# هندسة البرمجيات

## المحاضرة الاولى

## مقدمة عن هندسة البرمجيات

مدرس المادة: م.م نور حسن حسون

# المقدمة

## البرمجيات ( Software)

- الحاسب الآلي بدون برمجيات كإنسان بلا روح ، كما أن التطور الذي يحدث في أجهزة الحاسب الآلي ومكوناته يصاحبه أيضا تطور وتحديث دائم في عالم البرمجيات.

- والبرمجيات بصفة عامة هي عبارة عن مجموعة من الأوامر المرتبة منطقيا ، ويتم تنفيذها بواسطة وحدة المعالجة المركزية للحاسب الآلي ، ويختلف مستوى ونوع البرمجيات طبقا لعلاقاته وقربه من الحاسب الآلي من ناحية ، أو من قربه وعلاقته بالمستخدم من ناحية أخرى.

- فنجد أن نظام التشغيل Operting Systems بشكله الأولي هو الملتصق مباشرة بوحدة المعالجة المركزية CPUبينما نجد على الطرف الآخر ، التطبيقات البرمجية Applications هي الأكثر قربا وسهولة بالنسبة للمستخدم.

# المقدمة

**أنواع المنتجات البرمجية؟**

- **برمجيات عامة شاملة(Generic)** : وهي نظم مستقلة تنتج بواسطة شركات وتباع في السوق لأي عميل، وأحياناً يطلق عليها أسم البرمجيات المغلفة مثل قواعد البيانات ومعالجات النصوص(word) وحزم الرسوم(Paint) .

- **برمجيات جاهزة (تفصيل أو مخصصة Customized)** : وهي نظم مخصصة لعميل معين يطلب تجهيزها، ويتم تطويرها بواسطة شركة أو مطور خصيصاً لهذا العميل ومنها أنظمة التحكم في المعدات الإلكترونية والآلآت والنظم الخاصة بأعمال معينة.

# المقدمة

## انواع البرمجيات:

- هناك ثلاثة أنواع من البرمجيات هي برمجيات:

١. برمجيات النظم (System Software)

يتولى هذا النوع من البرمجيات العديد من تفاصيل إدارة نظام الحاسوب، مثل نظم التشغيل operating system، معالجة اللغات , Compiler ولغات البرمجة Programing Language

٢. البرمجيات التطبيقية (Application Software)

هذه البرمجيات تطوع الحاسوب من أجل تنفيذ وظائف مفيدة وخاصة مثل معالجة الحسابات باستخدام برنامج حسابات متخصص وإدارة المخازن وجدولة المواد الدراسية وغيرها من البرمجيات المتخصصة في إدارة الأعمال المختلفة ويتم شراء هذه البرمجيات حسب الطلب من شركات الحاسوب المعنية بالبرمجة .

# المقدمة

٣. برمجيات الأغراض العامة (General Software):

وهي البرمجيات التي يستطيع أي شخص أن يستخدمها ومن برمجيات الأغراض العامة برامج معالج النصوص مثل Word ،الجداول الإلكترونية Excel و حزم إدارة قواعد البيانات مثل Accessالتي تساعد في تنظيم واسترجاع.

# المقدمة

## عمليات البرمجيات Software Processes

هي مجموعة من الأنشطة التي تهدف إلى تنمية وتطوير البرمجيات. الأنشطة الرئيسية هي في عمليات البرمجية:

١. المواصفات Specifications: ما الذي يجب على النظام أن يفعله، وما هي قيود تطويره.

٢. التجهيز والتطوير Development : إنتاج نظام البرمجيات، أي أن نتيجة برمجيات يجب أن تحقق المواصفات.

٣. التحقق او التثبت Validation : فحص واختبار أن البرمجيات المنتجة تحقق المواصفات التي طُلِبَة من قبل العميل.

٤. تقييم البرمجيات Evolution : تغيير وتطوير البرمجيات استجابة للتغييرات الطارئة.

# المقدمة

## دورة حياة تطوير البرمجيات Software Lifecycle:

- عملية بناء أي منتج تمر بعدة مراحل يطلق علىها عادة "دورة الحياة " تتضمن المراحل التالية:

١. تحديد وتعريف المتطلبات Requirements analysis and definition .
٢. تصميم النظام System design.
٣. تصميم البرنامج Program design.
٤. كتابة البرنامج (تطويره) Program implementation.
٥. أختبار وحدات البرنامج Unit testing.
٦. أختبار النظام system testing.
٧. تسليم النظام system delivery.
٨. الصيا نة maintenance

# المقدمة

## خصائص البرمجيات الجيدة:

- يمكن وضع تعميم لخصائص البرمجيات الجيدة بحيث تحقق المواصفات التالية للبرمجيات الجيدة التى تسعى الغالبية العظمى من طرق هندسة البرمجيات والأدوات والتقنيات للمساعدة في إنتاج برمجيات تحقق هذه الخصائص وهي :

١. **قابلية الصيانة Maintainability:** يجب كتابة البرمجيات بطريقة تلبى احتياجات التغيير الى يحتاجها المستخدم، وهي خاصية حرجه لأن تغييرات البرمجيات نتاح حتمي لتغير بيئة العمل.

٢. **قابلية الاعتمادDependability:** تتضمن قابلية اعتمادي البرمجيات في نطاق من الخصائص تحتوى على الاعتماد عليها Reliabilityوالتأمين Security والأمان Safety، فالبرمجيات التى يعتمد عليها لا تسبب ضرراً أو تلفاً فيزيائياً او اقتصادياً في حالة حدوث انهيار النظام.

