

**AL- Mustansirya University**  
**College of Education**  
**Department of Computer Science**



**Course: Software Engineering**  
**Lecturer: Iman Hussein**  
**Third Class**

## **CHAPTER 1**

### ***Introduction to Software Engineering***

Topics:

- 1.1 The Computer Software
- 1.2 Software Engineering
- 1.3 Software Characteristics
- 1.4 Programmer & Software Engineer
- 1.5 The characteristic of software engineer
- 1.6 Software Applications
- 1.7 The Evolving Role of Software
- 1.8 Software: A crisis on the horizon
- 1.9 The Attributes of Good software
- 1.10 The goals of Software Engineering

## 1.1 The Computer Software

It is the product that software engineers design and build. It encompasses programs that execute within a computer of any size and architecture, documents that encompass hard-copy and virtual forms, and data that combine numbers and text but also includes representations of pictorial, video, and audio information.

Software engineers built it, and virtually everyone in the industrialized world uses it either directly or indirectly.

When you built computer software like you built any successful product, by applying a process that leads to a high-quality result that meets the needs of the people who will use the product. You apply a software engineering approach.

### The software might take the following forms:

1. Instructions: Computer programs, that when executed provide desired function and performance.
2. Data structured: That enable the programs to adequately manipulate information.
3. Documents: That describes the operation and use of programs.

يستخدم مصطلح برمجيات (software) عادة للإشارة إلى برامج الحاسوب (computer programs) . وفي الحقيقة ليس هذا إلا جزءاً من التعريف حيث تتكون البرمجيات بصورة عامة من :

- برامج (Programs / Instructions) : تعليمات للحاسوب لتنفيذ عمل محدد.
- التوثيق (Documentation / Documents) : وثائق تصف طريقة إعداد البرنامج واستخدامه ومن أمثلتها كتيب التشغيل (User Manual).
- البيانات (Date) : وتكون إما على صورة [ نص أو صوت أو صورة ثابتة أو متحركة (فيديو) ].
- Data structured : وهي التي تمكن البرامج من معالجة المعلومات .

في البداية كانت كلمة برنامج هي المستخدمة فيما يعرف اليوم بالبرمجيات software (المنظومات) بعد إضافة التوثيق . ولا شك ان نقص التوثيق او عدم وجوده يؤدي الى الارتباك في متابعة المنظومة وتشغيلها من قبل المستخدم بصورة مثلى. لذلك بدأ الاهتمام بالتوثيق ولكن لم يخل الامر من بعض المشاكل حتى بعد إضافة التوثيق فظهرت مشاكل متعددة بسبب تنوع وتعقد المنظومات المراد إعدادها .

هذه المشاكل عرفت باسم أزمة البرمجيات وهي تعني المشاكل التي تمت مواجهتها أثناء إعداد المنظومات البرمجية نظراً لزيادة احتياجات ومتطلبات الزبون وازدياد حجم البرمجيات . لذلك التجأ المختصون للبحث عن حل لهذه الأزمة وكان الحل هو اتباع الأسلوب الهندسي في إعداد المنتج على شكل مشروع مما يستوجب تعيين مدير مشروع وفريق عمل . والنقطة الأخرى في الأسلوب الهندسي هي ان المنتج يمر عبر مراحل هي : التحليل ، والتصميم ، والتنفيذ ، والاختبار ، والصيانة . والنقطة الثالثة التي استدعت اللجوء الى الأسلوب الهندسي هي الاهتمام بجودة المنتج ولا شك ان موضوع الجودة يعتبر من اولويات الاهتمامات .

## **1.2 Software Engineering**

- it is systematic approach to the development, operation, maintenance and retirement of software. It is the application of computer science along with mathematics and ergative science. In the current scenario the S.E has a specific importance for making particular software.

