوقعة مع العوائد المتوقعة في كل سنة وحساب الفترة الزمنية التي يسترد فيها المشروع تكاليفه. وبذلك يمكن معرفة متى يبدأ المشروع في استرداد التكاليف ومتى تبدأ مرحلة تحصيل الأرباح.

الجدوى الفنية: من هذه الدراسة نتوقع الاجابة على الاستفسارات التالية:

- ❖ هل للشركة المعدات والبرمجيات و الشبكة اللازمة للمشروع؟ إذا كان ذلك غير متوفر , هل يمكن الحصول على هذه المستلزمات بسهولة؟
- ❖ هل للشركة الخبرات الفنية اللازمة ؟ إن كان غير ذلك فهل يمكن توفيرها؟
- ❖ هل سيتمكن النظام من معالجة حجم المعاملات المتزايد في المستقبل؟ إذا كانت الإجابة لهذه الاسئلة بنعم فإن النظام المقترح يعتبر ذا جدوى فنية؟

الجدوى التشغيلية: هذه الدراسة لتجيب عن الاستفسارات التالية:

- ❖ هل ستكون المنظومة بعد إعدادها سهلة الاستخدام وتعمل بكفاءة؟
  - ❖ هل سيتقبل المستخدمون والمدراء المشروع الجديد, وهل سيكون لديهم الشعور بأن المنظومة الجديدة ستحقق احتياجاتهم؟
  - ❖ هل سيتم قبول المنظومة بعد استلامها من قبل المستخدم؟
  - ❖ هل ستطلب المنظومة الجديدة تدريب؟ وهل سيتم التدريب قبل عملية التسليم؟
- إذا كانت الاجابة عن هذه الاستفسارات بنعم فإن النظام ذو جدوى تشغيلية.

دراسة الجدوى هي عبارة عن تحليل مبدئي يتم قبل البدء في المشروع. وينتج عن هذه الدراسة تقرير كما هو مبين بالجدول التالي:

1.	اسم المشروع:
2.	وصف النظام الحالي:
3.	الحدود:
4.	الافتراضات والشروط:
5.	حلول بديلة:
	الخطوط العريضة للنظام
	الجدول الزمني
	القيود
	المستلزمات
	المزايا والعيوب
	الفوائد والتكاليف
6.	التوصيات والخلاصة
7.	الملحقات

محتويات تقرير دراسة الجدوى

#### وثيقة خطة المشروع:

يجب أن تعد هذه الوثيقة بعد الموافقة على دراسة الجدوى واختيار الحل المطلوب. وتتضمن وثيقة خطة المشروع النقاط التالية:

- ملخص المشروع ,
- جدول المشروع ,
- قيود المشروع ,
- الأهداف المرجوة ,
- الوظائف المطلوبة ,
- نطاق المشروع ,
- التكلفة التقديرية ,
- المستلزمات المطلوبة ,
- الأدوات اللازمة لمرحلة التحليل والتصميم ,
- خصائص الجودة المطلوبة ,
- تحليل الأخطار.

#### وثيقة مقترح المشروع:

يجب أن تعرض هذه الوثيقة البنود التالية:

- وصف المشكلة
- مشاكل النظام الحالي
- شرح للنظام المقترح وتبرير اختيار النظام مقارنة بين الحلول البديلة
- الجدوى الاقتصادية و الفنية والتشغيلية للنظام المقترح
- الأدوات التي ستستخدم في المشروع
- الأفراد ودورهم في المشروع
- تأثير المشروع المقترح على عمليات النظام الحالي
- نسخة من خطة المشروع التي تم عرضها في البند السابق.

### 3.2 مرحلة التحليل Analysis Phase:

الهدف الرئيسي من التحليل هو تعريف متطلبات المستخدم. بتعبير آخر يجب أن نحدد في هذه المرحلة ما هو متوقع من النظام المقترح حتى نحقق متطلبات المستخدم. يمكن أن نقسم متطلبات المستخدم إلى ما يلي:

1. متطلبات وظيفية: الوظائف التي ستؤديها المنظومة.
2. متطلبات الأداء: ماهي السرعة المطلوبة من المنظومة لأنجاز الوظائف ويقاس هذا الأداء بوقت الاستجابة.
3. متطلبات الإدخال: وهي البيانات التي تدخل المنظومة لغرض المعالجة.
4. متطلبات الأخراج: هي المعلومات التي تخرج من المنظومة بعد إدخال المدخلات و معالجتها. وقد تكون المخرجات على شكل كشوفات , تقارير أو إستعلامات.
5. متطلبات الجودة: الاعتمادية والأمن وسهولة الاستعمال وقابلية الصيانة.

لكي نحدد متطلبات المستخدم بالكامل , هناك بعض البنود التي يجب أن يحددها محلل النظم وهي:

**ملخص عن المشروع:** تعريف لنشاط المستخدم والمشاكل التي تحتاج إلى حل. مثلاً قد يوجد لدينا متجر لبيع الكتب نقداً أو على الحساب ويشعر صاحب المتجر بالحاجة إلى تطوير نظام سجلات الزبائن لزيادة الدخل.

**الأهداف:** شرح الغاية والغرض من المنظومة الجديدة مثلاً يجب على المنظومة أن تسرع معاملات معينة بنسبة 50% . وفي مثال آخر المستهدف من المنظومة الجديدة توفير تقارير فورية للإدارة من خلال الشبكة.

**الحدود:** أي نطاق العمل المطلوب أي توضيح المعلومات والعمليات التي ستشملها المنظومة. مثلاً قد تحوي المنظومة الجديدة على معلومات عن الزبائن والمبيعات ولكن دون أن تشمل على المخزون.

**البيئة:** هي كل ما يحيط بالمنظومة من عمليات و إجراءات وسياسات قد تؤثر على المنظومة أو تتأثر بالمنظومة. مثلاً قد يكون نظام المخزن بيئة لنظام المبيعات.

**القيود:** هي الضوابط التي تفرض على المنظومة المقترحة مثل الوقت المتوفر و الميزانية المرسودة. فمثلاً قد تحدد الميزانية بمبلغ محدد بالأرقام مع تحديد المدة الكافية لأنجاز المنظومة وتحديد المدة الزمنية لضمان عمل المنظومة.

وكما ذكرنا في جدول 1 تشمل هذه المراحل على نشاطين هما إيجاد الحقائق (جمع المتطلبات) و تحليل المتطلبات.

#### 1- إيجاد الحقائق (جمع المتطلبات):

يقوم المحلل في هذه المرحلة بجمع كل المعلومات التي تساعد على فهم طبيعة العمل وآلياته ومشاكله القائمة و يمكن جمع المتطلبات بعدد طرق وهي كالتالي:

البحث العام: ويستخدم فيه المراجع من المكتبة وشبكة المعلومات (الأنترنت).

الاستبيان: ويقوم المحلل بتصميم استبيان يضع فيه كل الاسئلة والاستفسارات المناسبة لكي يصل إلى المعلومة الصحيحة والمهمة.

المقابلات: تنسيق إجتماعات مع الموظفين أو مستخدمي النظام و المسؤولين على اتخاذ قرارات مهمة داخل النظام القائم. يقوم المحلل بالاستفسار حول النظام الحالي ومتطلبات المنظومة الجديدة.

العرض التجريبي: يمكن ان يقوم المحلل بتصميم عرض تجريبي للمنظومة لعرضها على المستخدمين والمسؤولين للوصول لكي تساعد على وضع تصور واضح للمنظومة الجديدة و تجنب اي أخطاء أو سلبيات غير متوقعة.

مشاريع مشابهة: من المفيد جداً للمحلل الحصول على مشاريع مشابهة نفذت لدراساتها لكي يتمكن من الاستفادة من أخطاء الآخرين.

عينات نماذج وتقارير: من المهم ان يتحصل المحلل على نماذج و تقارير وفواتير النظام الحالي نظراً لأنها تحتوي على البيانات المهمة في عمليات الادخال والاخراج مع وجود تفاصيل حول العمليات التي ستنفذ داخل المنظومة.

- يجب على المحلل عند القيام بهذه النشاطات العمل بالأرشادات التالي:
- ✓ المقابلات الشخصية يجب تحديدها بتاريخ ومكان وفترة معينة.
  - ✓ اهداف المقابلة يجب ان تحدد مسبقاً.
  - ✓ يجب أن يكون الاستبيان قصيراً.
  - ✓ عند تصميم الاستبيان يجب استعمال أسئلة ذات إجابات قصيرة مثل (نعم و لا) او اجابات اختيارية.
  - ✓ من المفيد تنسيق الزيارات مع المدراء والموظفين المتواجدين.
  - ✓ اكتب ملاحظات قصيرة وراجع ماكتبته مع المستخدمين.

## 2- تحليل المتطلبات:

يبدأ المحلل في هذه الخطوة بعمل تنظيم للمتطلبات لغرض التوصل لفهم واضح لنظام القائم والنظام المقترح. يوجد مجموعة من الأدوات يستخدمها المحلل لتنفيذ عملية تحليل المتطلبات مثل:

- المخطط الانسيابي للبيانات Data Flow Diagram
- قاموس البيانات Data Dictionary
- جدول القرارات Decision Table
- شجرة القرارات Decision Tree
- الانجليزية المركبة Structured English
- المخطط System Analysis and Design Technique (SADT)
- مخطط العلاقات Entity Relationship Diagram (ERD)

في نهاية هذه المرحلة يقوم المحلل بإعداد وثيقة توصيات المتطلبات والتي تتضمن النقاط التالية:

- ملخص المشكلة
- مخطط انسياب البيانات
- قاموس البيانات
- وظائف المنظومة
- متطلبات الأداء
- البيئة
- القيود
- حدود المنظومة
- معيار القبول
- لغة البرمجة المستخدمة
- جدول وشجرة القرارات
- مخطط العلاقات

ومن المهم ان نلاحظ ان هذه الوثيقة تصف المشاكل وليس الحلول وهي نتائج الدراسة وليس عملية المعالجة, كما أنها وثيقة بين الزبون والمحلل وسوف تستخدم فيما بعد في التصميم. بالإضافة إلى ذلك فهي تقوم بتحويل الاحتياجات إلى متطلبات وتبين ماهو المتوقع من المنظومة وليس كيف تعمل. من المهم مراجعتها جيداً مع المستخدم لتفادي أي أخطاء مستقبلية.

ملاحظة: الفرق بين الاحتياجات والمتطلبات هو ان مصطلح متطلبات ادق واكثر قابلية للاختبار مثلاً:  
الاحتياجات: الملف يحمل 100000 سجل موظف  
المتطلبات: الملف يجب ان يحمل أسماء كل الموظفين في الشركة.

### 3.3 مرحلة التصميم Design Phase:

تقوم عملية التصميم بترجمة المتطلبات إلى تمثيل الحل بحيث التصميم يركز على التالي:

#### تصميم معماري Software architecture design:

يتم تحزئة النظام البرمجي إلى مكونات بحيث كل مكونة يمكن أن تجزأ أكثر إلى وحدات (أجزاء برمجية) باستخدام المخطط الهيكلي كأداة تصميم من أعلى إلى أسفل.

#### تصميم هياكل البيانات Data structures design:

هي عملية وصف للبيانات من حيث نوعها وطولها أو حجمها وكذلك وصف للملفات (قواعد البيانات) من حيث وصف محتويات كل ملف والحقول البيانية التي يحتويها مع وصف العلاقات التي تربط بين الملفات.

#### تصميم الخوارزميات Algorithms design:

هي عملية كتابة الخطوات المنطقية لأجزاء البرمجيات في المنظومة وتستعمل طريقة التشفير المرمزة لهذه الكتابة.