# المقدمة

**خصائص البرمجيات الجيدة:**

٣. **الكفاءة Efficiency** :لا يجب على البرمجيات إهدار موارد النظام System Resourcesمثل:

- الذاكرة و دورات المعالج Processor Cycles، وبهذا تتضمن الكفاءة .
- الاستجابة . Responsiveness
- وقت المعالجة . Processing Time

٤. **قابلة الاستخدام Usability** : يجب أن تكون البرمجيات قابلة للاستعمال بدون مجهود غير مستحق للمستخدمين المصمم لهم النظام ، وهذا يعنى ضرورة وجود واجهة مستخدم (user interface) مناسبة مع المستندات التي تشمل على التوثيق الكافية.

# المقدمة

## هندسة البرمجيات Software engineering

- هي فرع من فروع المعلوماتية يهدف الى تطوير مجموعة أسس وقواعد تهدف إلي تحسين طرق تصميم وتطوير البرامج علي جميع المستويات وذلك بطريقه تلبي احتياجات المستخدمين .

- وهندسة البرمجيات لا تهتم بكتابة البرنامج نفسه أي بكتابة شفرته بل تحاول تحسين عملية تطوير وصنع البرنامج ابتداء من المواصفات التي يضعها المحترف وانتهاء عند مشكلة صيانة البرنامج أو توسعته

- كما انها تقوم علي دراسة احتياجات المستخدم وتصميم البرنامج المناسب لها قبل كتابة شفرته، وهناك العديد من الجوانب كالقدرة علي تطوير البرنامج بسهولة لاحقا، أو السرعة، أ و إمكانية إضافة ملحقات له بشكل ديناميكي

# مفهوم هندسة البرمجيات

- كان الاستخدام الرسمي الاول لهذا المصطلح في مؤتمر عقد من قبل اللجنة العلمية في منظمة حلف شمال الاطلسي عام ١٩٦٨ حول البرمجيات.

- عقد المؤتمر لمعالجة ما يعرف « ازمة البرمجيات» والتي ظهرت بسبب استخدام الوسائل التقليدية في بناء البرمجيات مما ادى الى ظهور برمجيات تحتاج الى وقت كبير لتطويرها.

- المكونات البرمجية ( software) شيء غير ملموس الى حد ما بالمقارنة مع المنتجات الاخرى كالمكونات المادية (Hardware).

- تمثل المكونات البرمجية سلسلة من الاف او ملاين الاوامر التي تطلب من الحاسوب اجراء عمليات معينة مثل عرض المعلومات او اجراء الحسابات او تخزين البيانات.

# مفهوم هندسة البرمجيات

- هندسة البرمجيات (Software Engineering) فهي فرع من فروع الهندسة يقوم على مجموعة اسس وقواعد تهدف الى تصميم وتطوير البرامج بوفرة ونوعية عالية تلبي احتياجات المستخدمين.

- هذا الفرع من الهندسة يتميز بانه لا يحتاج الى راس مال كبير وبالتالي الخسارة فيه تكون قليلة على عكس بقية الفروع من الهندسة.

- كما انه لا يكفي لا يجاد برمجيات متكاملة وجيدة بالاعتماد على عمل شخص واحد وانما يتطلب ذلك فريقا من المهندسين الاكفاء.

-

# مفهوم هندسة البرمجيات

## علاقة هندسة البرمجيات بعلوم الحاسوب

- علوم الحاسوب: يقوم بالتركيز على المكونات المادية، انظمة التشغيل، لغات البرمجة، المترجمات (Compiler).

- هندسة البرمجيات: هو تخصص يستخدم تكنولوجيا الحاسوب والبرمجيات كأدوات لحل مشكلة معينة.

- يمكن اخذ مثال بسيط يبين الفرق بين هندسة البرمجيات وعلوم الحاسوب في علم الكيماء واستخدامه في حل المشاكل التي نقابلها في حياتنا اليومية:

# مفهوم هندسة البرمجيات

## علاقة هندسة البرمجيات بعلوم الحاسوب

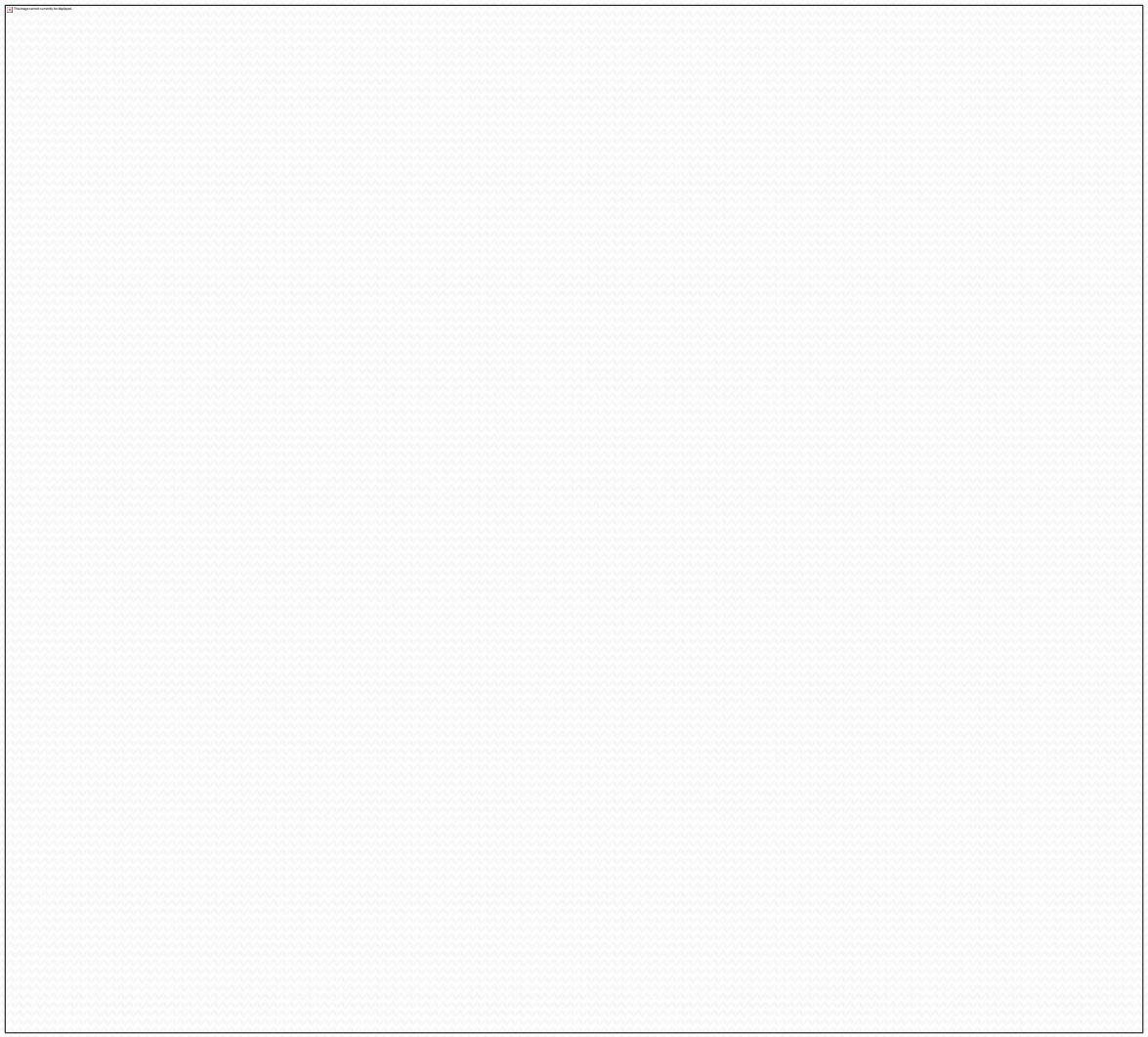
- يهتم الكيمائي بدراسة المواد الكيميائية ( تركيبها، تفاعلاتها، والنظريات التي تحكم سلوكها)

