

## ● أهم المشاكل التي تواجه محلل النظم :

نتيجة طبيعة عمل محلل النظم وارتباطه بنظام له عدة مكونات وخصائص، وله مشاكل معقدة، ومتشابهة، وأفراد مختلفين في الطباع والتفكير والتعامل، وتقنيات كثيرة ومتجددة، فإنه يوجد عدة مشاكل من أهمها:

- 1- الغموض في ظواهر وأسباب بعض المشاكل .
- 2- الإلتباس في معرفة السبب الحقيقي وراء مشكلة معينة أو قصور معين .
- 3- جمع بيانات غير صحيحة نتيجة الاعتماد على مصدر واحد أو عدم التفريق بين الحقيقة والرأي .
- 4- وضع توقعات غير صحيحة نتيجة السرعة والاستعجال في تحليل البيانات .
- 5- عدم تعاون المستخدم مع محلل النظم .
- 6- تعدد الحلول وصعوبة تحديد الأفضل .
- 7- التدقيق المبالغ فيه يزيد من كلفة التحليل من جانب الجهد والمال والوقت .
- 8- صعوبة اختيار أدوات وأساليب ومنهجيات التحليل لوجود عدة خيارات .
- 9- ضرورة متابعة التطورات في مجال تقنية المعلومات .
- 10- تغير بيئة الأعمال بشكل دائم .
- 11-

## ● تطور علاقة محلل النظم مع المستخدم النهائي :

منذ أن بدأت مهنة محلل النظم وهي لا تستغني عن دور المستخدم في تقديم المعلومات عن النظام القائم للمؤسسة ، ولكن هذا الدور لم يكن واضحاً في بداية ظهور تحليل وتصميم النظم الحاسوبية .  
الطريقة القديمة:

فكانت الطريقة القديمة لهذه العلاقة تعطي دوراً محدوداً للمستخدم نتيجة للفجوة الكبيرة بين محلل النظم الخبير في التقنيات الحديثة والمستخدم العادي الذي لا يعرف بهذه التقنيات شيء .  
الطريقة الحديثة:

المستخدم دور المشترك الأساسي في كافة مراحل التحليل والتصميم وهذا سبب تقلص الفجوة التقنية بين محلل النظم والمستخدم، واستيعاب محلل النظم لأهمية المستخدم كمصدر رئيسي للمعلومات.

## ● مهام محلل النظم الذي يعمل في شركة متخصصة في تحليل النظم:

1. القيام بالدراسات التمهيدية الخاصة بالنظام المطلوب تطويره .
2. تقديم الجدوى الاقتصادية والفنية وتحديد متطلبات النظم الجديد من أجهزة ومعدات وبرمجيات لأخذ الموافقة على بدء الدراسة التفصيلية.
3. القيام بالدراسة التفصيلية من جمع معلومات ومن ثم تحليلها وتحديد متطلبات تطور النظام القائم.
4. وضع التصاميم للنظام الجديد .
5. توثيق النظام الجديد.
6. إعداد أدلة النظام وتدريب الموظفين.
7. صيانة النظام في حالة الاتفاق على ذلك .
8. تقديم حلول تقنية للمشاكل الصغيرة.

## ● فريق عمل تحليل النظم:

### ● محلل نظم واحد:

- 1- لكافة مراحل تحليل وتصميم النظام وذلك في حالة النظم الصغيرة والسهلة.
  - 2- يكون مسئولاً عن مرحلة تحليل النظم ومن ثم يسلم نتائج عمله لمصمم النظم
- فريق من محلي ومصممي النظم:
- 1- لكافة المراحل وذلك في حالة النظم الكبيرة والمعقدة، وفي هذه الحالة يجب أن يكون للفريق قائد هو كبير محلي النظم وأكثرهم خبرة.
  - 2- في حالة النظم الكبيرة جداً، يمكن أن يكون هناك فريقان، فريق من محلي النظم وفريق من مصممي النظم.