#### تصميم واجهة المستخدم User interface design:

في هذه الخطوة يقوم المحلل بإعداد تصميم لشاشات الرئيسية للأدخال في المنظومة والتي سيستعملها المستخدم ولهذا من المهم أن يكون هذا التصميم بالتشاور مع المستخدم حتي يسهل عليه استعمال المنظومة ولتفادي أي أخطاء تنتج بسبب سوء الاستعمال للمنظومة. بالإضافة إلى ذلك يقوم المصمم بتصميم شاشات الإخراج والتقارير التي تطبع على الورق بتحديد البيانات المهمة في كل شاشة أو تقرير أو كشف.

### 3.4 مرحلة التنفيذ Implementation Phase:

الهدف الرئيسي من مرحلة التنفيذ هي انتاج جميع البرامج باستخدام إحدى لغات البرمجة للحصول على منظومة تقوم بتنفيذ جميع الوظائف التي تم إعداد وصف لها في المرحلة السابقة (مرحلة التصميم). تعتمد هذه المرحلة على المبرمج بالذات (ليس بضرورة أن يقوم المحلل بمهمة البرمجة) وهو يقوم بتنفيذ النشاطات التالية:

- ترجمة مواصفات التصميم لكل جزء برمجي إلى برنامج بلغة البرمجة (شفرة المصدر).
- ترجمة البرنامج و البحث و تصحيح الأخطاء.
- اجراء اختبار لكل برنامج على حدى ثم اختباره في نطاق المنظومة كاملة وباستخدام بيانات ومدخلات حقيقية لتأكد من صحة المخرجات حسب طلب المستخدم. وهنا قد يكتشف أخطاء في التصميم.
- كتابة كتيب التشغيل لمساعدة المستخدم في استعمال المنظومة بكفاءة.

عند البرمجة يجب على المبرمج أن يتقيد بالنصائح التالية لمساعدته في إعداد منظومة ذات جودة عالية:

- استعمال لغة البرمجة التي تناسب التطبيق المطلوب: يجب اختيار لغة البرمجة التي لها جميع الإمكانيات المطلوبة.
- اجعل قابلية قراءة المخرجات افضل ما يمكن باستعمال الواجهة الرسومية: يوجد برمجيات تساعد في إعداد نماذج رسومية لشاشات يمكن استخدامها لأعداد تصميم مبدئي لها في وقت اقصر و بما يناسب المستخدم.
- اجعل قابلية قراءة نص البرنامج (شفرة المصدر) افضل ما يمكن باستعمال أسماء ذات معنى وجمل
- تعليقاتية لشرح وظيفة كل جزء من البرنامج.
- اجعل وقت اعداد المنظومة أقل ما يمكن باستخدام لغة برمجة ذات كفاءة عالية: يجب عدم اختيار لغة البرمجة التي تحتاج لوقت أطول في البرمجة , ومن المعروف ان لغات البرمجة تختلف من حيث درجة التعقيد في البرمجة والزمّن المطلوب للإعداد البرامج .
- اجعل عدد جمل شفرة المصدر أقل ما يمكن: بعض الوظائف والعمليات معقدة وطويلة من حيث الإجراءات وعدد خطوات التنفيذ وفي هذه الحالة يجب ان تقسيم هذه الوظائف والعمليات الى عمليات اصغر وبالتالي يصبح عدد حمل شفرة كل جزء برمجي أقل ما يمكن.
- اجعل الذاكرة المطلوبة أقل ما يمكن.

- يجب أن يحقق الجزء البرمجي ما يلي:
- أن يكون له مدخل واحد ومخرج واحد.
- تجنب استخدام جملة (أمر) إذهب إلى (GOTO).
- أن يحتوي على أقل من 30 جملة (صفحة تقريباً).
- أن يتم تنسيق البرنامج بطريقة هيكلية باستعمال المسافة والاسطر الخالية.
- أن لا يحتوي البرنامج على حلقات دوران متداخلة كثيرة (مثلاً لا يزيد عن 5 حلقات دوران متداخلة).
- أن يحتوي على توثيق كافي حتي يسهل عملية تصحيح وتطوير أي برنامج في المنظومة.

### 3.5 مرحلة الاختبار Testing Phase:

الهدف الرئيسي من الاختبار هو تحديد إمكانية الحصول على النتائج المطلوبة عند تشغيل المنظومة. وهذا يعني أن اختبار المنظومة الجديدة يجب ان يحقق:

- التخلص من الأخطاء اللغوية
- التخلص من الأخطاء المنطقية
- التخلص من الأخطاء التنفيذية
- تحديد أخطاء مدخلات المستخدم
- تقييم سرعة أداء المنظومة
- تقييم أمن المنظومة
- اكتشاف أي وظيفة مفقودة
- تحقيق متطلبات المستخدم
- تقييم توثيق المنظومة

لأختبار منظومة جديدة , عادة ما نستعمل عينة من البيانات و نقارن المخرجات (النتيجة) مع النتيجة المتحصل عليها من نفس العينة يدوياً. يجب أن تحتوي هذه العينة على جميع الحالات المحتملة للتأكد من عدم وجود أخطاء منطقية. كما ذكرنا سابقاً فإن كل جزء برمجي (برنامج فرعي) من المنظومة يتم اختباره منفصلاً قبل اختبار المنظومة ككل. الاختبار يشمل:-

- اختبار التكامل
- اختبار النظام
- اختبار المستخدم

في اختبار التكامل يتم ربط واختبار الأجزاء البرمجية (التي تم اختبارها منفردة مسبقاً) كنظام متكامل (وحدة واحدة) للتأكد من أن المتطلبات كما عرّفها المستخدم – قد تم تحقيقها. يوجد استراتيجيتان شائعتان لأختبار التكامل هما:

استراتيجية من تحت الى فوق

استراتيجية من فوق الى تحت

اختبار النظام يتحقق من جميع المتطلبات باستخدام عتاد بيانات حقيقية.

اختبار القبول يتم تنفيذه من قبل الزبون أو المستخدم لأثبات أن المتطلبات المدونة في وثيقة المتطلبات قد تم انجازها. وبذلك تعتبر المنظومة جاهزة للعمل.

### 3.6 مرحلة الصيانة Maintenance Phase:

الصيانة هي عملية جعل المنظومة تعمل بطريقة صحيحة في مواجهة العوامل التالية التي قد تؤثر على عملها:

- التغيرات بسبب أخطاء تحدث بعد تسليم المنظومة.
- استخدام تقنية جديدة.
- منع أي مشاكل قد تحدث في المنظومة بسبب رداءة في التصميم (إعادة هندسة المنظومة).
- تحسين قدرات المنظومة بإضافة وظائف جديدة.
- تبدأ الصيانة عندما يتم تسليم المنظومة و تركيبها (تحميلها) في موقع الزبون والبدء في تشغيلها.

النشاطات التالية يتم انجازها في هذه المرحلة:

تصحيح الأخطاء: يتعلق بتصحيح الأخطاء التي تظهر أثناء عملية تشغيل المنظومة و التي لم يتم اكتشافها في مرحلة الاختبار.

التكيف: نظراً للتغيرات السريعة في التقنية (برمجيات و معدات) قد تحتاج المنظومة الى التكيف مع بيئة جديدة مثل منظومة تشغيل أو معدات حاسوب جديدة.

التحسين: ويشمل إما تطوير إمكانيات المنظومة أو توفير وظائف أكثر للمنظومة عند اكتشاف متطلبات جديدة.

إعادة الهندسة: نقوم هنا بإعادة تصميم النظام وبرمجته إذا لزم الأمر لمنع أي مشاكل متوقعة من التصميم السيئ.



## 4. تخطيط المشروع Project Planning:

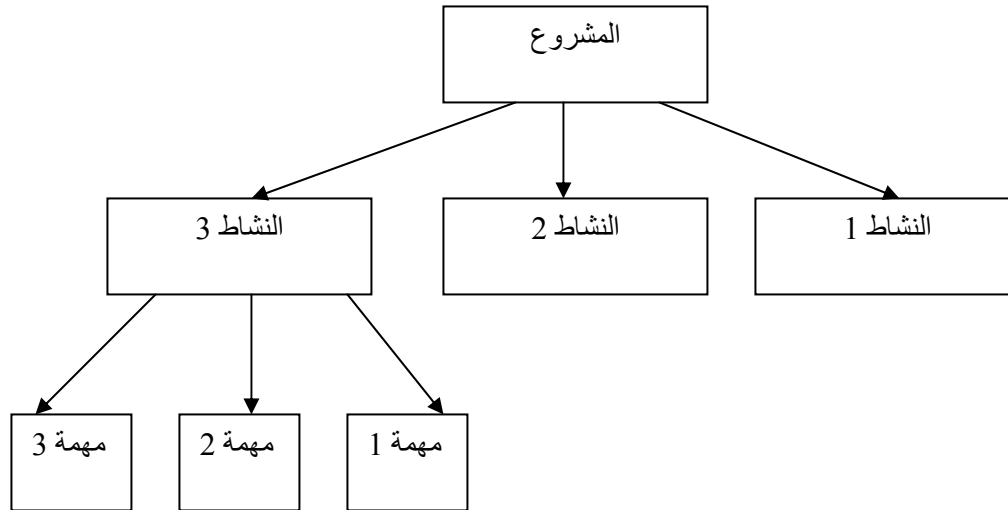
التخطيط هو كل الأنشطة الادارية التي تجرى لغرض تعريف الأهداف لمشروع ما والسبل اللازمة لتحقيق هذه الأهداف. يعتمد نجاح المشروع ألى حد كبير على التخطيط الجيد والتخطيط السيء عامل رئيسي في فشل المشروع.

### 4.1 أنشطة التخطيط Planning Activities:

- 1- تحديد الأهداف:  
مثال: قد يكون أحد أهداف المشروع هو تقليص الوقت الذي تستغرقه معاملة ما بنسبة 40% (مثل سحب صك مصرفي)
- 2- تحديد نطاق او حدود المشروع:  
مثال قد يكون نطاق منظومة برمجية للمعهد محصورة في قسم الدراسة والامتحانات فقط
- 3- تحديد مستلزمات وقيود المشروع:  
مثال: قد تكون مستلزمات مشروع إعداد الرواتب كما يلي:
  - العنصر البشري: محلل نظم و عدد إثنين مبرمجين.
  - الأجهزة: 10 أجهزة حاسب شخصي و عدد إثنين طابعة ليزيرية.
  - البرمجيات: VBASIC + Windows + SQLالقيود: في هذه الحالة تكون كما يلي:  
التكلفة: الميزانية محددة بقيمة 10000 دينار.  
الوقت: يجب ألا يزيد زمن المشروع عن 5 أشهر.
- 4- تجهيز جدول زمني للمشروع  
مثال: قد يكون الجدول الزمني لمشروع طوله 10 أشهر على النحو التالي:
  - التحليل شهران
  - التصميم شهران
  - البرمجة شهران
  - الاختبار أربع أشهر
- 5- تحديد وظائف المنظومة:  
مثال: وظائف منظومة المخازن يمكن أن تكون على النحو التالي:
  - اضافة أصناف جديدة للمخزون
  - الاستفسار عن صنف معين
  - تحديث بيانات صنف معين
  - تكوين تقارير
- 6- تحديد معايير المشروع:  
مثال: قد تكون بعض معايير منظومة مستشفى كما يلي:
  - استخدام لغة البيسك المرئية ولغة الاستفسار SQL كأدوات برمجية.
  - يجب أن تعمل المنظومة في بيئة شبكة.
  - يجب ألا تزيد عدد أسطر كل برنامج فرعي عن 30.
- 7- تعريف خصائص الجودة المطلوبة:  
مثال: منظومة المصرف يجب ان تتمتع بالاعتمادية , والكفاءة , وقابلية الصيانة , و سهولة الاستعمال.