### **تعريف هندسة البرمجيات (Software Engineering)**

**نعطى هنا ثلاثة تعريفات متكافئة :**

- 1- **هندسة البرمجيات :** هي اسلوب منظم لانتاج وتشغيل وصيانة البرمجيات .
- 2- **هندسة البرمجيات :** هي المنهاج الاداري والتقني الذي يؤدي الى انتاج وصيانة البرمجيات ذات الجودة في حدود الوقت والتكلفة التقديرية .
- 3- **هندسة البرمجيات:** هي دراسة واستعمال طرق وادوات وأسس وفنيات علم الحاسوب والادارة والهندسة والاقتصاد وعلم النفس ومهارة التواصل في اعداد وصيانة برمجيات ذات موثوقية (اعتمادية) عالية وقابلة للصيانة والاستعمال في حدود الوقت والتكلفة المتوقعة . اي انها عبارة عن تقنية تمزج بين ادوات وطرق وعمليات لغرض الحصول على منتج برمجي ناجح ،وقد حلت هذه الهندسة محل ما كان يعرف سابقا بمعالجة البيانات نظرا لاشتمالها على الجوانب الادارية والنفسية والاقتصادية المتعلقة بمشاريع البرمجيات .

## 1.3 Software Characteristics

- 1) Software are developed or engineered
- 2) Most of software is custom build rather than assemble from existing component.
- 3) Computer program and associated documentation.
- 4) Easy to modified.
- 5) Easy to reproduce.
- 6) Software product may be developed for a particular customer or for the general market.

### (Software Characteristics) خصائص البرمجيات

- 1- ان البرمجيات يتم تصميمها او تطويرها .
- 2- معظم البرمجيات يتم بناءها حسب رغبة الزبون بدلا من تجميعها من مكونات موجودة سابقا
- 3- البرمجيات فيها برامج الكمبيوتر ترافقها الوثائق المرتبطة بها
- 4- سهولة التعديل او التحديث
- 5- سهولة اعادة انتاجها
- 6- البرمجيات المنتجة ممكن ان توضع لعميل معين (زبون) أو للسوق العام.

## 1.4 Programmer & Software Engineer

### الفرق بين المبرمج ومهندس البرمجيات ؟

الاجابة على هذا السؤال غير واضحة للعديد من الناس ، والتعريفات التالية يجب ان تزيل هذا الغموض

المبرمج (Programmer) : هو الشخص الذي يهتم بتفاصيل تنفيذ وتعديل الخوارزميات وتراكيب البيانات باستخدام لغة برمجة معينة .

مهندس البرمجيات (Software Engineer) : هو الشخص الذي يهتم بأمر التحليل والتصميم والتنفيذ والاختبار والصيانة والتوثيق والادارة للمشاريع البرمجية.

من هذه المقارنة نرى ان مهندس البرمجيات له مهام اكثر من المبرمج ، فعليه ان ينظر الى مسألة اعداد البرمجيات من زاوية أوسع . كما عليه ان يبحث في الجوانب التالية من المشروع :

- 1- التحليل (Analysis)
  - 2- التصميم (Design)
  - 3- التشفير او البرمجة (Coding)
  - 4- اكتشاف الأخطاء وتصحيحها (Debugging & Testing)
  - 5- صيانة البرمجيات (SW Maintenance)
  - 6- التوثيق (Documentation)
  - 7- ادارة المشاريع البرمجية (SW Project Management)
- ومع الجدير بالملاحظة يجب ان يكون لمهندس البرمجيات خبرة ومهارة واسعة كمبرمج .

## تعريف اخر لمهندس البرمجيات :

مهندس البرمجيات **software engineer** هو الشخص الذي تقع على عاتقه تطوير المنتجات البرمجية التي تباع للزبائن أو التي يحتاج إليها العملاء والزبائن. ويجب عليه إن يتبنى أسلوباً منظماً ونظامياً في عمله لكي يحقق الأهداف المرجوة من استخدام علم هندسة البرمجيات، كما يتحتم عليه استخدام الأدوات المناسبة (كاختيار لغة برمجة مناسبة من لغات البرمجة عالية المستوى) والتقنيات الضرورية (كالخوارزميات مختلفة الأعراض)، وذلك كله اعتماداً على نوع المشكلة التي يقوم بحلها وقيود التطوير المفروضة والموارد المتاحة ( المادية منها والبشرية ). يقوم مهندسو البرمجيات بجعل الأشياء تعمل، فهم يطبقون النظريات والطرق والأدوات حيثما تكون هذه الأدوات مطلوبة، لكنهم يستخدمون تلك الأدوات بانتقائية ويحاولون اكتشاف الحلول حتى ولو لم تكن هناك نظريات تطبيقية أو طرق دعم. ويدرك المهندسون أيضاً انه يجب عليهم العمل من خلال القيود التنظيمية والمالية، لذلك يبحثون عن الحلول ضمن هذه القيود. ولا يهتم مهندس البرمجيات فقط بالعمليات التقنية لتطوير البرمجيات، ولكنه أيضاً يهتم بكل الأنشطة مثل إدارة مشروع البرمجيات وتطوير الأدوات والطرق والنظريات التي تدعم إنتاج البرمجيات.

