

Yarmouk University  
Faculty of Information Technology and Computer Sciences  
Department of Computer Information Systems

# Software Engineering

## هندسة البرمجيات

### CIS 240

تولمة: أسامة محمد العز

*Mobile: 0796484613*

*E-mail: osa\_alkhoun82@yahoo.com*

ﺗﺘﺒﯿﻪ ﻫﺎﻡ

ﺗﻢ ﺗﺮﺟﻤﺔ ﻫﺬﺍ ﻣﺴﺎﻕ ﻟﻤﺴﺎﻋﺪﺓ ﻃﺎﻟﺐ ﻋﻠﻰ ﻓﻬﻢ ﻣﻮﺍﺿﯿﻊ ﻣﺘﻌﻠﻘﺔ ﺑﻪ

ﻭﻫﺬﺍ ﻻ ﻳﻌﻨﻰ ﺃﻧﻚ ﻏﯿﺮ ﺑﺤﺎﺟﺔ ﻟﻔﻬﻤﻬﺎ ﺃﻳﺸﺎً ﺑﺎﻟﻠﻐﺔ ﺍﻟﺌﯩﻨﺠﯩﺰﯨﻴﺔ

ﻣﻊ ﺗﻤﻨﯩﺎﺗﻰ ﻟﻠﺠﻤﯿﻊ ﺑﺎﻟﺘﻮﻓﯿﻖ.

## Chapter Two

### الفصل الثاني

## Software Processes

### معالجات البرمجيات

## Software Processes (Chapter 4 from the textbook)

معالجة البرمجيات ( الفصل الرابع من الكتاب )

### Objectives

#### الأهداف

- To introduce software process models  
• لتقديم نماذج معالجة البرمجيات
- To describe Some generic process models and when they may be used  
• لوصف بعض تصميمات المعالجة العامة ومتى قد تستخدم
- To describe outline process models for requirements engineering, software development, testing and evolution  
• لوصف نماذج خلاصة المعالجة لمتطلبات الهندسة وتطوير البرمجيات، والاختبار والتطور
- To explain the Rational Unified Process model  
• لتوضيح تصميم نموذج المعالجة المنطقي الموحدة
- To introduce CASE technology to support software process activities  
• لتقديم تقنية هندسة البرمجيات بمساعدة الحاسوب لدعم نشاطات معالجة البرمجيات

### Topics covered

#### المواضيع تغطي

- Software process models  
• تصاميم معالجة البرمجيات
- Process iteration  
• المعالجة المكررة
- Process activities  
• نشاطات المعالجة
- The Rational Unified Process  
• المعالجة المنطقية الموحدة
- Computer-aided software engineering  
• هندسة البرمجيات بمساعدة الحاسوب
- Agile development methods and processes.  
• طرق وعمليات التطوير السريعة

## The software process

### معالجة البرمجيات

- A structured set of activities required to develop a software system. Major activities or stages include: Requirements, Design, Implementation, Testing, and Deployment.
  - المجموعة المنظمة من النشاطات تتطلب تطوير نظام البرمجيات. النشاطات أو المراحل الرئيسية تتضمنان: المتطلبات، التصميم، التطبيق، الاختبار، والانتشار
- Each stage of those listed may have one or more sub stages. For example, Requirement may include: communication, elicitation, specification, and verification. Design may include architectural and detail design.
  - كل مرحلة من هذه القائمة تدرج تحت مرحلة فرعية أو أكثر. على سبيل المثال، المتطلب قد يتضمن: الاتصال، التغيير، المواصفات، والتحقق. التصميم قد يتضمن المعمارية وتفاصيل التصميم
- In some projects, a stage maybe shortened, but it can never be ignored.
  - في بعض المشاريع ، ربما تكون المرحلة قصيرة، لكن لا يمكن أن يتم تجاهلها

## Framework Activities; different terminologies

### إطار النشاطات؛ المصطلحات المختلفة

- In different books or references, some of those processes may be mixed, combined, or have different names.
  - في الكتب أو المراجع المختلفة، البعض من تلك المعالجات قد تخط، تجمع، أو تمتلك أسماء مختلفة
- Communication
  - الاتصال
- Planning
  - التخطيط
- Modeling (Design and specs)
  - النمذجة ( التصميم و المواصفات )
  - Analysis of requirements
    - متطلبات التحليل
  - Design
    - التصميم
- Construction (Implementation and testing)
  - البناء ( التطبيق والفحص )
  - Code generation
    - توليد الشيفرة البرمجية
  - Testing
    - الفحص
- Deployment
  - النشر