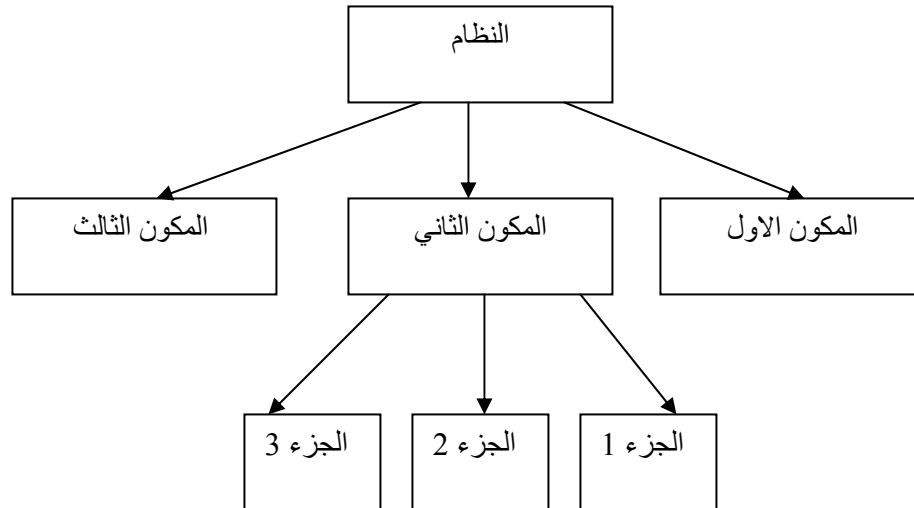
#### 4.2 مخطط تركيبة تجزئة العمل:

هذه الأداة تسمى Work Break\_down Structure و إختصارها (WBS) وهي عبارة عن أداة تخطيط تبين حدود المشروع كمجموعة من المهام بتركيبة هرمية من أعلى الى أسفل. في الشكل 7 التالي نجد ان شكلاً عاماً ل WBS حيث يقسم المشروع أولاً إلى عدد من النشاطات , ثم يمكن أن يقسم كل نشاط إلى مجموعة مهام.



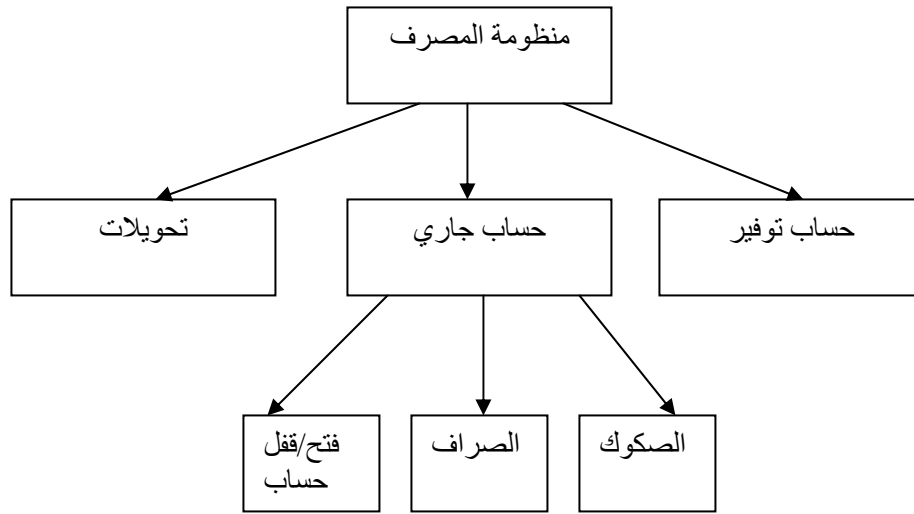
شكل 7: مخطط WBS لتجزئة عمل المشروع

يمكن ان يستعمل مخطط WBS لرسم أجزاء المنظومة بحيث يبين مجال النظام كمجموعة من الأجزاء الفردية بطريقة من أعلى إلى أسفل. الشكل 8 يوضح تقسيم المنظومة المطلوب إعدادها إلى مجموعة من المكونات ثم إلى مكونات فرعية.



شكل 8: مخطط WBS لتقسيم المنظومة

مثال: ارسم مخطط من نوع WBS لمنظومة مصرفية ذات ثلاث مكونات فقط هي (حساب التوفير , حساب جاري , تحويلات) حيث حدود المنظومة يخص جزئية الحساب الجاري فقط وهي ذات 3 مكونات جزئية هي (الصكوك , الصراف , فتح و قفل حساب). انظر شكل 9.



شكل 8: مثال على مخطط WBS لمنظومة مصرف

مميزات مخطط WBS:

- 1- أداة تخطيط قوية لتوضيح حدود المشروع.
- 2- أداة متابعة قوية لإعداد تقارير حول سير المشروع للإدارة والزبون.
- 3- يمكن استخدام WBS لتقدير التكلفة ومدة المشروع.
- 4- توفر آلية لقياس الأداء.

#### 4.3 الجدولة وتقدير زمن المشروع:

الجدولة وتقدير الوقت للمشاريع يعتبران الجزء الصعب في التخطيط. ويعتمد التقدير الزمني لأي مشروع على الفترة الزمنية التي يستغرقها إنجاز كل مهمة. عندما نقدر الزمن يجب ان نستعمل أيام العمل كوحدة قياس للزمن. لاحظ أن ساعات الدوام قد تختلف من مكان إلى آخر. وفي هذه الحالة تستخدم الساعة كوحدة قياس للزمن.

أرشادات بخصوص تقدير الوقت:

- يجب التعرف على المهام التي يمكن أن تنجز على التوازي.
- استعمل أقل قوى ممكنة.
- خذ في الاعتبار أن بعض الناس قد يتركوا العمل أو يتعرضوا للمرض.
- يتم التقدير كما لو سيكون كل شيء على ما يرام , ثم يتم زيادة 10% مثلا للمشاكل الغير متوقعة

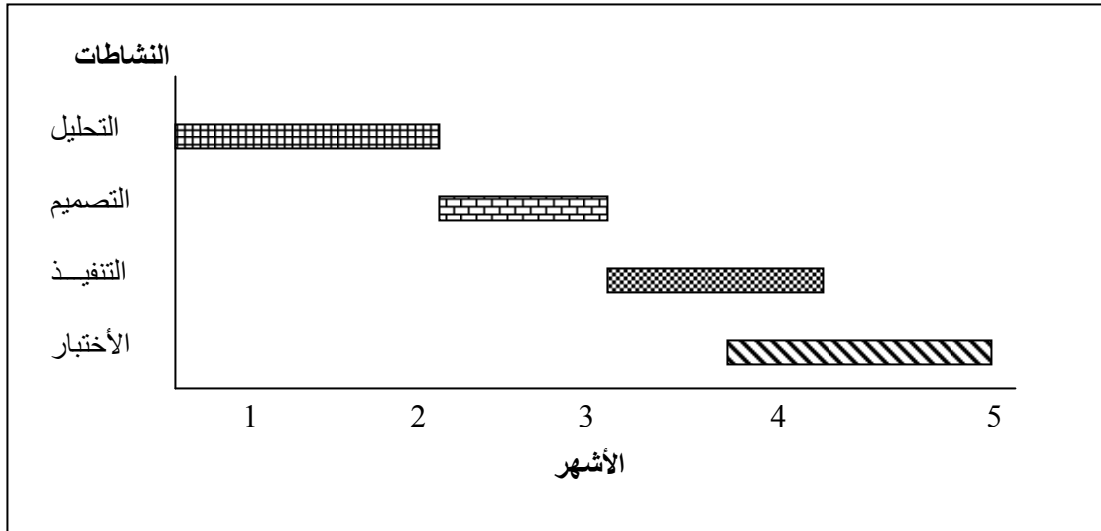
مثلا إذا كان الجزء البرمجي يستغرق يوما واحدا لإنهائه , وكان لدينا 30 جزء برمجيا في المشروع , فإننا نقدر الزمن لكتابة كل الأجزاء بمدة 30 يوما. يضاف إلى ذلك 3 أيام (أي 10%) للأحداث غير المتوقعة (مرض , إجازة , ....) فيصبح التقدير هو 33 يوما. يجب ملاحظة ان مواصفات المتطلبات والتصميم تتطلب ضعف الوقت اللازم لكتابة شفرة البرنامج.

#### 4.4 أدوات الجدولة:

أدوات الجدولة يجب أن تبين:

- المهام
- العلاقة بين المهام
- العنصر البشري المعين لكل مهمة

المخططات إما أن تنجز يدوياً , أو بأدوات خاصة مثل برنامج "مدير المشاريع" الذي أعدته شركة مايكروسوفت. هذه الأدوات تفترض أن المشروع يتكون من مجموعة من النشاطات أو المهام المعتمدة على بعض. فمنها مهام لا يمكن أن تبدأ قبل أن تنتهي مهام أخرى , ولكن بعض المهام يمكن أن تنجز بالتوازي مع نشاطات أخرى. في عملية الجدولة , يستخدم مدير و المشاريع أدوات مثل:



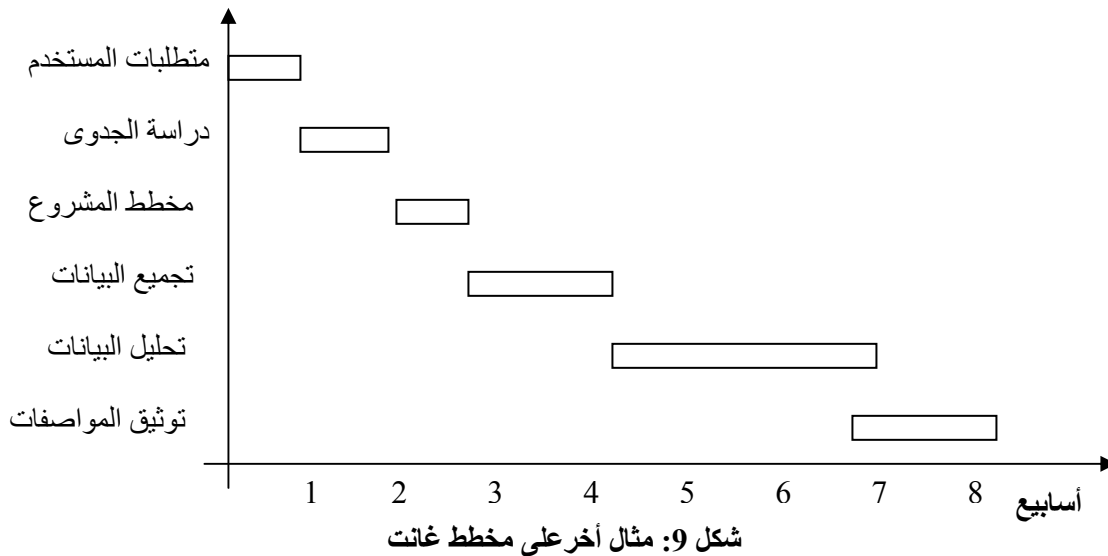
هذه الأدوات تستخدم في تخطيط ومتابعة المشاريع.