## 1.5 The Characteristic of Software Engineer

- 1- Good programmer and fluent in one or more programming language.
- 2- Well versed data structure and approaches.
- 3- Familiar with several designs approaches.
- 4- Be able to translate vague (not clear) requirements and desires into precise specification.
- 5- Be able to converse with the user of the system in terms of application not in computer”.
- 6- Able to a build a model. The model is used to answer questions about the system behavior and its performance.
- 7- Communication skills and interpersonal skills.

## خصائص مهندس البرمجيات (Characteristic of Software Engineer):

- 1- مبرمج جيد ويجيد واحدة أو أكثر من اللغات البرمجية
- 2- يمتلك دراية جيدة في هياكل البيانات ومناهجها .
- 3- لديه دراية بطرق التصميم المتعددة
- 4- أن يكون قادراً على ترجمة [المتطلبات الغامضة (الغير واضحة) والرغبات] الى مواصفات دقيقة.
- 5- أن يكون قادراً على التحدث مع المستخدم للنظام من حيث التطبيق وليس في "جهاز الكمبيوتر".
- 6- قدرة على بناء نموذج. يتم استخدام النموذج للإجابة على الأسئلة حول سلوك النظام وأدائه.
- 7- يمتلك مهارات الاتصال ومهارات التعامل مع الآخرين.

## **1.6 Software Applications**

The following software areas indicate the breadth of potential applications:

**1. Engineering and Scientific Software:** modern applications within the engineering/scientific area are moving away from conventional numerical algorithms. Computer-aided design, system simulation, and other interactive applications have begun to take on real-time and even system software characteristics.

**2. Business Software:** Business information processing is the largest single software application area. Discrete "systems" (e.g., **payroll, accounts receivable/payable, inventory**).

**3. Personal Computer Software:** Such as (Word processing, spreadsheets, computer graphics, multimedia, entertainment, database management).

**4. Artificial Intelligence Software:** It makes use of non-numerical algorithms to solve complex problems that are not amenable to computation or straightforward analysis. Expert systems, also called knowledge-based systems, pattern recognition (image and voice), artificial neural networks, theorem proving, and game playing are representative of applications within this category.

**5. Embedded Software:** Intelligent products have become commonplace in nearly every consumer and industrial market (e.g., keypad control for a microwave oven or digital functions in an automobile such as fuel control, and braking systems).

**6. System Software:** It is a collection of programs written to service other programs. Some system software (e.g. **Compilers, editors, and file management utilities**) process complex, but determinate, information structures. Other systems applications (e.g., **operating system components, drivers, telecommunications processors**) process largely indeterminate data.

**7. Real-time Software:** Software that monitors/analyzes/controls real world events as they occur is called real time. Real-time differs from "interactive" or "time sharing". A real-time system must respond within strict time constraints. The response time of an interactive (or time sharing) system can normally be exceeded without results.

**8. Web-based Software:** The Web pages retrieved by a browser are software that incorporates executable instructions (e.g., HTML, Perl, or Java), and data (e.g., hypertext and a variety of visual and audio formats).

## تطبيقات البرمجيات او انواع البرمجيات ( Software Applications / Types of Software ) :

تتنوع البرمجيات وتطبيقاتها على النحو التالي :

1- برمجيات علمية وهندسية ( Scientific & Engineering Software ) :  
يشار الى البرمجيات ذات الطابع الحسابي (العددي) احيانا بالبرمجيات العلمية . ويجد هذا النوع من البرمجيات تطبيقا واسعا في مجال العلوم التطبيقية مثل :

- علم الفلك (Astronomy)
- وتحليل الجهد (Stress Analysis)
- والتصميم باستعمال الحاسوب (Computer Aided Design)
- والمحاكاة (simulation)
- والتصنيع الالي (Computer- Aided Manufacturing)
- والاحصاء (Statistics)