## Umbrella Activities

### النشاطات الشاملة

- Those are considered as – secondary processes. Some projects may or may not include them.
  - تعتبر هذه - المعالجة الثانوية. في بعض المشاريع يمكن أن تتضمنهم أو لا
- Software project management
  - إدارة مشاريع البرمجيات
- Formal technical reviews
  - المراجعات التقنية الرسمية
- Software quality assurance
  - تأكيد الجودة للبرمجيات
- Software configuration management
  - إدارة تعريف البرمجيات
- Work product preparation and production
  - عمل التحضيرات والتصنيع للمنتج
- Reusability management
  - إدارة إعادة الاستخدام
- Measurement
  - القياس
- Risk management
  - إدارة المخاطر

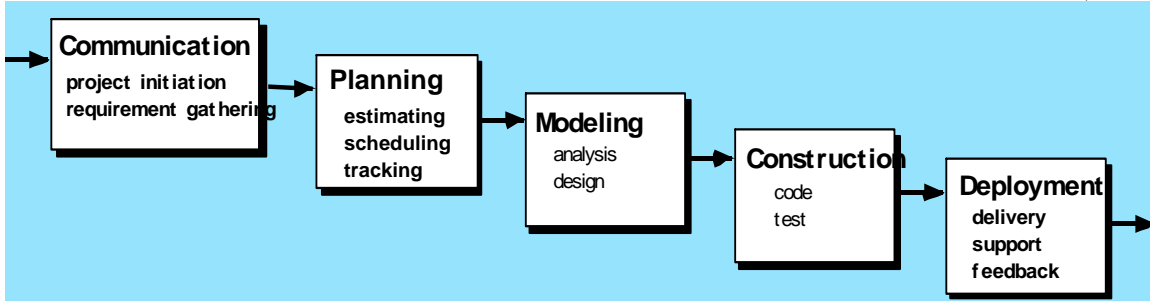
## Generic software process models

### نماذج معالجة البرمجيات العامة

- The waterfall model: Separate and distinct phases of specification and development. Does not go to the next phase until completing the current phase.  
• نموذج الشلال: المراحل المنفصلة والمتميزة للمواصفات والتطوير. لا يذهب إلى المرحلة التالية حتى إكمال المرحلة الحالية.
- Evolutionary development (Prototyping).  
• التطوير التطوري (النموذج الأولي).
  - Specification, development and validation are interleaved. Develop “prototypes” for customers to see and verify.  
• المواصفات، تطوير وتصديق المدخلات. تطوّر "النماذج الأولية" للزبائن لرؤيتها والتحقق منها
- Component-based software engineering  
• هندسة البرمجيات المعتمدة على المكونات
  - The system is assembled from existing components off-shelf components.  
• النظام مجمّع من المكونات الموجودة والمكونات الجاهزة
- There are many variants of these models e.g. formal development where a waterfall-like process is used but the specification is a formal specification that is refined through several stages to an implementable design.  
• هناك العديد من هذه النماذج المتغيرة ومثال على ذلك: - تطوير رسمي مثل عملية الشلال التي تستخدم. لكن المواصفات تكون مواصفات رسمية التي تكون مرتبة خلال عدّة مراحل إلى التصميم القادر على التطبيق.

## The Waterfall Model

تصميم الشلال



We don't go back to a stage once it is completed, and "usually" we don't start a stage until the previous stage is completed.

لا يتم العودة إلى المرحلة عندما تكتمل، و"عادة" لا يتم البدء بمرحلة حتى تكون المرحلة السابقة منتهية

### Waterfall model phases

مراحل تصميم الشلال

- Requirements analysis and definition  
تحليل المتطلبات وتعريفها
- System and software design  
تصميم البرمجيات والنظام
- Implementation and unit testing – developers tasks.  
التطبيق ووحدة الفحص – مهام المطورين
- Integration and system testing- testers tasks.  
التكامل ونظام الفحص – مهام الفاحص
- Operation and maintenance. Customer support and Customer service tasks.  
العمليات والصيانة. دعم الزبائن ومهام خدمات الزبائن
- We use the waterfall model when we have fixed, stable and known requirements.  
يتم استخدام تصميم الشلال عندما يكون هناك أخطاء (مشاكل)، دائم أو متطلبات غير معروفة
- The main drawback of the waterfall model is the difficulty of accommodating changes after the process is underway. One phase has to be completed before moving to the next phase.  
إنّ العائق الرئيسي لنموذج الشلال هو صعوبة تغيير التغييرات بعد معالجة المرحلة الجارية (الحالية). مرحلة واحدة يجب أن تكتمل قبل الانتقال إلى المرحلة القادمة (التالية)