#### 4.5 مخطط غانت Gantt Chart:

مخطط غانت هو أداة تخطيط رسومية تستخدم لعرض مهام المشروع ومدتها الزمنية. يمكن رسم مخطط غانت يدوياً أو بواسطة برنامج رسم. المحور الأفقي للمخطط يعبر عن الفترة الزمنية (ممثلاً بوحدة الأيام أو الأشهر أو الأسابيع) أما المحور العمودي فيمثل المراحل (أو المهام أو النشاطات) كما في الشكل 9.

#### شكل 9: مخطط غانت

كما نرى في الشكل 9 فإن مخطط غانت يتسم بالمراحل المتداخلة مثل مرحلتى التنفيذ والاختبار. في الشكل 10 نرى مثلاً على مخطط يدوي لمشروع يتكون من 6 نشاطات حيث نعبّر عن مدة كل نشاط بالأسابيع.



شكل 9: مثال آخر على مخطط غانت

لاحظ أن في المثال الثاني لمخطط غانت انه يشمل على نشاطات مرحلتى التخطيط والتحليل فقط. لو مثلنا مهام النشاطات الأخرى لأصبح المخطط طويلاً جداً. لحل هذه المشكلة يمكن استخدام جدول الخط الزمني ليمثل مهام المشروع بالتفصيل. مثلاً يمكن أن يبين الفترة الزمنية بالأيام بدلاً من أسابيع و أشهر.

#### 4.6 جدول الخط الزمني Time-Line Table:

هذا الجدول هو عبارة عن أداة تخطيط و متابعة ذات صفوف للمهام , و أعمدة لتواريخ البداية والنهاية المتوقعة والفعلية , مع امكانية إضافة عمود لمن يقومون بكل مهمة.

المهمة	البداية المتوقعة	النهاية المتوقعة	البداية الفعلية	النهاية الفعلية

جدول الخط الزمني Time-Line Table

يمكن إضافة عمودين لهذا الجدول الأول للتكلفة والثاني للتكلفة الفعلية. في جدول الخط الزمني نقوم بملء أعمدة البداية والنهاية المتوقعة قبل بداية المشروع. البداية والنهاية الفعلية لكل مهمة يمكن ان تعرف فقد أثناء سير عمل المشروع. يقوم مدير المشروع بملأ البيانات الفعلية عند وقوعها. هذه العملية تساعد في إعادة التخطيط للمشروع ومعرفة أسباب التأخير.

#### 4.7 تحليل المخاطر Risk Analysis:

تحليل المخاطر اصبح من المواضيع المهمة في تخطيط المشاريع , فهو ذو علاقة وطيدة مع نجاح المشروع أو فشله. ومع أن أدوات وطرق إعداد المشاريع قد تطورت تطوراً هائلاً في الوقت الحالي , إلا أن المشاكل و المخاطر قد تكون موجودة في المشروع مما يؤثر سلباً عليه. هذه المشاكل أو المخاطر قد تكون ذات علاقة بجدولة المشروع أو تقدير التكاليف أو الجودة. بعض من هذه المشاكل قد لا يمكن تجنبها و لكن يمكن التقليل من مخاطرها. المخاطر هي عبارة عن مشكلة يمكن أن تسبب في خسارة أو تهديد نجاح المشروع. بعبارة أخرى , إنها مشكلة كامنة سيكون لها تأثير سيء على المشروع. هذا التأثير قد يكون على:

- تكلفة المشروع.
- الجدول الزمني للمشروع.
- جودة المنتج.

تحليل المخاطر هو عبارة عن عملية تحديد و معالجة المخاطر لمشروع ما. لماذا يجب علينا أن نعرف مخاطر المشروع قبل تنفيذه؟

أولاً لأن المخاطر تهدد نجاح المشروع.

ثانياً لأن هذه المخاطر المحتملة إذا تحققت فإن تسليم المشروع سيتأخر وتكلفته ستزداد. لإنجاز تحليل المخاطر يجب أن:

- نعد قائمة بالمخاطر.
- نحدد احتمال كل مخاطرة (نسبة حدوث المخاطرة).
- نحدد توابع كل مخاطرة.
- نحدد إستراتيجية الحل لكل مخاطرة.

الجدول التالي يبين نموذجاً لتحليل المخاطر:

المخاطرة	الاحتمال	التأثير	الحل
تغير المتطلبات	80%	حرج	• استخدام العرض المبدئي
التدريب على الأدوات غير كافي	70%	سهل التدبير	• التدريب. • التركيز على الدراسة.
غياب العاملين بالمشروع	60%	حرج	• المزيد من التدريب • تحسين اوضاع العاملين
التأخر في تسليم المنظومة	60%	حرج	• إعادة استعمال الوحدات البرمجية. • استعمال أدوات مناسبة لتقدير الزمن.

## 5. أدوات تحليل النظم System Analysis Tools.

### 5.1 مخطط انسياب البيانات (DFD) Data Flow Diagram

يعتبر مخطط انسياب البيانات (DFD) إحدى الأدوات المهمة للحصول على وصف مفصل للنظام القائم والمقترح. فهي أداة تحليل تستخدم لتوضيح انسياب البيانات خلال النظام , وبين النظام والبيئة المحيطة به . وهي تستعمل الرسومات لتوضيح مصدر البيانات ومعالجتها وأماكن تخزينها.

ملاحظة: هناك فرق بين مخطط انسياب البيانات (DFD) والمخطط الانسيابي. الأول يبين انسياب البيانات والعمليات في نظام ما بينما المخطط الانسيابي يبين انسياب التحكم في العمليات علي البيانات. بعبارة اخرى مخطط انسياب البيانات هو أداة تحليل بينما المخطط الانسيابي هو أداة تصميم.

خصائص مخطط انسياب البيانات:

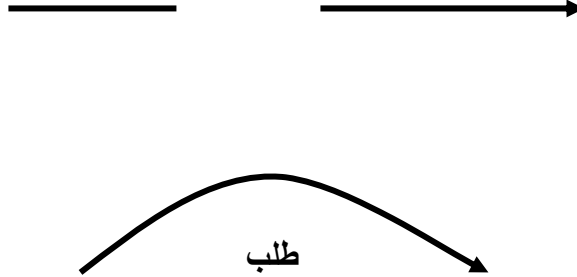
- يعتبر مخطط انسياب البيانات وسيلة اتصال بين المحلل و الزبون او المستخدم والغرض منها في الأساس هو الوصول الى فهم افضل لمسار انسياب البيانات في النظام القائم.
- ويمكن استعماله كأساس لمرحلة التصميم.
- يعتبر وسيلة اتصال بين المحلل والمصمم.
- المخطط يصف ماذا يحدث وليس كيف يحدث.
- تستخدم لتوضيح العلاقة بين الإدخال والإخراج و المعالجة ومخازن البيانات التي تكون النظام.

#### أشكال و رموز المخطط:

رموز (DFD) هي عبارة عن أشكال هندسية مثل الدوائر والمستطيلات. ويوجد العديد من البرمجيات التي تساعد في اعداد ورسم المخططات على الحاسوب مثل برنامج VISIO لشركة ميكروسوفت. و في الجزء التالي نستعرض هذه الرموز:

#### سهم انسياب البيانات:

السهم يبين تدفق البيانات في مخطط (DFD) وهو يمكن ان يكون خط مستقيم او منحنى وفي كلتا الحالتين يجب ان يبين اتجاه تدفق البيانات.



#### رمز العملية (أو المعالجة):

وهو يصف عملية معالجة على البيانات. ويستخدم لتمثيل عملية المعالجة شكل مستطيل أركانها مقوسة أو شكل بيضاوي أو شكل مستطيل.



#### مخزن البيانات:

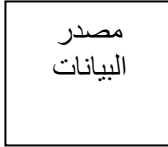
وهو مكان تخزين البيانات , يأخذ شكل صندوق مفتوح من جهة واحدة او جهتين كما يلي:



يمثل مخزن البيانات مكاناً تكون فيه البيانات في حالة سكون. وهو قد يكون ملف حاسوب (قاعدة بيانات) أو ملف يدوي (مثل بطاقات الفهرس).

### صندوق المصدر أو النهاية:

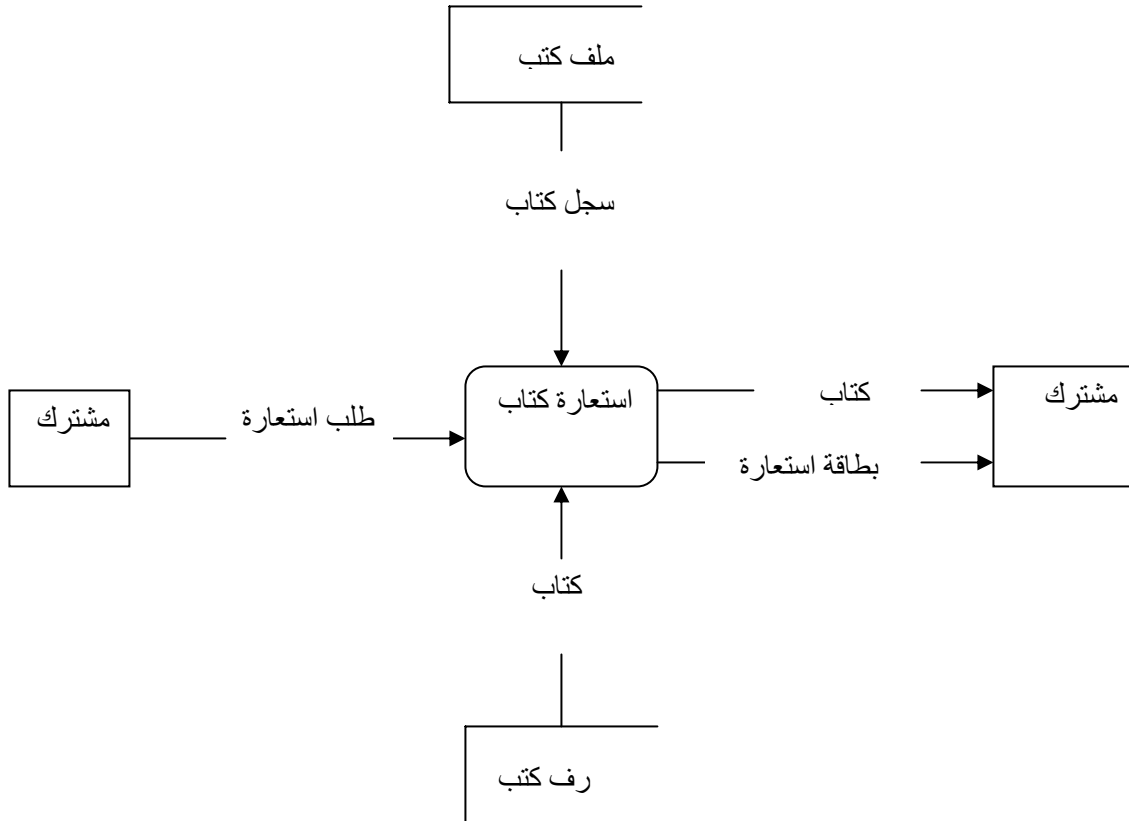
صندوق المصدر أو النهاية هو عبارة عن كيان خارجي تنطلق منه البيانات (ويسمى في هذه الحالة مصدر) أو تنتهي عنده المعلومة (ويسمى في هذه الحالة غاية). كلا المصدر والغاية يعتبران نظامين مستقلين. ويستخدم شكل مربع لتمثيل مصدر ونهاية البيانات كما يلي:



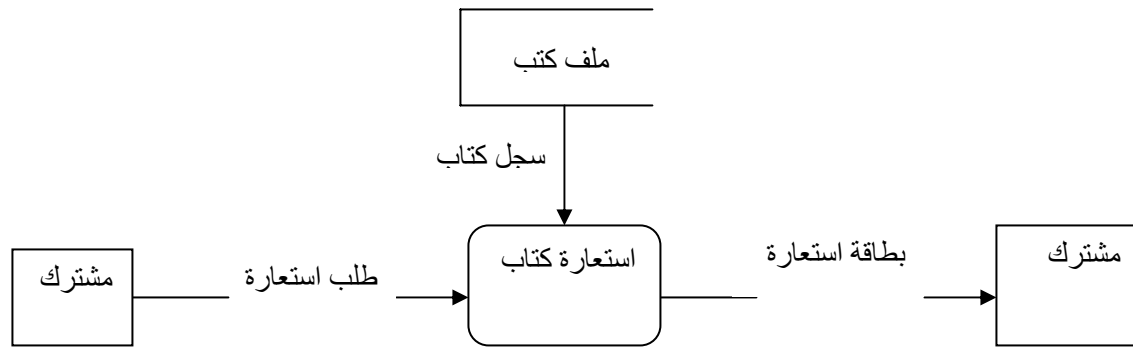
### أنواع مخطط انسياب البيانات:

مخطط انسياب البيانات الفيزيائي: هذا النوع يبين النظام الحالي بأشياءه الفيزيائي (الملموسة) كما يراها المستخدم. مخطط انسياب البيانات المنطقي: هذا النوع يبين النظام كما يراه محلل النظام والمصمم أي بدون عناصر فيزيائية (ملموسة).