2- برمجيات تجارية (Business Software)  
وهي برمجيات تتعامل مع البيانات للحصول على معلومات ضرورية للعمليات التجارية وعمليات اتخاذ القرار .  
وهي ذات تطبيقات متعددة مثل :

- منظومة المرتبات (جدول الرواتب) Payroll System
- منظومة مبيعات (accounts receivable/payable) / Sales System
- منظومة مخازن Inventory System

3- برمجيات الحاسوب الشخصي (Personal Computer Software)  
وهي برمجيات مصممة لتطبيقات الحاسوب الشخصي (PC) لغرض تحسين انتاجية الموظفين والمستخدمين على حد سواء . ومن امثلة تطبيقاتها :

- معالجة الكلمات Word Processing
- الجداول الالكترونية Spreadsheets
- رسومات الحاسوب Computer Graphics
- ادارة قواعد البيانات DBMS
- الوسائط المتعددة Multimedia
- ادارة الاعمال Project Management
- الترفيه Entertainment

4- برمجيات الذكاء الاصطناعي [ Artificial Intelligence Software (AI) ]  
وهي برمجيات تجعل الحاسوب يستفيد من خبرته في اتخاذ القرار بصورة مشابهة لسلوك الانسان في التعلم والاسنتاج . ومن امثلة تطبيقاتها :

- الانظمة الخبيرة ( Expert Systems/ knowledge-based systems ) مثل منظومة التشخيص الطبي
- التعرف على الصورة والصوت pattern recognition(Image and Voice Recognition)
- برهنة النظريات (Theorem Proving)
- الالعاب (Games)
- الانسان الآلي ( Robots )
- الشبكات العصبية الاصطناعية (Artificial Neural Networks)

#### 5- البرمجيات المقيمة/ البرمجيات المدمجة ( embedded software )

- برمجيات تقطن في ذاكرة روم (ROM) للتحكم في الآلات الأخرى مثل الأجهزة المنزلية. ومن تطبيقاتها :
- الفرن (الميكروويفي) Microwave Oven
  - الوظائف الرقمية في السيارات (digital functions in an automobile) مثل
    - التحكم في الوقود (Fuel Control)
    - نظام الكابح للسيارات (Braking Systems)

#### 6- برمجيات النظم (System Software)

وهي مجموعة برامج أعدت لخدمة برامج أخرى. ولها تطبيقات في المجالات التالية :

- المترجمات Compilers
- المحررات Editors
- خدمة ادارة الملفات File Management Utilities
- نظام التشغيل Operating System

#### 7- برمجيات الوقت الحقيقي (Real-time Software)

وهي برمجيات تتحكم في الأحداث اثناء وقوعها بحيث يتراوح وقت الاستجابة من 1 ميلي ثانية (او اقل) الى دقيقة واحدة . تتكون برمجيات الوقت الحقيقي من 4 اجزاء :

- 1- جزء لتجميع البيانات Data gathering component
- 2- جزء التحليل Analysis component
- 3- جزء الاخراج Output component
- 4- جزء تحكم ينسق العمل بين جميع الاجزاء الأخرى لتحقيق الاستجابة الفورية

ومن التطبيقات لبرمجيات الوقت الحقيقي هي :

- منظومة التحكم والمراقبة في حركة الطيران (Air traffic control)
- منظومة حجز رحلات الخطوط الجوية (Airline reservation)

#### 8- برمجيات الانترنت (Internet Software/ Web-based Software)

وهي برمجيات تتعلق باستخدام الانترنت ومن امثلة هذه التطبيقات :

- المصاريف بالانترنت E-Banking
- الادارة الالكترونية E-Government
- التعليم بالانترنت E-Learning
- التجارة الالكترونية E-Trade



## 1.7 The Evolving Role of Software / History of Software

Today, software takes on a dual role. It is a product and, at the same time, the vehicle for delivering a product.

1- As a product: it delivers the computing potential embodied by computer hardware or, more broadly, a network of computers that are accessible by local hardware. Whether it resides within a cellular phone or operates inside a mainframe computer, software is information transformer (producing, managing, acquiring, modifying, displaying, or transmitting) information that can be as simple as a single bit or as complex as a multimedia presentation.

2- As the vehicle used to deliver the product: software acts as the basis for the :

- a. control of the computer (operating systems).
- b. The communication of information (networks).
- c. The creation and control of other programs (software tools and environments).