## ● أسباب فشل تطوير نظم المعلومات الحاسوبية في بعض المؤسسات :

1. محاولة المؤسسة إلغاء أو تقليص مرحلة التحليل ، ونقل العمل اليدوي بمشاكله وعيوبه إلى نظام حاسوبي .
2. عدم إقتناع المستخدمين بالنظام الجديد أو تخوفهم منه مما يؤدي إلى عدم تعاونهم مع محلل النظم .
3. الروتين والبطء في إجراءات التطوير وأخذ وقتاً طويلاً فقد يؤدي لتغير متطلبات المستخدمين أو تغير محلل النظم .
4. إهمال جانب النوعية لمفهوم نظم المعلومات الحاسوبية .
5. عدم استيعاب إمكانيات النظام الحاسوبي وافترض أن النظام الجديد يجب أن ينفذ الأوامر مهما كانت.

6. تتجه بعض المؤسسات النظام الحاسوبي وافترض أن النظام الجديد يجب أن ينفذ الأوامر مهما كانت.
7. عدم تخصيص إدارة خاصة بنظم المعلومات وإدراج النظام الحاسوبي تحت إدارة غير متخصصة.
8. وجود فجوة تكنولوجية بين العاملين في عمل التطوير وبين موظفي المؤسسة.
9. أنصاف محلل النظم بالتعالي مما يخلق علاقة غير جيدة مع المستخدمين.
10. عدم قدرة محلل النظم إقناع الإدارة ( المستخدمين ) بمميزات نظام المعلومات الحاسوبي.
11. تخويف محلل النظم المستخدمين التعامل مع النظام الجديد.
12. تلميح محلل النظم بضرورة إحلال موظفين جدد مختصين محل الموظفين الأصليين.

### ● أساليب ومنهجيات تطوير نظم المعلومات الحاسوبية Systems Information Development Approaches and Methodologies

#### أساليب تطوير نظم المعلومات الحاسوبية

هي المراحل التي يمر بها تطوير النظام من بداية مرحلة تحليل نشاط المؤسسة وحجم عملها حتى تسليم النظام، وكل أسلوب له مراحله المحددة، ويمكن اختيار أحد الأساليب لتطوير نظم المعلومات بما يناسب طبيعة وإمكانية النظام في المؤسسة

#### ● أساليب تطوير نظم المعلومات

- أ- دورة حياة تطوير النظم System Development life Cycle
- ب- التطوير التدريجي - على مراحل Staged Development
- ت- التطوير المعتمد على فريق العمل Team Centered Development
- ث- النمذجة الأولية Prototyping
- ج- أسلوب التطوير العاجل Ad-hoc Approach
- ح- أسلوب التحليل من أعلى الى أسفل Top-Down Approach
- خ- أسلوب التحليل من أسفل الى أعلى Down-Top Approach
- د- أسلوب التركيب Synthesis Approach

## أ- أسلوب دورة حياة تطوير النظم System Development life Cycle (SDLS)

### خصائص الاسلوب:

1. يعتبر الأكثر استخداماً في النظم الصغيرة والكبيرة نسبياً لما له من خصائص أهمها:
2. تتميز الخطوات المحددة في هذا الاسلوب بالوضوح وسهولة الفهم.
3. يعتبر مناسباً للنظم التي يمكن تحديدها بدقة (مثل النظم الادارية)
4. يعمل على تطبيق اجراءات التأكد من الجودة عن طريق المراجعة بعد كل خطوة.
5. يعمل على تدقيق المخرجات للتأكد من مطابقتها للمتطلبات.
6. يعمل على اختبار النظام.
7. لا يعتبر مناسباً للمنظمة الكبيرة جداً أو التي لا تتصف بالوضوح.

### مراحل الاسلوب:

- 1- تحديد المشكلة Problem Recognition
- 2- دراسة الجدوى Feasibility study
- 3- تحليل النظام Analysis
- 4- تصميم النظام Design
- 5- تنفيذ النظام Implementation (ويشمل التشييد Construction)
- 6- الصيانة Maintenance

### 1- مرحلة تحديد المشكلة Problem Recognition

- يتم خلالها فهم النظام القائم من خلال جمع المعلومات وتحديد أنه توجد مشكلة.
- مخرجات هذه المرحلة تحديد المتطلبات التي يتوقعها المستخدم من النظام الجديد .