- بينما المهندس الكيميائي يستخدم النتائج التي توصل اليها الكيميائي لحل المشاكل التي يطلب منه ايجاد حل لها.

- اي انها من وجهة نظر الكيميائي هو موضوع الدراسة بحد ذاتها. بينما من وجهة نظر المهندس الكيميائي هي اداة Tool تستخدم لإيجاد حلول لمشكلة عامة ( قد لا تكون مشكلة ذات طبيعة كيميائية بحد ذاتها). الشكل التالي يبين العلاقة بين هندسة البرمجيات وعلوم الحاسوب.

# مفهوم هندسة البرمجيات

## علاقة هندسة البرمجيات بعلوم الحاسوب



# الفرق بين البرمجة وهندسة البرمجيات

- في البرمجة تعتبر عملية كتابة الكود هي اهم عملية في بناء البرنامج بغض النظر عن الجدوى من البرنامج او امكانية قبول المستخدم له او حتى قابلية التطوير.

- في حين ان هندسة البرمجيات تعمل على بناء النظام البرمجي كمشروع متكامل و دراسته من كافة الجوانب: البناء البرمجي، الدعم الفني والصيانة، التسويق والمبيعات، التطوير والتدريب على استخدامه.

# مهندس البرمجيات ( Software Engineer)

- مهندس البرمجيات هو الشخص الذي يقع على عاتقه تطوير المنتجات البرمجية التي تباع للزبائن او التي يحتاج اليها العملاء او الزبائن. لذلك يجب عليه ان يتبنى اسلوبا منظما ونظاميا في عمله لكي يحقق الاهداف المرجوة من استخدام علم هندسة البرمجيات.

- كما يتحتم عليه استخدام الادوات المناسبة ( **كاختيار لغة برمجية مناسبة من لغات البرمجة عالية المستوى)** والتقنيات الضرورية **(كالخوارزميات مختلفة الاغراض)** ويكون ذلك بالاعتماد على نوع المشكلة التي يقوم بحلها وقيود التطوير المفروضة والموارد المتاحة **(المادية والبشرية)**.

- يدرك مهندس البرمجيات انه يجب عليه العمل من خلال القيود التنظيمية والمالية. لذلك عليه ان يبحث عن الحلول ضمن هذه القيود.