The role of computer software has undergone significant change over a time span of little more than 50 years. Dramatic improvements in hardware performance, profound changes in computing architectures, vast increases in memory and storage capacity, and a wide variety of exotic input and output options have all precipitated more sophisticated and complex computer-based systems. Sophistication and complexity can produce dazzling results when a system succeeds, but they can also pose huge problems for those who must build complex systems.

The lone programmer of an earlier era has been replaced by a team of software specialists, each focusing on one part of the technology required to deliver a complex application.

### تاريخ البرمجيات History of Software

مرت برمجة الحاسوب بخمسة عصور لتصل الى ما وصلت اليه اليوم . وفيما يلي التقسيم لهذه الحقبات التي تسمى عادة بأجيال الحاسوب :

#### **الجيل الاول : (1946-1965) The First Era**

تميز هذا الجيل بما يعرف بالمعالجة بالدفعه Batch Processing حيث كانت البرامج تجمع في مجموعات (دفعات) لغرض المعالجة الجماعية كما كان التوزيع محدودا Limited Distribution فلم تكن اغلب الشركات تهتم بالحاسوب وتطبيقاته وكان البرنامج الواحد مخصصا لمستخدم واحد وهو ما يعرف بالبرمجيات المخصصة Custom software حيث نجد ان كل برنامج مصمم لزيون معين وكانت معظم البرمجيات ذات طابع علمي وهو ما يعرف بالبرمجيات العلمية .

#### **الجيل الثاني : (1966-1975) The Second Era**

في هذا الجيل اصبح الحاسوب قادرا على ان يخدم اكثر من مستعمل في وقت واحد، وهو ما يعرف عادة بنظام تعدد المستعملين Multi-user system وقد نتج عن هذا النظام اسلوب جديد في البرمجة (البرمجة المتعددة Multiprogramming) حيث اصبح بالامكان تنفيذ عدة برامج في نفس الوقت ،وتطورت مهام انظمة التشغيل لتشمل المعالجة متعددة الوظائف Multitask processing التي تسمح لأكثر من برنامج واحد ان يبقى في الذاكرة في نفس الوقت .كما تميز هذا العصر بالمعالجة التفاعلية Interactive Processing التي مكنت المستعمل من الاتصال مباشرة بالحاسوب وان يتجاوب مباشرة مع النتائج في نفس الجلسة .الى جانب ذلك توفرت المعالجة بالوقت

الحقيقي Real-time processing وهي ما يعرف عادة بالمعالجة المباشرة On Line مع وقت استجابة قصير وترجع سريع .  
وتطورت البرمجيات واصبحت تعتبر منتوجا تجاريا ، وبدا ظهور مؤسسات البرمجة ، ورافق ذلك ظهور الجيل الاول من نظم ادارة قواعد البيانات (Data Base Management Systems (DBMS).

وخلال هذه الفترة ظهرت ازمة البرمجيات Software Crisis وبدا هذا المصطلح في التداول بين المختصين في البرمجيات ليعبر عن المشاكل التي كانت تواجههم اثناء اعداد واستخدام المنتج البرمجي.

### الجيل الثالث: (1976-1985) The Third Era

تميز هذا الجيل بما يلي :

- النظم الموزعة Distributed systems وهو نظام توجد فيه عدة حواسيب تعمل معا في نفس الوقت بطريقة منسقة .
- هبوط في اسعار المعدات Low cost hardware لتصبح رخيصة الثمن في متناول العديد من الشركات والافراد.
- ازمة البرمجيات ازدادت مع تنوع التطبيقات واصبح البحث عن حل لهذه الازمة ضروريا.
- استعمال المعالج الدقيق microprocessor في كثير من التطبيقات مثل :
  - السيارات Automobiles .
  - افران الموجات الدقيقة Microwave ovens.
  - الانسان الآلي Robot .
  - محطات الطاقة Power Plants .
  - التحكم في الحركة الجوية Air traffic Control .
  - استعمال الحواسيب الدقيقة Microcomputers .حيث اصبح الحاسوب في متناول الكثير من عامة الناس .

### الجزء الرابع : (1986-1994) The Fourth Era

تميز هذا الجيل بما يلي :

- انظمة تشغيل الحواسيب الشخصية PC operating systems مثل : DOS و WINDOWA .
- حواسيب صغيرة وقوية Powerful Microcomputers .
- شبكات الحاسوب المحلية والعالمية Local / global networks .