كقاعدة عامة يجب ان يتم رسم المخطط الفيزيائي قبل المنطقي. المثال التالي يبين مخطط انسياب البيانات الفيزيائي والمنطقي لعملية الاستعارة في نظام مكتبة.



شكل 10: مخطط فيزيائي



شكل 11: مخطط منطقي

لاحظ الفرق بين الشكلين أعلاه. في المخطط الفيزيائي نرى تدفق الأشياء المادية (الكتاب) بينما يظهر المخطط المنطقي تدفق البيانات عن الكتاب فقط.

لرسم المخطط بالكامل يجب عمل المخططات الآتية تباعاً:

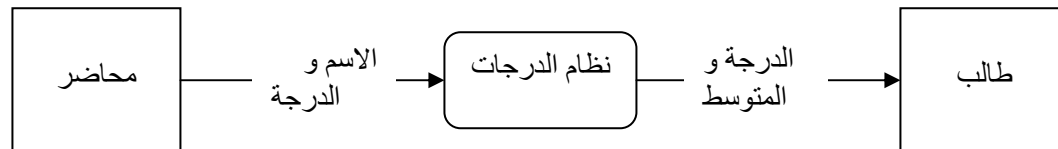
1. المخطط البيئي
2. مخطط فيزيائي للنظام الحالي
3. مخطط منطقي للنظام الجديد

#### كيفية رسم مخطط انسياب البيانات:

يقوم المحلل بالخطوات التالية قبل الحصول على المخطط النهائي.

1. المخطط البيئي: يبين هذا المخطط النظام بأكمله كعملية (معالجة) واحدة متصلة بالكينونات الخارجية. وبذلك فهو يعتبر أوسع صورة للنظام. ويبين حدود النظام ونطاقه. لاحظ أن المخطط البيئي يمثل صورة شاملة لنظام مبسطه و يمكن فهم العمليات داخل النظام بسهولة.

مثال 1: ارسم المخطط البيئي لمنظومة درجات الطلبة في قسم الدراسة والامتحانات حيث يستطيع الطالب عن طريقها الاطلاع على متوسط درجاته بعد أن يقوم الأستاذ بإدخالها.



شكل 12: المخطط البيئي لنظام درجات الطلبة

من خلال الشكل والذي يمثل المخطط البيئي لمنظومة درجات الطلبة يمكن معرفة التالي:

- مصدر البيانات هو المدرس.
- مستهدف البيانات هو الطالب.
- النظام نفسه يمثل العملية (المعالجة) الأساسية.

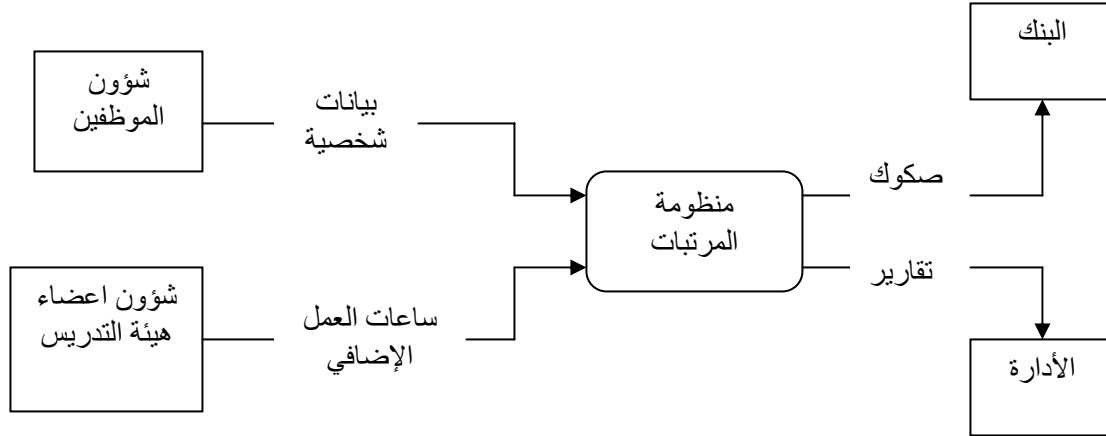


مثال 2: ارسم المخطط البيئي لمنظومة الرواتب حيث يشمل المدخلات التالية:

- بيانات شخصية عن الموظف من قسم شؤون الموظفين.
- بيانات عن عضو هيئة التدريس (المحاضر) من قسم شؤون اعضاء هيئة التدريس.
- عدد الساعات الإضافية من قسم شؤون اعضاء هيئة التدريس.

أما المخرجات فهي تشمل التالي:

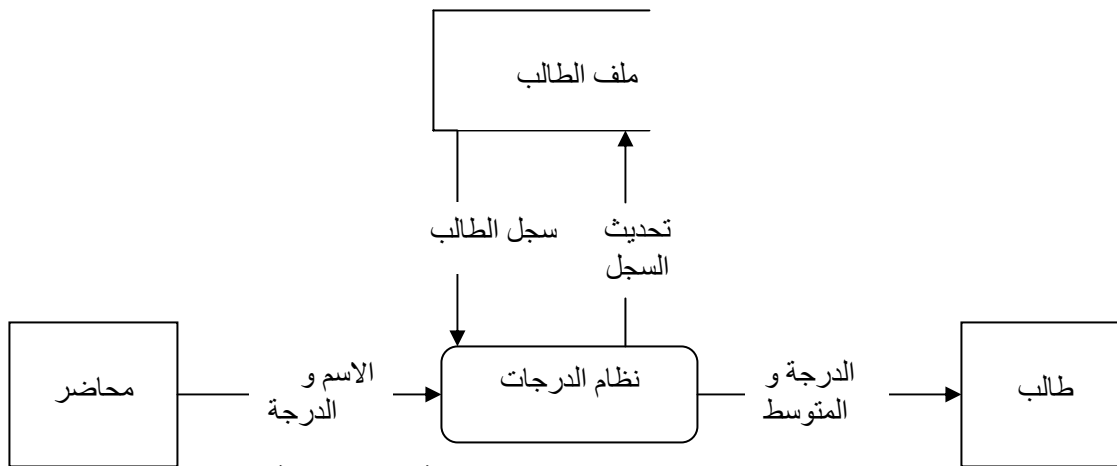
- صكوك المرتبات موجهة للمصارف.
- تقارير للإدارة



شكل 13: المخطط البيئي لمنظومة المرتبات

2. مخطط المستوى 0: في هذا المرحلة المخطط يوضح تفاصيل أكثر من المخطط البيئي. وتتمثل التفاصيل أساسا في إضافة مخازن البيانات.

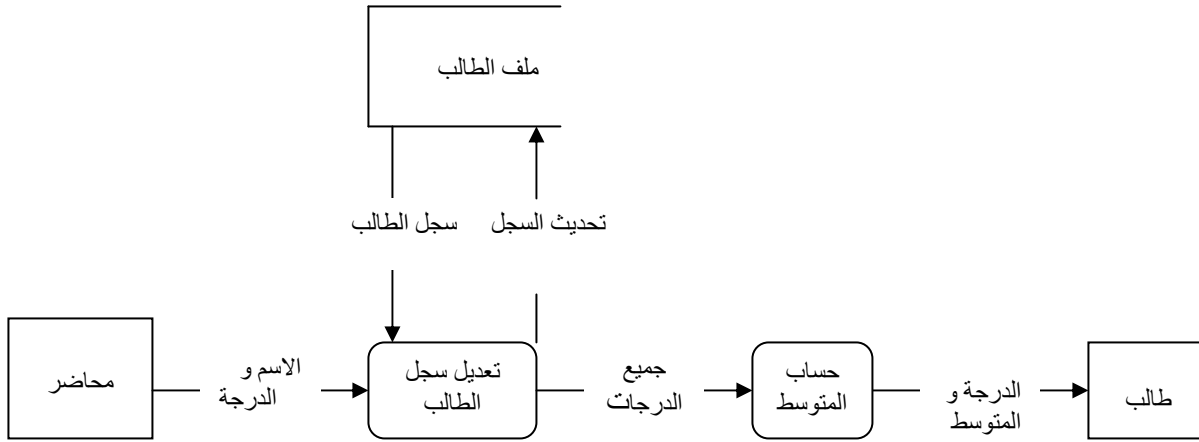
مثال 1: ارسم مخطط المستوى 0 للمثال أعلاه حيث يتم تخزين البيانات كلها في ملف واحد يسمى ملف الطالب.



شكل 14: مخطط مستوى 0 لمنظومة درجات الطلبة

3. مخطط المستوى 1: تتجزأ العملية الرئيسية هنا إلى عمليات مفصلة لكي يتضح النظام بصورة أكثر. لذلك يسمى مخطط DFD أحياناً بمخطط الفقاعة حيث كل عملية (معالجة) تشبه الفقاعة التي تنقسم إلى فقاعات (عمليات) أصغر. مثال: ارسم مخطط المستوى 1 للمثال السابق (درجات الطلبة) حيث تقسم العملية الرئيسية إلى عمليتين فرعيتين (وظيفتين فرعيتين).

- عدّل في ملف الطالب
- احسب المتوسط

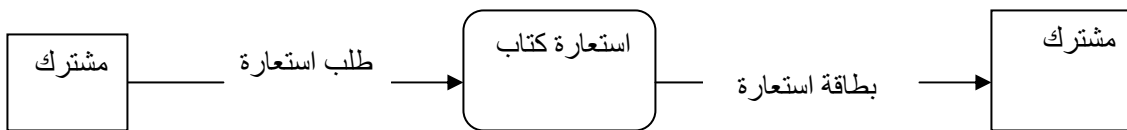


شكل 15: المستوى 1 لمنظومة درجات الطلبة.