# مهندس البرمجيات ( Software Engineer)

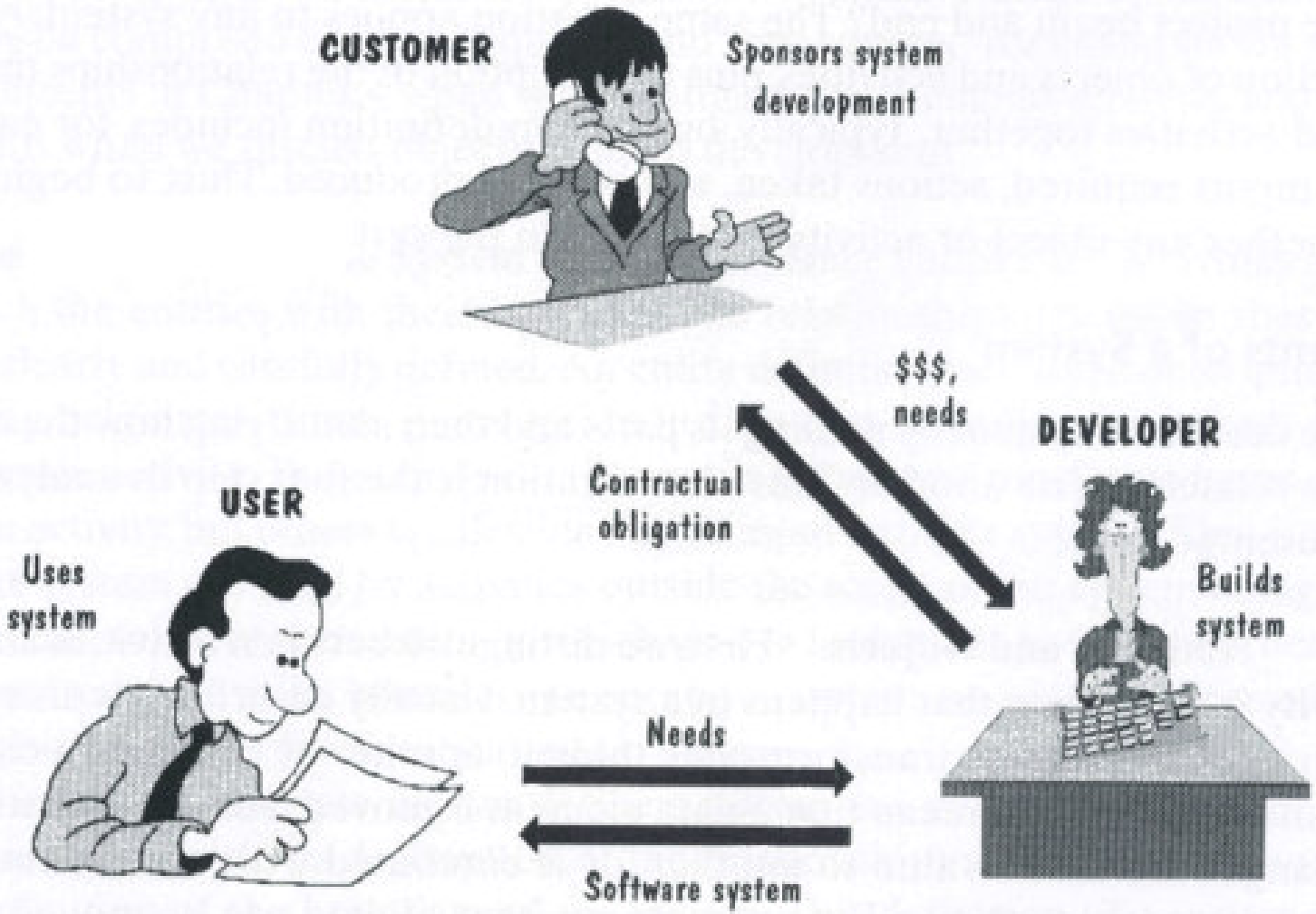
- لذلك فان هندسة البرمجيات تهتم بتصميم وتطوير برامج ذات جودة عالية. السؤال التي يجب سؤاله هنا: من يشارك في عملية التصميم وتطوير البرامج؟؟

- المشاركون في عملية صناعة البرنامج، عادة ما يندرجون تحت ثلاث مجموعات:

١. **الزبون (Customer)**: وهو الشركة (او الشخص) الممولة لمشروع تطوير البرنامج المطلوب.
٢. **المستخدم (User)**: الشخص ( او مجموعة الاشخاص)الذي سوف يقوم فعلا باستعمال البرنامج والتعامل معه مباشرة.

٣. **المطور ( Developer)**: وهو الشركة (او الشخص) الذي سوف يقوم بتطوير البرنامج لصالح الزبون. الشكل التالي يبن العلاقة بين الاجزاء المذكورة انفا:

# مهندس البرمجيات ( Software Engineer)



# مهندس البرمجيات ( Software Engineer)

- **سؤال : ما هي الاسئلة التي يجب على مهندس البرامجيات ان يسأل نفسه عند تصميم هذا للنظام ؟**

١. تغير العمليات Process Change : أي هل النظام المقترح سيؤدي الى تغير في شكل معالجة العمليات فمثلا العمليات الحسابية في احد الشركات فهل نظامنا سوف يؤدي الى تحول تلك العمليات الى عمليات الية داخل الحاسوب ام لا ؟

٢. تغير وظيفي Job Change: أي هل طبيعة العمل للموظفين ستتغير بمعنى اخر هل سيبقى الموظفون في وظائفهم بحيث سيحتاجون الى اعادة تاهيل ام لا؟

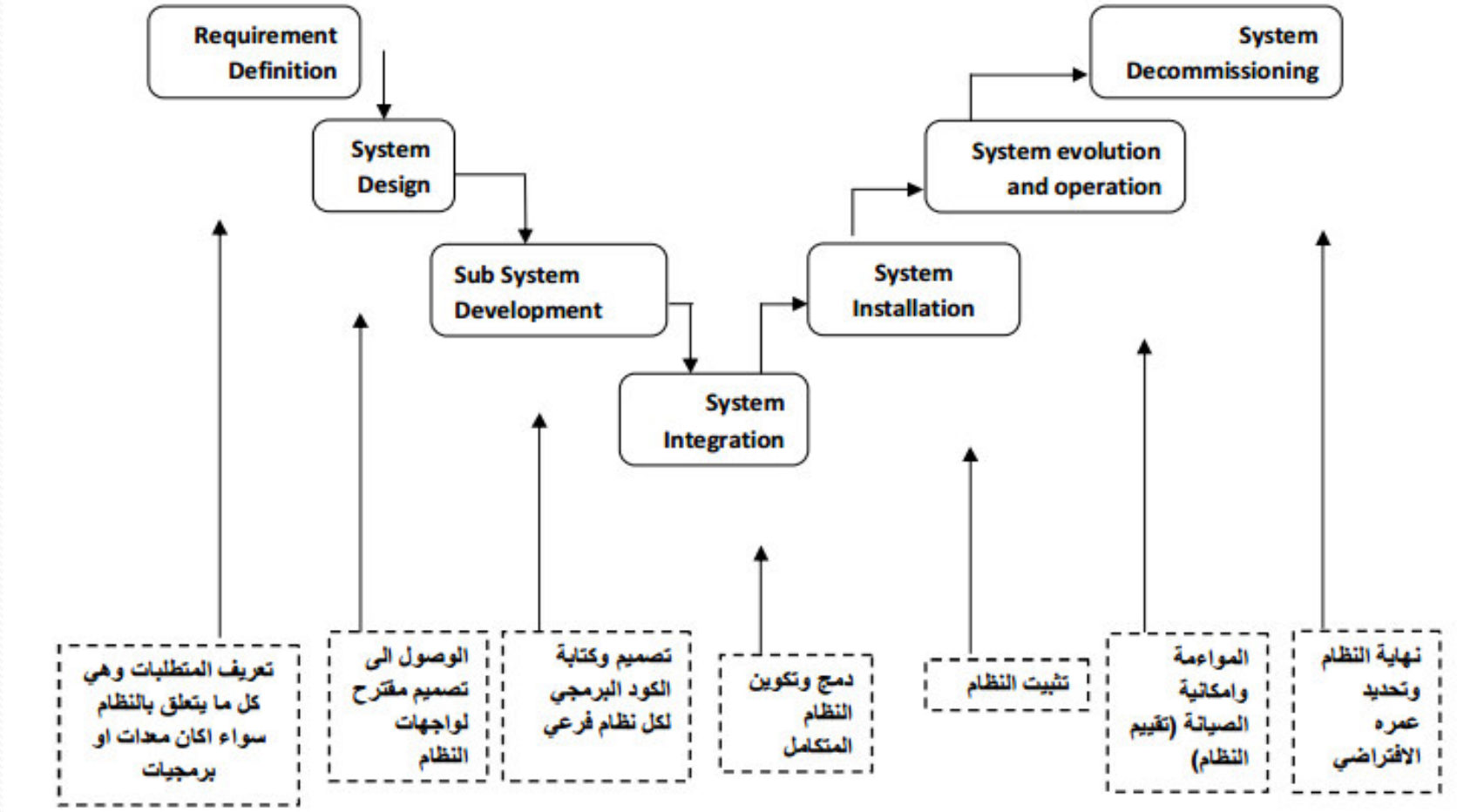
# مهندس البرمجيات ( Software Engineer)