### الجيل الخامس : (1995-؟)

تميز هذا الجيل بما يلي :

- اتساع تطبيقات الشبكات المحلية والواسعة Networks(Wan/LAN) واستعمال الانترنت على نطاق واسع .
- الاهتمام المتزايد بمفهوم الذكاء الاصطناعي Artificial intelligence والنظم الخبيرة (Expert systems) .
- التقنيات الشبئية Object Oriented technology .
- البرمجة الجزئية Component programming .
- اعادة استعمال البرمجيات Software reuse .
- انجاز العديد من التطويرات مثل المعالجة المتوازية parallel processing والاقراص البصرية optical disks والوسائط المتعددة multi-media .

## 1.8 Software: A crisis on the horizon

Whether we call it a software crisis or affliction, the term alludes to a set of problems that are encountered in the development of computer software. The problems are not limited to software that “doesn’t function properly”. Rather, the affliction encompasses problems associated with how we develop software, how we support a growing volume of existing software, and how we can expect to keep pace with a growing demand for more software.

### Elements of the Continuing Software Crisis

- Software is not delivered on time
- Software is over budget (usually by a factor of 2 or more)
- Software is unmaintainable due to:
  - Poor design
  - Poor documentation (most software can be understood only by its author, and then only within a few months of writing it)
  - Software is inefficient (new versions of complex software require machine upgrades)
- Software is unreliable due to:
  - Poor design (Therac-25 disaster)
  - Inadequate testing (market pressures, beta releases)
  - Impossible testing (SDI)

### ماذا يعنى مصطلح أزمة البرمجيات Software Crisis

ان الانتشار الواسع في استعمال الحاسوب جعل المجتمع يعتمد بصورة متزايدة على البرمجيات . ومن امثلة هذه البرمجيات او المنظومات البرمجية التطبيقية ما يلي :

- **Banking system** النظام المصرفية
- **Stock market** سوق البورصة
- **Airline reservation** حجز الرحلات الخطوط الجوية

هذه النظم وغيرها لا تتسامح مع مشاكل البرمجيات والخطأ مكلف للغاية ، لذلك ونظرا للعديد من المشاكل المتعلقة بأعداد البرمجيات، فقد نشأت في عالم الحاسوب أزمة تسمى بأزمة البرمجيات . فرغم التقدم الكبير في اعداد البرمجيات الا ان كثيرا من الصعوبات لا زالت موجودة وتشكل اعاقا لعملية اعداد برمجيات حاسوب عالية الجودة .

### والمشاكل التي غالبا ما كانت وراء أزمة البرمجيات هي :

(The problems that were often behind the Software Crisis are :)

- الجدول الزمني وتقدير التكاليف غير دقيق حيث يصل التأخير الى اشهر او سنوات.
- جودة البرمجيات احيانا كانت اقل من المستوى المطلوب.
- متطلبات المستخدم احيانا لا تتحقق.
- انتاجية المبرمجين لم تواكب الازدياد المستمر في طلب الحاجات، والزبائن لم يكونوا دائما راضين عن البرمجيات الجديدة بسبب معدل الأخطاء.
- لم يأخذ منتجو البرمجيات الوقت الكافي لجمع البيانات والمتطلبات ذات العلاقة بعملية الانتاج.
- البرمجيات المتوفرة كانت صعبة الصيانة بدرجة كبيرة .
- البرمجيات التي تم تسليمها كانت تأتي متأخرة وبتكلفة أعلى من المتوقع.(هذه المشكلة قد تكون ذات علاقة بالمشكلة الاولى).
- البرمجيات المتوفرة لم يتم اختبارها في الاغلب بشكل صحيح وبشكل كامل وبالتالي فهي غير موثوق بها.

## ( What are the reasons for Software Crisis ) ما هي اسباب الأزمة ؟

### تتمثل اسباب الأزمة فيما يلي :

- 1- عدم استقرار متطلبات المستخدم
- 2- تغيير المتطلبات ( Requirement change ) بصورة اسرع مما يجب .
- 3- ادوات العتاد والبرمجيات (S/W) تتغير بمعدل اسرع .
- 4- بعض المدراء المسؤولين عن اعداد البرمجيات ليس لديهم خلفية كافية عن منهجيات اعداد البرمجيات (المنظومات).
- 5- نقص التواصل بين معدي البرامج والزملائن .
- 6- التقدير السيء للامكانيات.
- 7- التخطيط السيء للمشروع .
- 8- الطرق الضعيفة لقياس جودة البرمجيات .
- 9- الاهداف والغايات ذات تعريف غامض .
- 10- عدم التوافق مع المعايير القياسية .