شروط رسم مخطط انسياب البيانات:

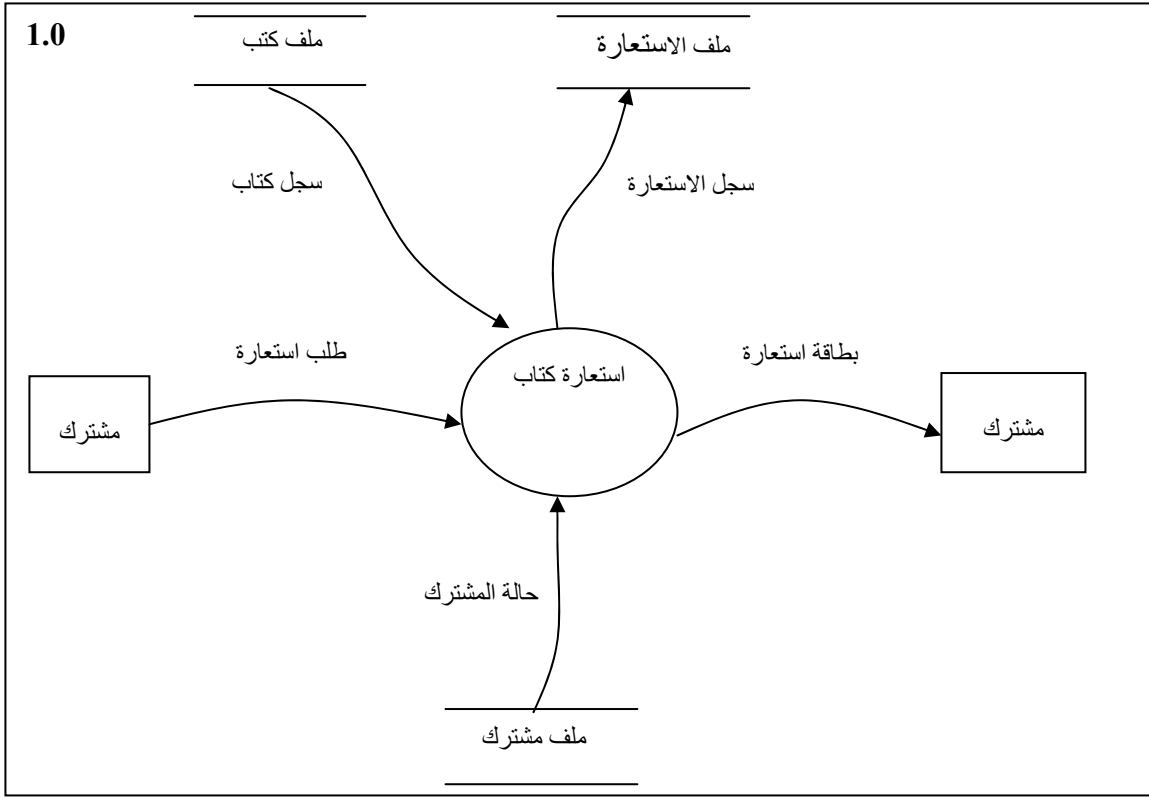
1. ابتعد عن التفاصيل في المخطط البيئي والمستوى 0.
2. ابدأ الرسم من اليسار إلى اليمين ومن الأعلى إلى الأسفل.
3. عنون كل صندوق معالجة بفعل أمر.
4. لا تبيين المعدات أو الأشياء المادية في المخطط المنطقي.
5. يمكن تكرار مخازن البيانات و الكينونات الخارجية لغرض تجنب التقاطعات والحصول على وضوح أفضل.
6. يحتوي المستوى 1 على عدد من العمليات (من 2 إلى 7).
7. عنون انسياب البيانات على كل سهم.
8. ابدأ برسم أولي ثم اعمل مراجعة مع زميلك او مع المستخدم لتأكد من أن المخطط يمثل فعلاً النظام تحت الدراسة.
9. يمكن أن يكون للمخطط مستويات مختلفة من حيث التفاصيل (أي التجزئة).
10. بعض العمليات تحتاج إلى تجزئة وعادة ما تحتاج إلى عملية ترقيم لتوضيح مستوى التجزئة. تسمى هذه الطريقة بتحديد المستويات.
11. هذا المخطط لا يبين حلقات دوران او حسابات أو قرارات.

مثال على منظومة المكتبة: في هذه المنظومة يقوم المشترك بطلب استعارة كتاب معين مستخدماً نموذج الاستعارة وعليه تكون أول معالجة هي البحث عن الكتاب. ويتم البحث في ملف الكتب. في حالة وجود الكتاب تتم إعارة الكتاب للزبون ويسجل ذلك في بطاقة الإعارة.



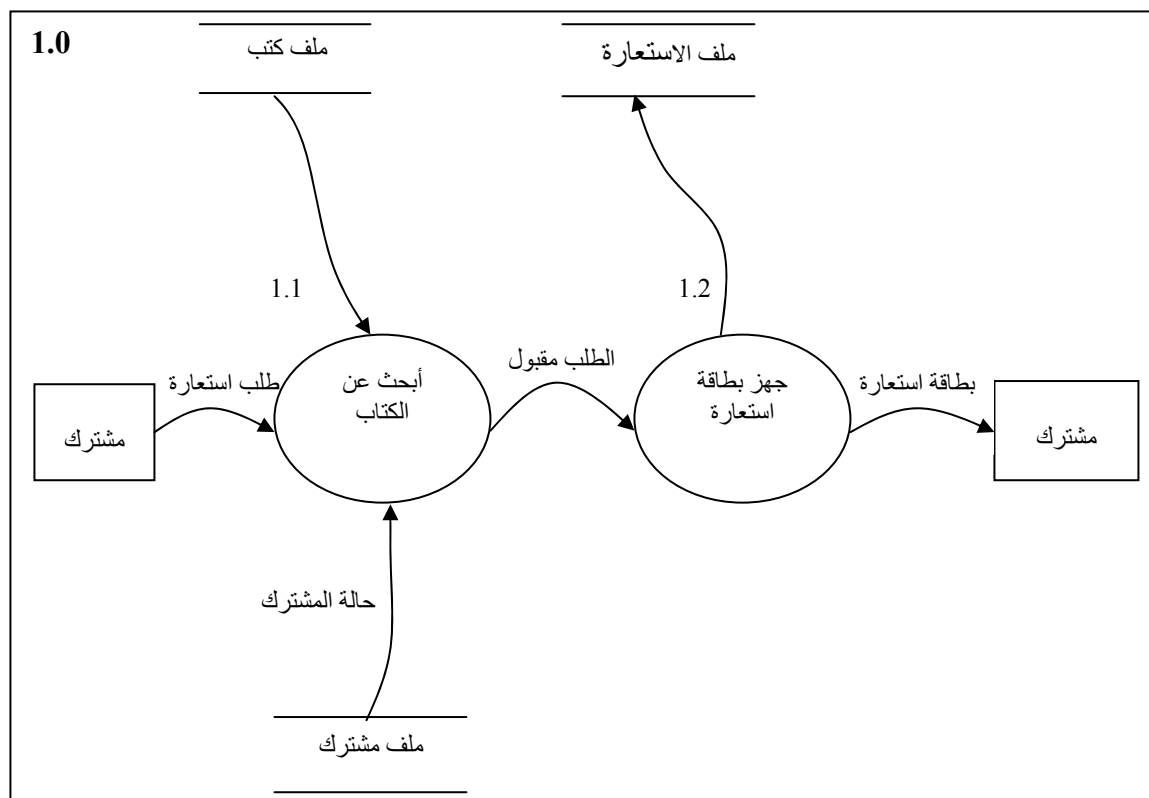
شكل 16: المخطط البيئي لنظام المكتبة

المستوى 0 في هذا المثال موضح في شكل 17 حيث قمنا بتغيير العملية (منظومة المكتبة) إلى (معالجة طلبية) و أضفنا 3 مخازن بيانات (ملفات). في هذا المخطط افترضنا وجود الكتاب المطلوب وذلك لغرض تبسيط الشكل. لا حظ اننا عنونا هذه المعالجة بالرقم 1.0



شكل 17: المستوى 0 لمخطط منظومة المكتبة

الان نحاول تقسيم المستوى 0 من منظومة المكتبة كما في الشكل 18 الى مستوى 1. العملية السابقة 1.0 (معالجة طلبية) تجزأ إلى عمليتين فرعيتين فقط هما:  
 1.1 التأكد من وجود الكتاب  
 1.2 إعداد بطاقة استعارة  
 لاحظ ان المستوى 1 في الشكل 18 يعطي فهماً أوضح للنظام تحت الدراسة.



شكل 18: المستوى 1 لنظام المكتبة

من خلال شكل 18 نلاحظ الحاجة الى المزيد من التفاصيل. مثلاً قد نحتاج الى العمليات التالية لغرض تفصيل انسياب البيانات للنظام بدقة أكثر:

1. تحقق من وجود الكتاب
2. إذا لم يكن موجود اعرض رسالة بذلك
3. جهّز سجل زبون في حالة عدم وجوده
4. جهّز سجل استعارة
5. جهّز بطاقة استعارة

## 5.2 الانجليزية المركبة (Structured English (SE).

الانجليزية المركبة هي أداة تستخدم في التحليل لوصف سياسة عمل أو طريقة تنفيذ العملية في مخطط انسياب البيانات بلغة انجليزية دقيقة نصياً. لا حظ ان هذا المصطلح SE يمكن ان يطلق على أي لغة أخرى غير الانجليزية مثلاً اللغة العربية. وفي هذه الحالة نسميها العربية المركبة وهي ضرورية في حالة أن المستخدم لا يتقن اللغة الانجليزية.

### استعمالات الانجليزية المركبة.

تستخدم الانجليزية المركبة في المسائل التي تدمج بين سلسلة من الأعمال و الحسابات مع اختبارات وتكرارات.

### قواعد كتابة الانجليزية المركبة.

1. وضح المنطق كمجموعة تعليمات خطوة خطوة بما في ذلك الحسابات وتراكيب القرارات.
2. استعمل مصطلحات من قاموس البيانات و مخطط انسياب البيانات.
3. يمكن التعبير عن منطق العمليات في النظام بمجموعة من التراكيب التتابعية.
4. استعمل التنسيق والمصطلحات الخاصة في كتابة جمل انجليزية لوصف منطق المعالجة و السياسات.
5. استخدم حروفاً كبيرة في اسماء الأوامر مثل IF , THEN , ELSE , REPEAT , WHILE , CASE .
6. استعمل أفعال مألوفة و مناسبة.

### مصطلحات الانجليزية ال مركبة.

القائمة التالية تحتوي على أفعال مناسبة تستعمل في الانجليزية المركبة:

حدد	Determine	احسب	Compute
افحص	Check	تحقق	Verify
جهّز	Prepare	عدل	Update
كوّن	Generate	أدخل	Enter
اطبع	Print	اقرأ	Read
شكّل	Format	اعرض	Display
قارن	Compare	استرجع	Retrieve
أنقل	Move	حرّر	Edit
صنّف(رتب)	Sort	أضف	Add
		أدمج	Merge

المصطلحات المعرفة في قاموس البيانات مثل:

الرصيد الحالي	Current-balance
سعر الصنف	Item-price
إسم الطالب	Student -name
عدد الطلبة	Number-of- student
تاريخ آخر معاملة	Date-of-last-transaction

الكلمات المحجوزة (اسماء اوامر لغة البرمجة) المستعملة في البرمجة الهيكلية مثل:

لو- فإن - وإلا	IF – THEN - ELSE
في حالة أن	CASE
أنجز – طالما	DO – WHILE
كرر – حتى	REPEAT – UNTIL

**مثال 1:** اكتب الانجليزية المركبة لحساب أجور المستخدمين بضرب عدد الساعات في أجر الساعة الواحدة إذا كان عدد ساعات العمل 40 أو أقل وإذا زاد عدد الساعات على 40 فإن سعر ساعة العمل الإضافي تساوي سعر ساعة ونصف من العمل العادي.