### 2- دراسة الجدوى Feasibility study:

- يتم اقتراح حلول للمشكلة ، قد تكون تحسين بعض الوظائف أو تطوير النظام القائم ، أو اقتراح نظام جديد .
- المخرجات:المقترح الأنسب لحل المشكلة من حيث الفائدة والكلفة.

### 3- تحليل النظام Analysis :

يتم في هذه المرحلة:

- دراسة النظام القائم بالتفصيل من حيث:
  - \* تحديد الوظائف التي يقوم بها النظام وكيفية أدائها .
  - \* تحديد البيانات المطلوبة لانجاز الوظائف
  - \* تحديد المتطلبات
- الأدوات الأساسية: أساليب جمع الحقائق, قاموس البيانات, رسومات تدفق البيانات, مواصفات العمليات , جداول القرارات, خرائط سير النظام....

المخرجات: نموذج تفصيلي للنظام القائم ونموذج للنظام المقترح

في هذه المرحلة يتم الاجابة على السؤال : ماذا يتوجب علينا فعله لحل المشكلة .

What the system should do?

### 4- تصميم النظام Design:

- \* تصميم قاعدة البيانات .
  - \* تصميم البرامج .
  - \* تصميم المدخلات .
  - \* تصميم المخرجات.
  - \* تصميم واجهات الاستخدام (الشاشات) .
  - \* تصميم أدلة الاستخدام .
  - المخرجات: مواصفات النظام الجديد .
- في هذه المرحلة يتم الإجابة على السؤال : كيف يجب أن تحل المشكلة ؟ (How?).

### 5- تنفيذ النظام Implementation :

يتم في هذه المرحلة:

- \* بناء مكونات النظام.(اعداد البرامج)
  - \* اختبار النظام.
  - \* التحويل الى النظام الجديد .
  - \* تدريب المستخدمين على استخدام النظام الجديد .
- المخرجات: نظام جديد

## 6- الصيانة Maintenance :

ويتم فيها:

- \* مراقبة المخرجات من العمل الفعلي.
  - \* اكتشاف الأخطاء وإجراء التحسينات اللازمة
  - \* (اصلاح, وتكبير النظام عند الحاجة لذلك) إجراء التعديلات على النظم بناءً على طلب المستخدمين.
- المخرجات: بداية بناء نظام جديد :
- Begin Building a new System

## ب- أسلوب التطوير التدريجي-على مراحل staged Development

مناسب للنظم الكبيرة جداً التي يمكن تقسيمها الى نظم فرعية .

### مراحل الاسلوب:

1. دراسة جدوى النظام الكلي.
2. تقسيم النظام الى عدة نظم فرعية.
3. استخدام أسلوب " استخدام اسلوب دورة حياة تطوير النظم " في تطوير كل نظام فرعي.
4. ربط النظم الفرعية مع بعضها البعض.

## ت- أسلوب التطوير المعتمد على فريق العمل Team Centered Development

مناسب عندما يكون فريق محلي النظم قوياً ووثقاً من دقة عمله وله خبرة سابقة كبيرة .

### مراحل الاسلوب:

1. يعتمد على اعطاء فريق العمل الحرية في تطوير نظام المعلومات
2. يمكن أن يتبع " استخدام اسلوب دورة حياة تطوير النظم " مع عدم التقيد بالمراحل بدقة.(يمكن البدء بمرحلة تالية مع الاستمرار في إكمال المرحلة السابقة أوالعودة لاكمال معلومات معينة) .
3. يجب وضع حد زمني حتى لا تطول مرحلة تطوير النظام.

## ث- أسلوب النمذجة الأولية Prototyping

تستخدم النمذجة الأولية في تكوين صورة أولية عن النظام النهائي.