## 1.9 The Attributes of Good Software

As well as the service which they provide software products have a number of other associated attributes which reflect the quality of that software.

These attributes are not directly concerned with what the software does; rather they reflect its behavior which it is executing and the structure and organization of the source program and associated documentation. Examples of these attributes (some time called non-functional attributes) are the software's response time to use query and the understandability of the program code. The specific set of attributes which you might expect from a software system obviously depends on its application. Therefore a banking system must be secure, an interactive game must be responsive, a telephone switching system must be reliable, etc. these can be generated in the following attributes:

1- **Maintainability:** software should be written in such a way that it may evolve to meet the changing needs of customer. This is critical attribute because software change is an Inevitable

2- **Dependability:** software dependability has a range of characteristics, including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure.

3- **Efficiency:** software should not make wasteful use of system resources, such as memory and processor cycles. Therefore efficiency includes responsiveness, processing time, memory utilization etc.

4- **Usability:** software must be usable, without under effort by the type of user for whom it is designed. This means that it should have an appropriate user interface and adequate documentation.

## سمات البرمجيات الجيدة (The Attributes of Good Software)

### (Non-Functional Attributes of Good Software)

فضلا عن الخدمة التي يقدمونها منتجين البرمجيات لدينا عدد من الصفات التي تعكس نوعية تلك البرامج المرتبطة بها. هذه الصفات ليست المعنية مباشرة بما تعمله البرمجيات، بل تعكس سلوكها الذي ينفذ وهيكل وتنظيم البرنامج الاساسي (المصدر) والوثائق المرتبطة بها. ومن أمثلة هذه الصفات (في بعض الاحيان تدعى سمات غير وظيفية) هي وقت البرمجيات للرد على استعلام المستخدم وفهم كود البرنامج . وهي مجموعة محددة من الصفات التي قد تتوقع من نظام البرمجيات ان يعتمدها بوضوح في تطبيقه. مثال : ( لذلك يجب على النظام المصرفي ان يكون آمن، يجب أن تكون اللعبة التفاعلية مستجيبة، يجب أن يكون نظام التحويل الهاتفي موثوق به، الخ من الصفات .

### تتصف البرمجيات الجيدة في عملية بنائها وتطويرها بالصفات التالية :

#### 1- البرمجيات يجب ان تكون قابلة للصيانة (The Software Should be Maintainable)

**Maintainability** : الصيانة وهي قدرة النظام على تلبي الاحتياجات المتغيرة للعملاء بسهولة وبدون الكلفة العالية في البناء . وهي السمة الهامة لتغيير البرمجيات والتي لا مفر منها

#### 2- البرمجيات يجب ان تكون ذات درجة عالية من الوثوقية

#### (The Software Should be Reliable)

**Dependability**: الاعتمادية او الموثوقية لديها مجموعة من الخصائص، بما في ذلك الموثوقية والأمن والسلامة. اي تعني ان البرمجيات الموثوقة لا ينبغي أن تسبب الضرر المادي أو الاقتصادي في حال فشل النظام .

#### 3- البرمجيات يجب ان تكون كفوءة (The Software Should be Efficient)

**Efficiency** : الكفاءة وهي ان البرمجيات تبني بدرجة كفاءة عالية مع مراعاة الكلفة المستخدمة . وتعني لا ينبغي في البرمجيات الإسراف في استخدام موارد النظام مثل الذاكرة والمعالج . لذلك تتضمن كفاءة الاستجابة و وقت المعالجة و استخدام الذاكرة الخ .

#### 4- البرمجيات يجب ان تمتلك واجهات تخاطبية عالية المستوى

#### (The Software Should be offer appropriated user interface)

**Usability**(سهولة الاستخدام) : يجب أن تكون المنظومة قابلة للاستخدام، دون جهد حسب نوع المستخدم الذي تم التصميم له . وهذا يعني أنه ينبغي أن يكون لها واجهة تخاطبية مناسبة (user interface ) وملفات التوثيق تكون كافية لان المستخدم هو شخص غير خبير بعملية البناء وما يهمله هو كيفية استخدام البرمجيات لتلبي احتياجاته الخاصة .