٣. تغير المنظمة Organization Change: هل هذا النظام سيؤدي الى مايسمى بالتغيرات المنظمية أي هل ستتغير هيكلية المنظمة ام لا فعلى سبيل المثال ادارة الارشفة فهي مستقبلا ايلة الى الاختفاء مع ظهور انظمة قواعد البيانات ذات الامنية والكفائة العالية .

- وبعد ان يسأل مهندس البرامجيات نفسه الاسئلة السابقة ينتقل الى "خطوات هندسة البرامجيات".

# خطوات هندسة البرمجيات Software Engineering Process

- ويمكن توضيح تلك الخطوات في الشكل التالي:



# ما الفرق بين هندسة البرمجيات وهندسة النظم

**هندسة النظم** : تهتم هندسة النظم بجميع الجوانب المتعلقة بتطوير النظم المعتمدة على الحاسوب بما فيها من تصميم وبناء للتجهيزات والبرمجيات ومن عناية بإجرائية التطوير وتهتم بجميع نواحي الكمبيوتر من معدات وبرمجيات وعمليات هندسيه

**أما هندسة البرمجيات** فهي ذلك الجزء من هذه الإجرائية الذي يهتم بتطوير البنية البرمجية الأساسية والتحكم والتطبيقات وبناء المعطيات التي تدخل في تكوين هذا النظام وتعتبر هندسة البرمجيات هي جزء من الأنظمة الهندسية.

- الأنظمة الهندسية تهتم بالمواصفات ، التخطيط، التكامل والتطبيق للأنظمة.
- وتعتبر هندسة النظم أقدم من هندسة البرمجيات فقد تمكن المهندسون وخلال الأعوام المائة الماضية من بناء وتجميع العديد من الأنظمة الضخمة كالطائرات والمعامل.

# هندسة البرمجيات

## المحاضرة الثانية

## تكنولوجيا الطبقات، نماذج عمليات البرمجيات

مدرس المادة: م.م نور حسن حسون

# هندسة البرمجيات – تكنولوجيا طبقية:

- تعريف Fritz Bauer الذي وضعه لهندسة البرمجيات ينص علي التالي:

- ***(هندسة البرمجيات) هي استخدام مبادئ هندسية صحيحة للحصول اقتصاديا علي برمجيات موثوقة وتعمل بكفاءة علي كمبيوترات حقيقية.***

- هذا التعريف لا يقول الكثير عن المظاهر التقنية لجودة البرمجيات.
- لا يذكر مباشرة الحاجة الي ارضاء الزبون او تسليم المنتج في الوقت المناسب
- لم يذكر اهمية عملية البرمجة الناضجة

# هندسة البرمجيات – تكنولوجيا طبقية:

- وضع معهد المهندسين الكهربائيين والالكترونيين IEEE تعريفا اشمل وهو:

- ***(هندسة البرمجيات )هي تطبيق منهج مرتب ومنظم وقابل للقياس لعمليات تطوير وتشغيل وصيانة البرمجيات اي تطبيق الهندسة علي البرمجيات.***

- لذلك فان هندسة البرمجيات هي تكنولوجيا طبقية (Layered Technology) . ان حجر الاساس الذي يدعم هندسة البرمجيات هو التركيز على:

١. الجودة (Quality): وهي من اهم اهداف استخدام هندسة البرمجيات في عملية بناء وتطوير البرمجيات حيث تأخذ بعين الاعتبار تخصيصات المستخدم، ومتطلباته على جميع المستويات.

# هندسة البرمجيات – تكنولوجيا طبقية:

- **المعالجة( Process)**: هي عملية اختيار نوع المعالجة المستخدمة في بناء البرنامج وهنا يبدا مدير العمل (Software Developer) بتحديد نوعية وطريقة المعالجة المطلوبة مثلا هل يستخدم طريقة الخطية وتدفق الشلال ( Waterfall) Approachاو طريقة ( Exploratory Approach)او اي طريقة اخرى في العمل ؟ مع مراعاة تلائم نوعية المعالجة مع طبيعة النظام المراد بناؤه .