- أدخل عدد ساعات الشغل و أجر الساعة الواحدة
- لو عدد الساعات الشغل أقل من أو يساوي 40
  - فأحسب الأجر بضرب عدد الساعات في أجر الساعة الواحدة
- وإلا فأحسب المقدار 1 بضرب أجر الساعة الواحدة في 40
  - واحسب المقدار 2 بضرب عدد ساعات الشغل الزائدة عن 40 في 1.5 \* أجر الساعة الواحدة.
  - احسب الأجر بجمع المقدار 1 مع المقدار 2.

**مثال 2:** اكتب الانجليزية المركبة لحسب نسبة التخفيض للزبائن الذين يتمتعون بميزة التخفيض.

- لو نوع الزبون = "زبون قديم" فإن
  - لو مقدار المبيعات أكبر من 10000
    - فإن التخفيض = 10%
  - وإلا فالتخفيض = 8%
- وإلا
  - لو مقدار المبيعات أكبر من 10000
    - فإن التخفيض = 5%
  - وإلا فالتخفيض = 3%

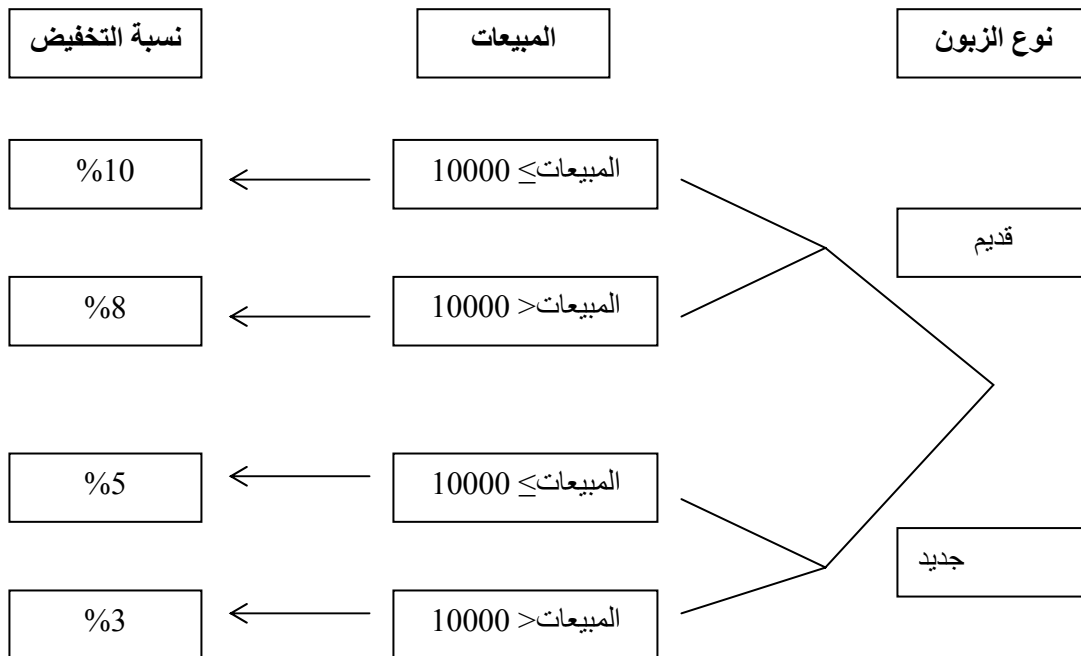
### 5.3 شجرة القرار Decision Tree.

شجرة القرار هي عبارة عن أداة تحليل تستخدم لتعريف سياسة أو علاقة في نظام. وعادة ما يتم استعمال مخطط لتوضيح الأفعال التي تنفذ في الحالات (الشروط) المختلفة.

استخدام شجرة القرار:

1. تتعامل شجرة القرار مع السياسات ذات التفرع المتعدد مثل حساب تخفيضات المبيعات ومكافآت الإنتاج وحساب الأجور.
2. تستخدم عندما يكون عدد الافعال صغيراً ويمكن توضيح كل الاحتمالات.
3. من الأفضل استخدامها للتحقق من المنطق وعندما تكون القرارات غير معقدة.

**مثال 3:** هذا المثال يخدم نفس الغاية لوصف سياسة التخفيض في منظومة المبيعات في المثال 2 السابق.



## 5.4 جدول القرار Decision Table:

هذه الطريقة يمكن أن تكون بديلة لشجرة القرار لغرض تمثيل سياسة النظام بطريقة مختصرة. جدول القرار هو عبارة عن أداة تحليل وتصميم تبين المنطق الذي يربط بين الحالات و الأفعال. وهو عادة يكون على هيئة مصفوفة (صفوف واعمدة) تستخدم لتعريف سياسة عمل النظام.

استخدام جدول القرار:

1. تستخدم عندما تكون الأفعال معتمدة على مجموعة كبيرة من الشروط تؤخذ على مجموعة جزئية متعددة.
2. تستخدم في تطبيقات الادارة و التحكم.
3. كما يمكن استعمالها في التطبيقات التي تتعلق بالآتي:
  - a. التصنيع
  - b. التأمين
  - c. المالية
  - d. البحث والتطوير

مكونات جدول القرار:

**الشرط:** حقائق أو أحداث تقرر الأفعال التي يجب أن تنفذ.

**الأفعال:** العمليات أو المعالجة التي تنفذ تحت شروط معينة.

**قواعد القرار:** تعبر عن العلاقات بين الشروط والأفعال.

الجدول التالي يمثل الشكل العام لجدول القرار:

قواعد القرار					
					شرط
					أفعال

جدول القرار

مثال 1: المطلوب إعداد جدول القرار لحساب مرتب موظف حيث المرتب يساوي مجموع المرتب الأساسي ومكافأة العمل الاضافي وهي تحسب فقط إذا كانت ساعات العمل أكبر من 40 ساعة.

قواعد القرار			الشروط
3	2	1	
لا	لا	نعم	ساعات العمل > 40
لا	نعم	لا	ساعات العمل = 40
نعم	لا	لا	ساعات العمل < 40
			أفعال
X	X	X	احسب المرتب الأساسي
X	-	-	احسب العمل الإضافي

الجدول أعلاه يمثل استعمال جدول القرار لتوضيح سياسة حساب مرتب موظف. لاحظ في هذا الجدول استعمال الرمز (X) يعني قابل للتطبيق والرمز (-) يعني غير قابل للتطبيق. لاحظ أيضاً أن عدد قواعد القرار يعتمد على كل من عدد الشروط وعدد الأفعال.

في المثال المذكور أعلاه لدينا 3 شروط وفعلين اثنين وثلاثة قواعد قرار. ولكن ليس دائماً يتساوى عدد القواعد وعدد الحالات. المثال التالي يبين ان عدد قواعد القرار يمكن ان يكون أكبر من عدد الحالات.

مثال 2: مطلوب اعداد جدول قرار لسياسة شركة مبيعات تعطي معاملة خاصة للزبائن الذين يتوفر فيهم شرطان من الشروط التالية:

- قيمة المشتريات تتجاوز 10000 دينار.
- الدفع في الوقت المحدد.
- زبون لأكثر من 3 سنوات.

في هذا المثال نلاحظ ان الحالات الثلاثة مستقلة عن بعضها البعض. اي أنها يمكن ان تتوفر كلها في زبون واحد أو بعض منها. في الواقع , عدد قواعد القرار في هذا المثال هو 2 أس 3 أي يساوي 8 قواعد قرار ( $2^3=8$ ).  
بصور عامة , إذا كان لدينا n حالة مستقلة , فإن

$$\text{عدد قواعد القرار} = 2^n$$

وبذلك يكون جدول القرار كما يلي:

الشروط								قواعد القرار			
المشتريات $10000 <$	لا	نعم	لا	نعم	لا	نعم	لا	لا	نعم	لا	نعم
الدفع في الوقت المحدد	لا	لا	نعم	نعم	لا	نعم	لا	لا	نعم	نعم	نعم
السنوات $3 <$	لا	لا	لا	لا	لا	لا	لا	لا	نعم	نعم	نعم
الأفعال											
سعر عادي	X	X	X	-	X	-	X	X	-	-	-
سعر خاص	-	-	-	-	-	-	-	X	X	X	X

مزايا جدول القرار:

- وسيلة لوصف سياسة عمل النظام.
- تساعد في التصميم و التشفير.
- يكمن عيبها في صعوبة تكوينها خاصة لمن يستعملها لأول مرة.

مقارنة بين شجرة القرار و جدول القرار والانجليزية المركبة.

معامل المقارنة	شجرة القرار	جدول القرار	الإنجليزية المركبة
بساطة الاستعمال	جيد جداً	ضعيف	متوسط
البرمجة	متوسط	جيد	جيد جداً
قابلية التغيير	متوسط	ضعيف	جيد
التركيب المنطقي	جيد جداً	متوسط	جيد
تحقق المستخدم	جيد	ضعيف	متوسط

نستطيع أن نقول أن جداول القرار تستخدم عندما يوجد عدد كبير من الأفعال والحالات. بينما تستخدم شجرة القرار عندما يوجد بسيط من الافعال و الحالات ويكون كل شرط محتمل.

دراسة مقارنة لدواعي استخدام أدوات التحليل لوصف سياسة عمل المنظومة.

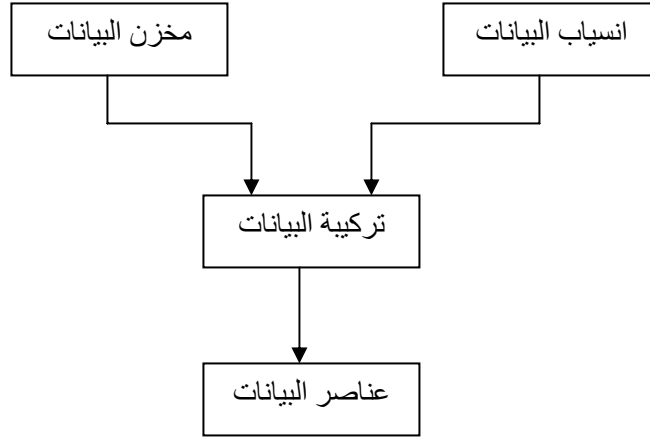
شجرة القرار	جدول القرار	الإنجليزية المركبة
<ul style="list-style-type: none"> <li>• تستعمل للقرارات البسيطة</li> <li>• تستعمل للتأكيد المنطقي</li> </ul>	<ul style="list-style-type: none"> <li>• تستعمل للمسائل ذات المنطق المركب من عدة شروط</li> </ul>	<ul style="list-style-type: none"> <li>• تستعمل للمسائل التي تحتوي على الحلقات و القرارات والسلسلة من الأفعال</li> </ul>

من هذا الجدول نستخلص أنه إذا كانت المسألة لدينا بسيطة يمكننا أن نستعمل شجرة القرار. أما إذا كانت المسألة معقدة فنستعمل إما جدول القرار أو الانجليزية المركبة.



#### 5.4 قاموس البيانات Data Dictionary:

قاموس البيانات هو أداة تحليل وتصميم تستخدم لحفظ انسياب البيانات و مخازن بيانات (ملفات) في مخطط انسياب البيانات DFD. بتعبير آخر فإن قاموس البيانات هو أداة تستخدم لتعريف وتسجيل كل عنصر بيانات في المنظومة. كما يجب ان نفرق بين انسياب البيانات هو يمثل تركيبة البيانات وهي في حالة حركة داخل النظام. أما مخزن البيانات فهو يمثل تركيبة البيانات وهي في حالة سكون داخل النظام. ولهذا فإن انسياب البيانات و مخزن البيانات يعتبران اساس تركيبة البيانات , حيث تتكون تركيبة البيانات من عناصر البيانات كما هو موضح في شكل 19.