## 1.10 The Goals of Software Engineering /

### The Objectives of Software Engineering

#### أهداف هندسة البرمجيات :

لهندسة البرمجيات أهداف يسعى لتحقيقها، وأسس يستند عليها لتحقيقها .

ومن أهم الأهداف التي يسعى إليها علم هندسة البرمجيات الآتي:-

- التحسين في جودة البرمجيات .

- زيادة الإنتاجية في إعداد البرمجيات.

- تحسين طرق الاتصال بين المستخدم ومعد البرنامج.

- التقليل من تكلفة المنتجات البرمجية .

ولكي تتحقق هذه الأهداف فمن المهم توفير درجة عالية من التواصل بين الزبائن والمستخدمين ورئيس وأعضاء إعداد البرمجيات (Development team) لأن التواصل الجيد (الشفوي والكتابي والشخصي) له أهمية كبيرة في هندسة البرمجيات .

ومن المهم أيضا الدراية بعلم الاقتصاد والإدارة فهو يوفر الأساس لتقدير الموارد والمستلزمات والتحكم في التكلفة . ولاشك أن المعرفة الأساسية بالعوامل البشرية ضرورية، فهي تؤثر ليس فقط على العمل الجماعي والإبداع، ولكنها تؤثر أيضا على مستويات وقدرات المستخدمين لغرض الحصول على برمجيات يتم قبولها بسهولة من قبل الزبون . كما يجب أن يكون مهندس البرمجيات (Software Engineer) قادرا على كتابة الوثائق الفنية بوضوح وبدقة .

### Foundations of Software Engineering ( اسس هندسة البرمجيات )

تعتمد هندسة البرمجيات بصورة كبيرة على معرفة المجالات التالية :

1- علم الحاسوب (Computer Science)

2- علم الإدارة ( Management Science )

3- مهارات الاتصال (Communication Skills)

4- الأسلوب الهندسي ( Engineering Discipline )

5- علم الاقتصاد (Economics)

6- علم النفس (Psychology)

7- الكتابة الفنية او الكتابة التقنية (Technical Writing)

## **Questions for Chapter One:**

- 1- What are forms for Software? / What are the parts of the Software?
- 2- What does the term Software Engineering?
- 3- What are Software Characteristics?
- 4- Remember at least one difference between the Programmer and Software Engineer.  
( اذكر فرقا واحدا على الأقل بين المبرمج ومهندس البرمجيات )
- 5- What are Characteristic of Software Engineer?
- 6- What are Software Applications?
- 7- List the only five of Software Engineering Applications.(اسرد خمسة فقط من تطبيقات هندسة البرمجيات)
- 8- Remember only five of types of Software Engineering and Applications its.  
(تذكر فقط خمسة أنواع من هندسة البرمجيات وتطبيقاتها)
- 9- What are Applications for Business Software?
- 10- What are Applications for Artificial Intelligence Software?  
(ما هي تطبيقات برمجيات الذكاء الاصطناعي)
- 11- Explain the Evolving Role of Software.
- 12- What does the term Software Crisis? (ماذا يعني مصطلح ازمة البرمجيات)
- 13- What are the problems that were often behind the software crisis?  
(ماهي المشاكل التي غالبا ما كانت وراء ازمة البرمجيات)
- 14- What are the reasons for Software Crisis? (ماهي اسباب ازمة البرمجيات)
- 15-What are Attributes of Good Software?
- 16-What are Non-Functional Attributes of Good Software?
- 17- What are the Goals of Software Engineering? (ماهي اهداف هندسة البرمجيات)
- 18- What are the Objectives of Software Engineering? (ما هي اهداف هندسة البرمجيات)
- 19-What are Foundations of Software Engineering? (ماهي اسس هندسة البرمجيات)
- 20- What is characterized the fourth generation of the evolving role of Software?  
(بماذا يتميز الجيل الرابع من تاريخ البرمجيات )
- 21- What is characterized the third Era of the evolving role of Software?  
(بماذا تتميز الحقبة الثالثة من الدور المتطور للبرمجيات؟)