### طرق النمذجة الأولية:

1- النمذجة الأولية الارتقائية: Evolutionary Prototyping

2- النمذجة الأولية الشبيهة: Throw-away prototyping

### النمذجة الأولية الارتقائية: Evolutionary Prototyping

يتم تطوير نموذج أولي بسيط يوضع قيد الاستخدام والاختبار، ثم يتم ترقيته بشكل متدرج من خلال

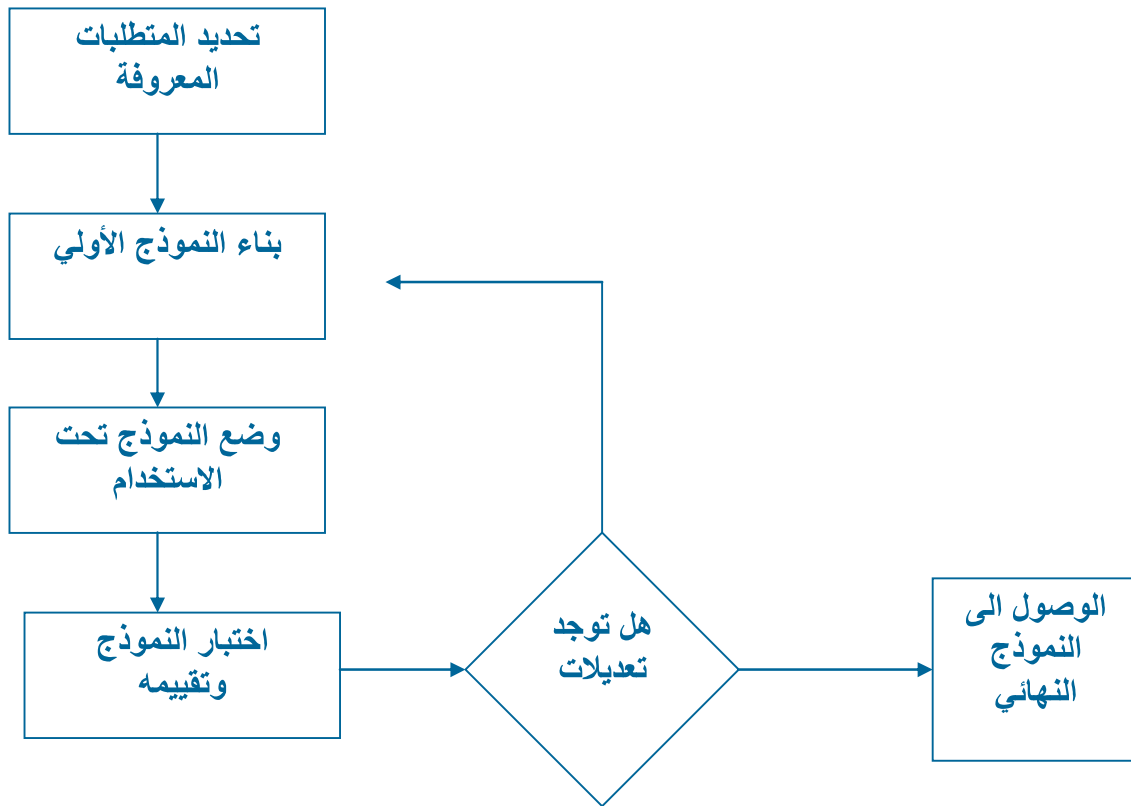
ملاحظات الاستخدام مع الاختبار بعد كل خطوة. ومع تكرار الاستخدام والاختبار

يتم اكتساب المعرفة لمتطلبات الخطوة التالية ويواصل العمل حتى الوصول الى النظام النهائي.

### حالات الاستخدام:

- عندما يكون النظام غير واضح.
- عندما يكون النظام جديد كلياً وغير مستخدم من قبل.
- عندما يتضمن النظام حوارات كثيرة بين المستخدم والحاسوب لا تكون واضحة من البداية.
- نظم دعم القرارات التي تتطلب حل مشاكل غير محددة حلولها بشكل دقيق.