شكل 19: هرمية وصف البيانات

ملاحظات على قاموس البيانات:

1. قاموس البيانات يمكن كتابته على ورق أو آلياً كجزء من نظام قاعدة البيانات مثل أوراكل أو SQL ومن أدوات كيس Case.
2. لاحظ أن قاموس البيانات هو أساس تصميم قواعد البيانات التي يتم تجهيزها في مرحلة التصميم.
3. لاحظ أيضاً أن الجزء الرئيسي من قاموس البيانات هو عنصر البيانات (المتغير) الموجود في مخزن البيانات و انسياب البيانات.

#### خصائص عنصر البيانات:

يجب ان يحتوي عنصر البيانات على الخصائص التالية:

1. اسم عنصر البيانات: هو اسم وحيد أي غير مكرر لعنصر البيانات ويفضل أن يكون الاسم مفهوم ويدل على محتواه والغرض منه مثل (اسم الطالب). أمثلة لأسماء عناصر البيانات:

- اسم الزبون Customer-name
- رقم الزبون Customer-no
- العلاوة Allowance
- ضريبة الدخل Income-tax

2. الوصف: يعطي الوصف النصي لعنصر البيانات ليبين غرض استخدام هذا العنصر.

3. النطاق: نطاق عنصر البيانات ومعنى القيم. قد نأخذ عنصر البيانات أي قيمة خلال نطاق معين (مثلاً من 0 إلى 99999999.999 أو قد يأخذ قيمة واحدة من قيم محدده مثل الجدول التالي:

القيمة	المعنى
1	أعزب
2	متزوج
3	مطلق
4	أرملة

4. المصدر: يمكن أن يأتي عنصر البيانات من مصادر مختلفة مثل أداة إدخال (مثلاً يمكن إدخال رقم الصنف بضاعة من لوحة المفاتيح) أو من جداول (مثل جدول الراتب الأساسي) أو من عملية حساب (مثل حساب صافي المرتب).

5. أين يستخدم عنصر البيانات: هذا المصطلح يستعمل للإشارة إلى المكان الذي سيستعمل فيه عنصر البيانات. مثلاً قد يستعمل عنصر البيانات (سعر الصنف) في حساب الفاتورة أو تقرير المبيعات.

6. التخزين: قد يتم تخزين عنصر البيانات في ملف أو قد يكون للاستعمال المؤقت كوسيط مثل مجموع للحصول على متوسط حساب لمجموعة قيم (مثل متوسط درجات الطلبة).

7. النوع والطول: يجب أن يحدد قاموس البيانات نوع و طول عنصر البيانات ويكون النوع إما صحيحاً (بدون فاصلة عشرية) أو حقيقياً (بفاصلة عشرية) أو منطقياً (صح وخطأ) أو تاريخاً أو زمن أو حرف. أما الطول فهو حجم عنصر البيانات الذي يقاس عادة بعدد البايت فيه. مثلاً اسم الطالب يأخذ طول 30 حرفاً سعر الصنف 99999.99 أي خمسة أرقام حقيقية و رقمين عشريين.

8. القيمة الافتراضية لعنصر البيانات: هي القيمة المعرفة مسبقاً وذلك لغرض تقليل أخطاء الإدخال وتقليل وقت الإدخال. ويتم اختيارها عادة بحيث تكون هي القيمة الأكثر احتمالاً وذلك لغرض تقليل احتمال تغييرها. على سبيل المثال القيمة الافتراضية لتاريخ فتح الحساب بالمصرف من الأفضل أن تكون التاريخ الحالي (تاريخ الجهاز أثناء عملية ادخال البيانات) لأنه الأكثر احتمالاً.

9. هل يسمح بقيمة لاشيء (Null value): إذا كان عنصر البيانات إجبارياً فلا يسمح بقيمة (لاشيء) , ولكن إذا كان اختيارياً فيمكن وضع (لاشيء) كقيمة له. مثلاً إذا كان عنصر البيانات يستخدم كمفتاح للسجل مثل رقم الطالب فلا يسمح بقيمة (لاشيء) فيه.

لاحظ أن جميع الخصائص تستخدم لتوضيح كل عنصر بيانات لمحلل النظم لغرض تصميم قواعد البيانات والمبرمج لاستعمالها داخل شفرة المصدر.

في المثال التالي نوضح قاموس البيانات لملف الطلبة.

الوصف	الاسم	النوع	الطول
رقم الطالب (مفتاح رئيسي للملف)	Student_Number	صحيح Integer	5
الاسم رباعي	Student_Name	نص String	50
تاريخ الميلاد	Birth_Date	تاريخ Date	8

#### مزايا قاموس البيانات:

1. يستطيع المستخدم أن يحصل على قائمة بالبيانات الخاصة بالمنظومة المتوفرة.
2. يتمكن المحلل والمستخدم عن طريق قاموس البيانات من التفاهم والتواصل بشأن مفردات المنظومة.
3. هل مشكلة الغموض لمعنى المصطلحات بين المصمم و المبرمج.

## 6. العرض التجريبي Prototyping:

العرض التجريبي هو عبارة عن عملية أعداد وعرض منظومة كعينة للزبون تمثل المنظومة المطلوبة ولكن بشكل مبدئي. باستخدام العرض التجريبي (المبدئي) , يعرض معد المنظومة على الزبون منظومة ذات وظائف محددة في مرحلة مبكرة وإن كان ذلك في مستوى منخفض من الأداء والاعتمادية. الفكرة هنا هي جعل الزبون يقرر شكل المنظومة النهائي في مرحلة مبكرة. يقوم الزبون بمشاهدة شاشات (نماذج) وتقارير وحوارات تفاعلية , والأهم من ذلك بعض الوظائف المطلوبة للنظام. ثم تؤخذ ملاحظاته و اقتراحاته بعناية فائقة. الميزة هنا أن الزبون يعطي فرصة لتغيير متطلباته في مرحلة مبكرة من المشروع حيث تعتبر تكلفة التغيير قليلة نسبياً. تستخدم طريقة العرض التجريبي كحل لأهم مشكلة في النموذج التساقطي (نموذج شلال المياه), ألا وهي مشاكل نقص المتطلبات أو عدم دقتها.

### 6.1 لماذا تحتاج إلى العرض التجريبي؟

1. لتحديد وتنقيح متطلبات المستخدم بوضوح أكثر.
2. لنعطي الزبون صورة واضحة عن شكل المنظومة بعرض عينات منها مثل شاشات وتقارير وحوارات تفاعلية.
3. يعتبر العرض التجريبي عملي أكثر عندما تكون متطلبات الزبون غير واضحة
4. لعرض أو تسويق أو بيع منتج برمجي وتوضح الفوائد التي يمكن جنيها منها.
5. لتوضيح الجدوى الفنية للمنظومة

### 6.2 أنواع العرض التجريبي.

العرض التجريبي الساكن: في هذا النوع من العرض التجريبي يتم عرض الشاشات ونماذج مطبوعة من التقارير على الزبون ولكن دون أن يتمكن من إدخال البيانات أو الحصول على تقارير مباشرة من جهاز الحاسب. بعبارة أخرى فإن نماذج العرض التجريبي هي ببساطة تصاميم وليست منظومات قابلة للتنفيذ باستخدام لغة برمجة مرئية ذات شاشات وتقارير مثل لغة البيسك المرئي. بعد دراسة هذه الشاشات والتقارير , يستطيع الزبون أن يطلب أي تغييرات لهذه الشاشات وتصاميم التقارير.

العرض التجريبي المحذوف: الغرض من العرض التجريبي هو تحديد متطلبات المستخدم في البداية يتم إعداد منظومة ذات وظائف محددة في وقت قصير باستخدام لغات البرمجة (فيجوال بيسك مثلاً) وبدون اهتمام بجودة المنظومة ثم يتم تنقيحها مع الزبون. ولكن يتم التخلص من هذه العينة البرمجية بعد وضوح متطلبات المستخدم , ثم يتم الانتقال إلى المرحلة التالية (مرحلة التصميم) من النموذج التساقطي (التتابعي).

نموذج العرض التجريبي التطوري: الغرض من هذا النموذج هو الوصول إلى المنظومة النهائية ابتداء من نسخة أولية بوظائف محدودة ثم إعداد عدة إصدارات متعاقبة بوظائف أكثر حتى يتم تلبية كل المتطلبات ويتم إعداد المنظومة النهائية. وذلك يعني ان هذا النموذج يمكن استخدامه كبديل للنموذج التساقطي.

يمكن تلخيص نشاطات أعداد نموذج العرض التجريبي في الخطوات التالية:

1. يتم التعرف على متطلبات الزبون من قبل معد المنظومة بمساعدة المستخدم و الزبون.
2. يتم تصميم وتنفيذ منظومة العرض التجريبي للزبون لتقوم بأداء الوظائف المهمة (وظائف محددة). ومن المناسب هنا استعمال لغة برمجة مرئية مثل فيجوال بيسك لإعداد هذه المنظومة.
3. يتم تجربة طريقة عمل منظومة العرض التجريبي من قبل المستخدم.
4. يعطى الزبون فرصة استعمال منظومة العرض التجريبي ونتوقع منه في المقابل أن يعطي ملاحظات او يقترح تغييرات.
5. من التغييرات المطلوبة في الخطوة 4 يمكن لمعد المنظومة البرمجة أن يجهز النسخة اللاحقة.
6. يتم تكرار الخطوات 3 و 4 و 5 حتى يتم تحقيق متطلبات المستخدم تماماً و نحصل على المنظومة النهائية.

### 6.3 مزايا العرض التجريبي.

1. يقلل من مشاكل تحديد المتطلبات.
2. يوفر مشاركة ودعم أكبر من طرف الزبون في المراحل المبكرة من المشروع (التحليل و التصميم).
3. يقلل من المخاطرة في تأخير تسليم المشروع.
4. يمكن استعماله كقاعدة لكتابة المواصفات بجودة عالية.
5. يمكن استعماله لتدريب المستخدمين قبل تسليم المنظومة.

6. يقلل من وقت اعداد المنظومات.
7. يقلل من التكلفة الإجمالية لإعداد المنظومات
8. يقلل من احتمال فشل المشروع.
9. طريقة أفضل للتواصل بين المستخدم و معد المنظومة.

#### 6.4 عيوب العرض التجريبي.

1. أحياناً يحدث التباس للزبون بأن العرض التجريبي هو المنتج النهائي. وفي هذه الحالة يكون من الصعب إقناع الزبون أن منظومة العرض التجريبي هي للعرض فقط (منظومة تجريبية) , وقد يطلب بالإسراع في استلام المنظومة بعد اجراء تعديلات بسيطة عليها.
2. للحصول على منظومة العرض التجريبي بسرعة , يستخدم معد المنظومة عادة أدوات وخوارزميات بسيطة وقليلة الكفاءة وهذه الأدوات والخوارزميات عادة ما تصبح هو الأدوات الأساسية في إعداد المنتج النهائي مما يؤدي إلى إعداد منظومة قليلة الجودة.
3. تخلق إحساساً خادعاً من التفاؤل عن المنظومة.
4. لا ينتج عنها توثيق وهذا يجعل من صيانة المنظومة أمراً صعباً.