- **الطرق( Methods)** : هنا مصمم النظام يبدا باخيار طريقة العمل مثلا هل يريد استخدام ( Object _Oriented Methods) او طرق الكلاسيكية بالعمل وهذة الطبقة توفر تقنية الاجابة عن "How" كيف نبني البرانامج

# هندسة البرمجيات – تكنولوجيا طبقية:

- **الادوات(Tools) :** وهذه الطبقة توفر ادوات اتوماتيكية او شبة مؤتمتة في توفير ادوت وبرمجيات تساعد المصمم في عمله ومثال عنها الـ (Computer Aided Software Engineering (CASE))والمخطط التالي يبين طبقات هندسة البرمجيات.

مخطط (1) يوضح طبقات هندسة برامجيات

# نماذج عمليات البرمجيات Software Process Models

- النموذج عبارة عن تمثيل مبسط لدورة حياة تطوير النظام حيث تعرض هذه العمليات من منظور خاص. من امثلة منظور العمليات المستخدمة: منظور تتابع العمل وتتابع الاطوار، ومنظور تدفق العمليات وتدفق البيانات (تدفق المعلومات)، ومنظور قواعد واعمال( تحديد اعمال).

- والنماذج بطبيعتها هي تبسيط لدورة حياة النظام عبارة عن موجز مجرد للعمليات الفعلية الموصوفة، وقد يحتوي على الاطوار التي هي جزء من عمليات البرمجيات التي ينشغل بها العاملون في هندسة البرمجيات.

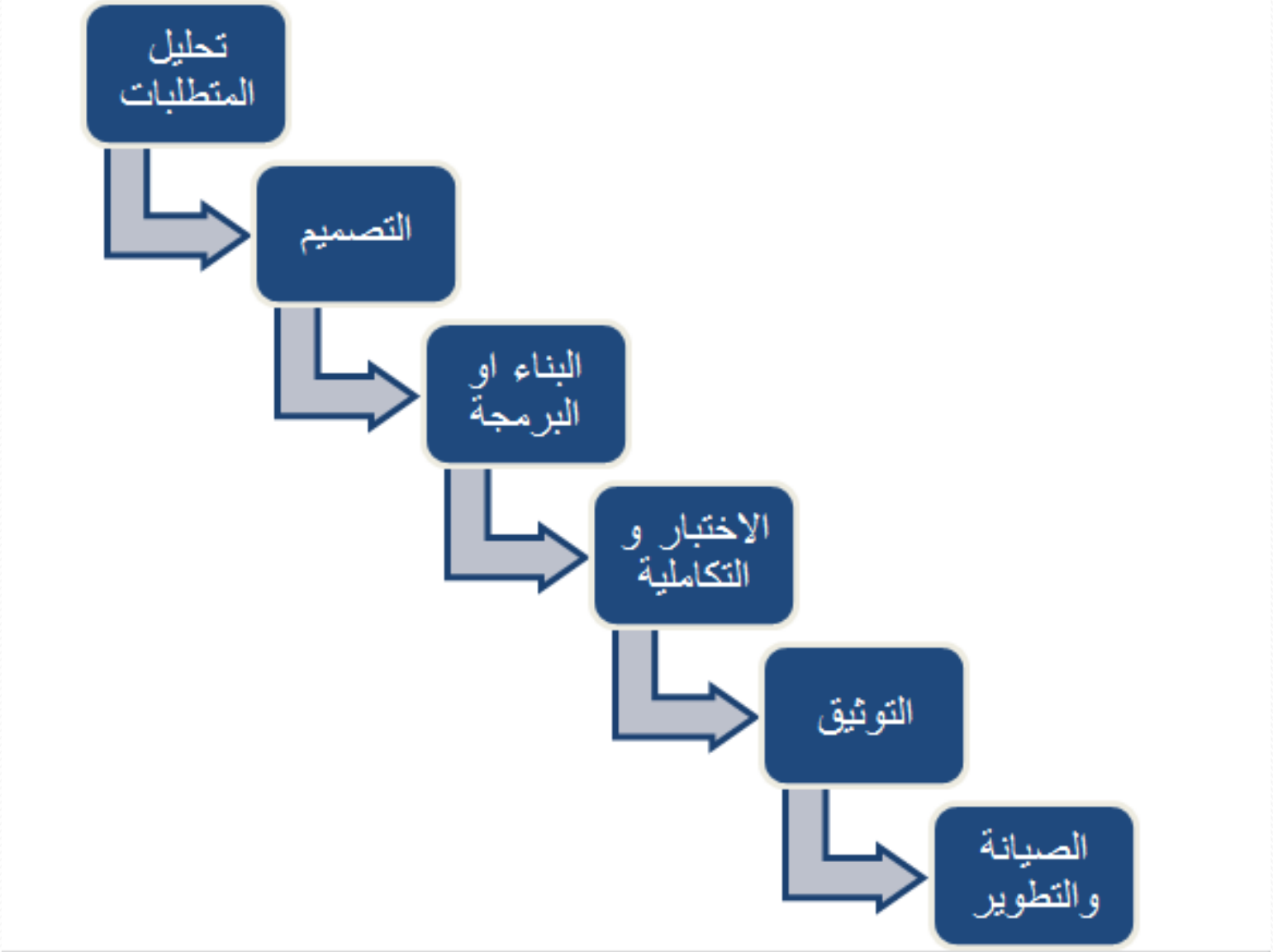
# نماذج عمليات البرمجيات Software Process Models

## ١. النموذج الانحداري او الشلالي Waterfall Model

- في هذا النموذج تسير دورة الحياة بشكل تدريجي بدأ من الخطوة (١) وحتى الخطوة (٨)، وكما يظهر بالشكل (١) فإن كل مرحلة تبدأ بعد الانتهاء من المرحلة التي تسبقها مباشرة.