### النمذجة الأولية الارتقائية: Evolutionary Prototyping



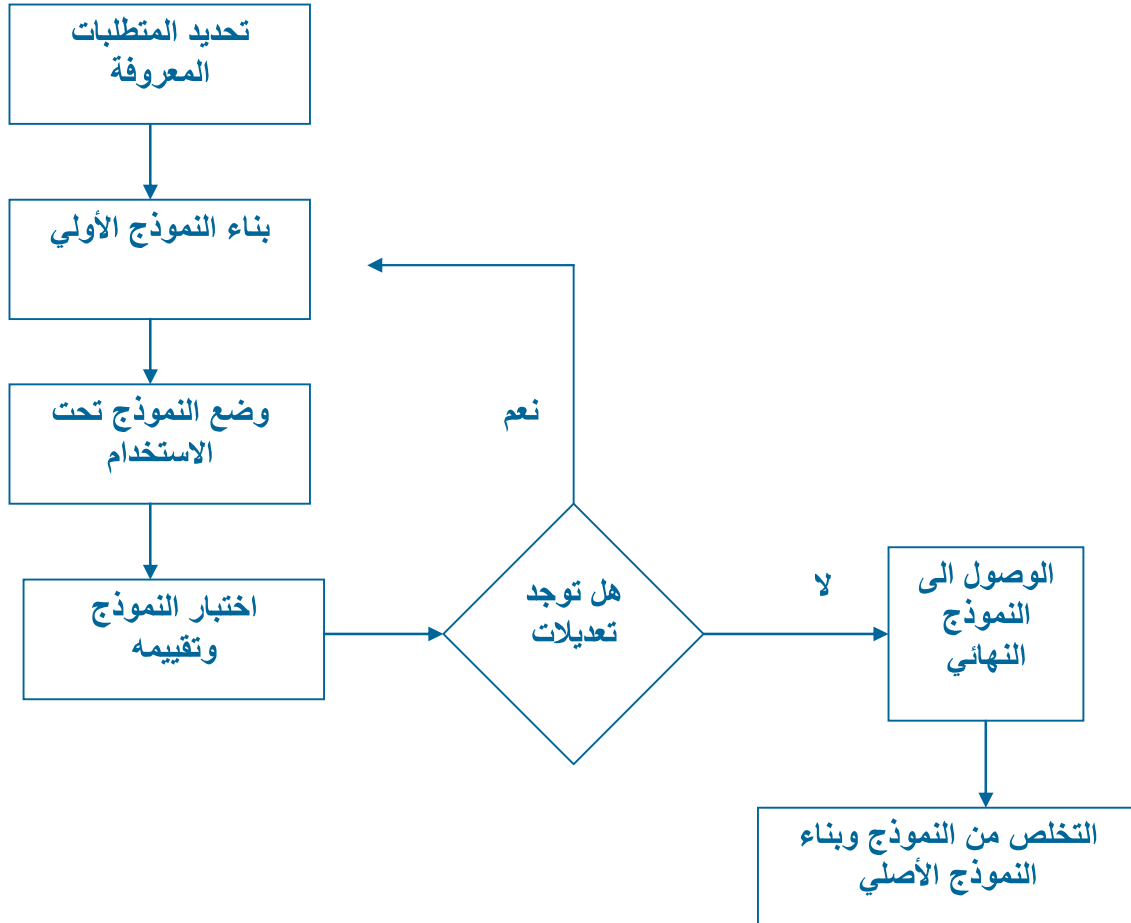
## النمذجة الأولية الشبيهة (المنبوذة): Throw-away prototyping

يتم بناء نموذج أولي شبيه بالنظام المطلوب بطريقة سريعة، وتستخدم في بنائه أدوات سهلة. وعند التأكد من تلبية الاحتياجات المحددة، يتم بناء النظام الأصلي بعد فهم المتطلبات والاحتياجات بشكل صحيح ودقيق من خلال النموذج الشبيه، وهنا يتم استخدام الأدوات والتقنيات القوية والمناسبة ويتم التخلص من النموذج الأولي.

**حالات الاستخدام:**

- عندما يكون النظام غير واضح.
- عندما يكون النظام جديد كلياً وغير مستخدم من قبل.
- عندما يكون تطوير النظام مكلفاً جداً، ويفضل التأكد من عمل النظام.
- عندما تكون مخاطر الانتقال الى النظام الجديد كبيرة.
- عندما يكون هناك شك من امكانية تنفيذ النظام عملياً وعلى أرض الواقع.

## النمذجة الأولية الشبيهة: Throw-away prototyping



### مميزات النمذجة الأولية:

1. يمكن اعتبارها اقتصادية حيث تقل الكلفة بسبب تلافي الأخطاء والالتباس وسوء فهم الاحتياجات في مرحلة مبكرة.



2. سهولة الفهم حيث تتجه الى بناء النظام خطوة بخطوة, ولا يتم الانتقال الى خطوة تالية إلا بعد التأكد من الخطوة الأولى.