- يتميز النموذج الانحداري بالبساطة، ولذا فإنه يسّهل على المطور توضيح كيفية سير العمل بالمشروع للعميل (الذي عادة لا يعرف الكثير عن صنع البرمجيات) والمراحل المتبقية من العمل. وقد كان هذا النموذج أساس عمل كثير من المؤسسات لفترة طويلة مثل وزارة الدفاع الامريكية، واستنبط منه العديد من النماذج الاكثر تعقيدا. الشكل التالي يبن مراحل تطوير البرمجيات في النظام الانحداري

# نماذج عمليات البرمجيات Software Process Models



# نماذج عمليات البرمجيات Software Process Models

- **مراحل النموذج**

١. **المرحلة الأولى : تحليل المتطلبات (Requirements Analysis)**

- في هذه المرحلة يقوم محلل النظام او مهندس البرمجيات بتحديد متطلبات النظام من برمجيات، ومعدات ،و المهام والوظائف التي سيقوم بها البرنامج ، وصف هذه المهام بدقة تامّة ،كما يتم عمل دراسة الجدوى لهذا البرنامج

- ، فالعميل في هذه المرحلة يضع تصوراً للبرنامج ليقوم بعمليات معينة وهنا تأتي مهمة مهندس البرمجيات في استخلاص هذه الأفكار الخاصة بالعميل وتحديدها ، لذلك فهي تتطلب مهارة عالية في التعامل مع العملاء والقدرة على التحليل الصحيح. ينتج في نهاية هذه المرحلة وثيقة تدعى جدول الشروط والمواصفات.

- حسب النموذج الإنحداري فإنه يجب على المطورين إنهاء مرحلة تحليل النظام بشكل تام قبل البدء في التصميم، هذه المرحلة قد تتطلب وقت طويل في بعض المشاريع وقد تمر عدة سنوات قبل أن يرى البرنامج النهائي النور.

# نماذج عمليات البرمجيات Software Process Models

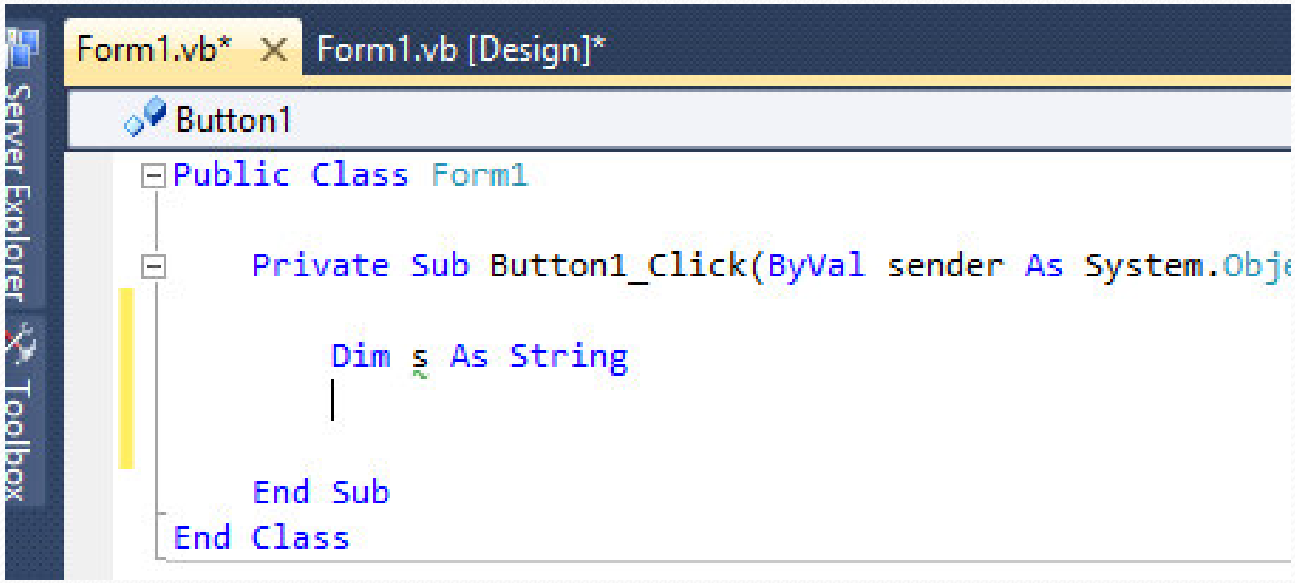
١. **المرحلة الثانية : التصميم (Design)**

- عملية التصميم العام للنظام هي عبارة ترتيب النوافذ المختلفة للنظام وجعلها تعمل بطريقة متكاملة لتحقيق الأهداف الخاصة بالنظام ، ويجب دراسة وتقويم مجموعة من العناصر الهامة والمؤثرة في عملية التصميم والتي سوف نتعرف عليها بشكل أكثر تفصيلاً في المقالات التالية.

٣. **المرحلة الثالثة : البناء او البرمجة (Implementation and coding)**

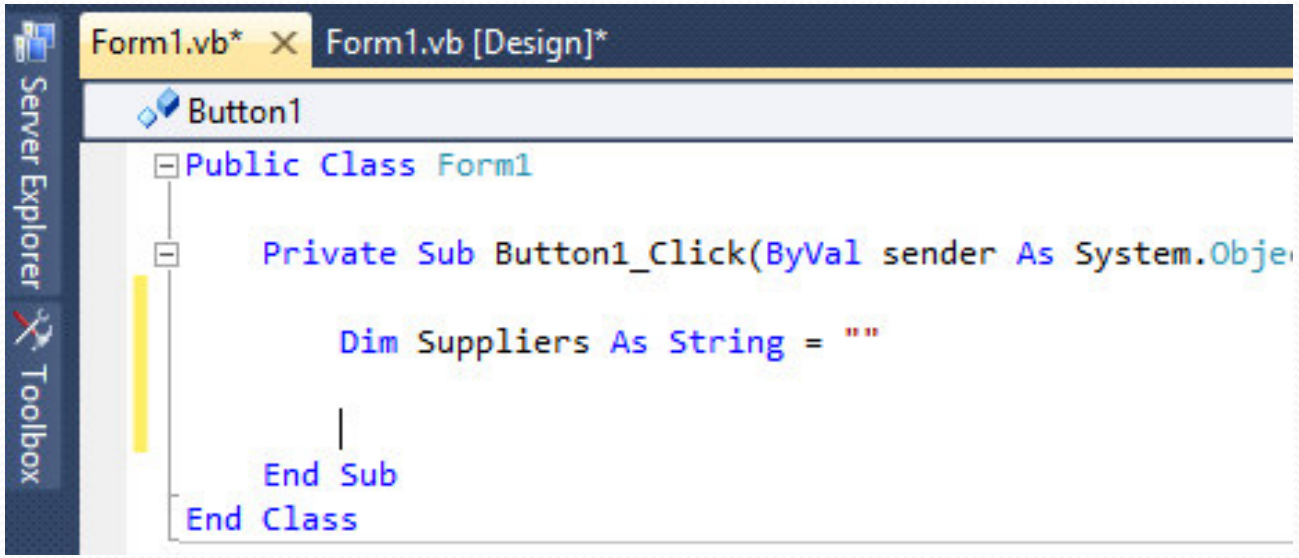
- مرحلة كتابة الأكواد البرمجية تعد جزءاُ من التصميم ، حيث يجب عليك كمبرمج تمثيل الأكواد البرمجية في مخيلتك أثناء تصميم النوافذ ، كما يجب وضع الموصفات القياسية لكتابة الأكواد البرمجية في عين الاعتبار ،مثل التسميات للمتغيرات والثوابت بشكل واضح وبدون اختصار فإن كنت مثلا ستصرح عن متغير خاص بالموردين فلا تقوم بالتصريح عنه كالتالي – باستخدام فيجوال بيسك دوت نت :-

# نماذج عمليات البرمجيات Software Process Models




- فتخيل مثلا أن هناك مبرمج آخر سوف يقوم بالقراءة والتعديل على الأكواد فماذا سيفهم من متغير باسم (S) ؟.لذلك وجب عليك التصريح عن المتغير بشكل يوحي بمضمون عمل هذا المتغير كالتالي – باستخدام فيجوال بيسك دوت نت :

# نماذج عمليات البرمجيات Software Process Models




٤. **المرحلة الرابعة : الاختبار و التكاملية (Testing and Integrative)**

- يتم جمع الكتل البرمجية مع بعضها و اختبار النظام البرمجي للتأكد من موافقته للشروط و المواصفات المتفق عليه عند تحليل النظام ، و خاصة اذا كانت الكتل البرمجية قد صممت وكتبت أكوادها من قبل عدة أعضاء في فريق تطوير النظام .

# نماذج عمليات البرمجيات Software Process Models

٥. **المرحلة الخامسة : التوثيق (Documentation)**

- وهي مرحلة هامة من مراحل بناء النظام البرمجي حيث يتم توثيق البناء الداخلي للبرنامج وذلك بغرض الصيانة والتطوير، ويفضل عادة أن يترافق التوثيق مع كل مرحلة من المراحل السابقة واللاحقة، وأن يكون هناك فريق خاص يهتم بعملية التوثيق لجميع المشاكل والحلول التي يمكن أن تظهر أثناء بناء النظام البرمجي

- وبدون التوثيق قد يصل مطور النظام البرمجي إلى مرحلة لا يعود بعدها قادراً على متابعة صيانة النظام وتطويره مما يزيد من الكلفة المادية والزمنية الخاصة بهذه البرمجية إلى حدود غير متوقعة، أو بمعنى آخر الفشل في بناء نظام برمجي ذات جودة عالية .

# نماذج عمليات البرمجيات Software Process Models

٦. **المرحلة السادسة : الصيانة و التطوير (Maintenance and development)**

- إن هذه المرحلة هي المرحلة الأطول في حياة النظام البرمجي لبقاء النظام قادراً على مواكبة التطورات و المعدات الحديثة، جزء من هذه المرحلة يكون في تصحيح الأخطاء البرمجية التي تظهر من كثرة الاستخدام وإدخال الكثير من البيانات والجزء الآخر يكون في التطوير و إضافة إمكانيات جديدة.

# نماذج عمليات البرمجيات Software Process Models

- **ايجابيات النموذج (Model Advantages)**

١. نموذج سهل للفهم
٢. سهل للإدارة
٣. المراحل تكتمل وتعالج مرحلة تلو الاخرى
٤. العمل يقسم الى مشاريع صغيرة حيث المتطلبات تصبح سهلة للفهم
٥. يفضل في المشاريع حيث الجودة هي اكثر اهمية بالمقارنة مع الجدول الزمني والتكلفة

# نماذج عمليات البرمجيات Software Process Models

- **السلبيات النموذج (Model Disadvantages)**

١. لا يمكن ان تعود خطوة ، اذا مرحلة ما يوجد بها خطأ لا يمكن الرجوع اليها وتعديلها لا نها تصبح العمليات معقدة في المرحلة، حتى لو تغيير بسيط في اي مرحلة سابقة يمكن ان يسبب مشاكل كبيرة للمراحل اللاحقة حيث ان جميع المراحل تعتمد على بعضها.

٢. نسبة كبيرة من المخاطر واحتمال حصول اخطاء

٣. ليس نموذج جيد للمشاريع المعقدة

٤. نموذج ضعيف للمشاريع الطويلة والمستمرة

٥. غير مناسب للمشاريع حيث المتطلبات فيها مخاطر عالية من التغيير