3. يستوعب المستخدم النظام جيداً قبل تركيبه وتنفيذه بشكل نهائي.

### عيوب النمذجة الأولية:

1. في النمذجة الشبيهة, قد لا يكون النظام الأصلي مطابقاً تماماً للنظام المطلوب.
2. اذا لم يتم التحكم بالمراحل يمكن أن تطول عملية النمذجة.

### ج - اسلوب التطوير العاجل Ad-hoc Approach :

- مناسب عندما يكون لدى المؤسسة مشكلة محددة أو طلب محدد.
- يركز محلل النظم على تحديد المشكلة وأهم النقاط المتعلقة بها وتقديم الحلول المناسبة لها دون اعتبار للاحتياجات الأخرى خارج حدود المشكلة قيد التحليل.

### ح- أسلوب التحليل من أعلى الى أسفل Top-Down Approach :

- يهدف الى تحقيق الأهداف الاستراتيجية للمؤسسة بالدرجة الأولى.
- يبدأ بتحديد احتياجات النظام ككل, ثم النظم الفرعية والأنشطة الفرعية.
- يسمى احياناً الاسلوب التنازلي.

### خ- أسلوب التحليل من أسفل الى أعلى Down-Top Approach :

- يبدأ بتحليل أنشطة المؤسسة ونظمها الفرعية ويتجه الى أعلى ويحدد الاحتياجات الاستراتيجية.
- يسمى احياناً الاسلوب التصاعدي.

### د- أسلوب التركيب Synthesis Approach :

- يتم بناء نظام المعلومات من وحدات وظيفية موجودة .
- يناسب النظم المعروفة والتي يمكن تركيب نظامها من وحدات برمجية جاهزة يتم اعادة استخدامها (Reusable)

### عوامل اختيار الاسلوب الأنسب

- درجة وضوح وتحديد بنية النظام.
- خبرة فريق العمل.
- حجم النظام.
- نوع الاحتياجات المطلوبة (عامة شاملة أم مشاكل محددة)
- أولوية الاحتياجات (استراتيجية أم فرعي)

## ● منهجيات تطوير نظم المعلومات System Development Methodologies

هي مجموعة الطرق والأدوات المستخدمة في تطوير النظام.

- تقنيات معتمدة على رسم نماذج (Models) تمثل النظام الحالي أو الرؤية المستقبلية للنظام الجديد وتمثل :

- منهجية التحليل والتصميم الهيكلي

(Structured System Analysis and Design Methodology)

- منهجية هندسة المعلومات (Information Engineering Methodology)

- منهجية الحزم الجاهزة (Commercial off-the-shelf software COTS)

### منهجية التحليل والتصميم الهيكلي

تم تطوير هذه المنهجية وأصبحت منهجية قياسية، تعتمد على نماذج العمليات (Process Models) أي التوصيف الهيكلي للعمليات باستخدام الأدوات الهيكلية مثل مخططات تدفق البيانات (DFD)، ومن أهم خصائصها:

- أصبحت منهجية قياسية
- تعتمد على الجمع ما بين عملية تحليل البيانات وتحليل التدفقات في آن واحد
- تحدد بدقة وبالتفصيل الكامل جميع المهام المطلوبة والتسلسل اللازم.

### منهجية هندسة المعلومات

تعتبر هذه المنهجية معتمدة على البيانات من التخطيط الاستراتيجي للمؤسسة ككل، وبناء نموذج البيانات (Data Model) الذي يأخذ بعين الاعتبار الاحتياجات المستقبلية، ومن أهم خصائصها:

- تعتمد على التخطيط الاستراتيجي للمؤسسة
- بناء نموذج البيانات (Data Model)
- تتطلب من محلي النظم دراسة مجال العمل الذي تمارسه المؤسسة واستكشاف التطورات المحتملة.

### التحليل الهيكلي و هندسة المعلومات

- في التحليل الهيكلي يتم رسم نموذج العمليات أولاً (Process Model)، بينما في هندسة المعلومات يتم رسم نموذج البيانات (Data Model) أولاً.
- النموذجين مطلوبين في المنهجتين، لأن نظام المعلومات يتضمن العمليات والبيانات معاً.