## Waterfall model problems

### مشاكل تصميم الشلال

- Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements (not suitable for unstable requirements).
  - التقسيم الغير مرن للمشروع إلى المراحل المتميّزة تجعل الأمر صعباً للردّ على تغيير متطلبات الزبون (ليست مناسبة للمتطلبات الغير مستقرة).
- Therefore, this model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.
  - لذا، هذا النموذج ملائم فقط عندما تكون المتطلبات مفهومة بشكل جيد والتغييرات ستكون محدودة جداً أثناء عملية التصميم
- Few business systems have stable requirements.
  - بعض أنظمة الشركات لها متطلبات مستقرّة ( لا تحتوي على تعديلات)
- The waterfall model is mostly used for large systems engineering projects where a system is developed at several sites. (Such as space or govt projects).
  - إنّ نموذج الشلال يستخدم في الغالب لهندسة الأنظمة الكبيرة المشاريع حيث أنّ النظام تم تطويره في عدّة مواقع. (مثل الفضاء أو المشاريع الحكومية).

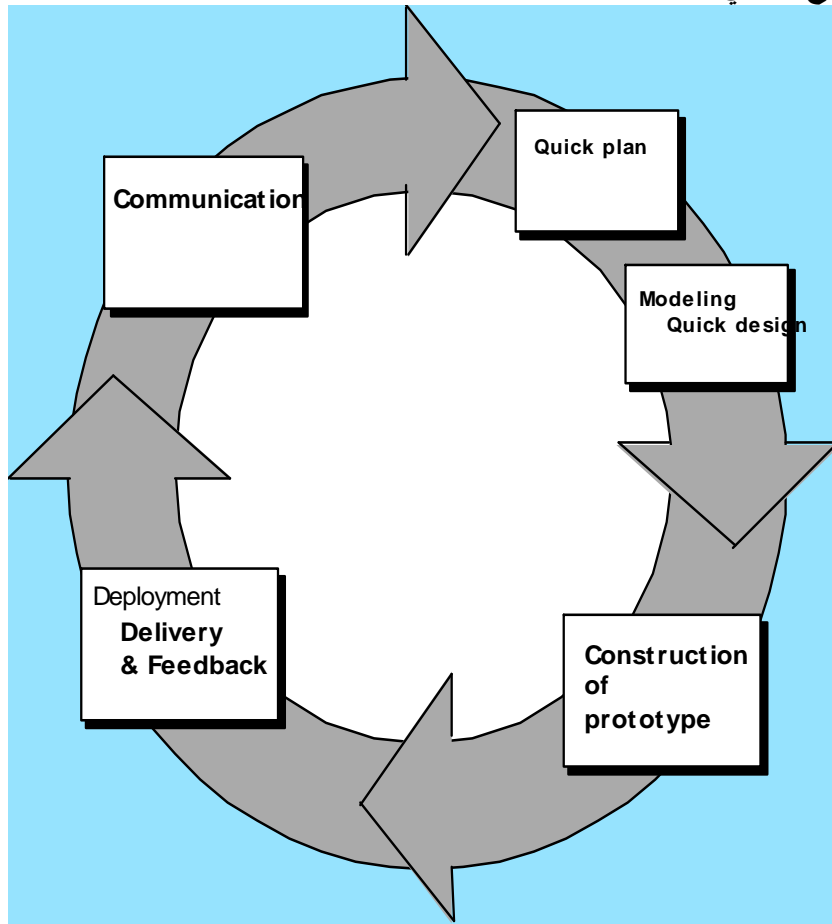
## Evolutionary development

التطوير التطوري ( في حالة عدم وضوح متطلبات النظام )

- We want the software to “evolve” to its final form “requirements are not completely known, or we need to verify them with the customers.  
• نريد للبرمجيات أن "تتطور" إلى شكلها النهائي "متطلبات لم تعرف بالكامل، أو نحن نحتاج لتحقيقها مع الزبائن
- Exploratory development: The Objective is to work with customers and to evolve a final system from an initial outline specification.  
• التطوير الاستطلاعي: الهدف أن يعمل مع الزبائن ولتطوير نظام نهائي من مواصفات الخلاصة الأولية ( العمل بشكل جزئي عليه ومن ثم تبدأ عملية التطوير )
- We should start with those requirements that are well-understood and agreed upon and then add new features as proposed by the customer.  
• يجب أن نبدأ بتلك المتطلبات والتي تكون مفهومة بشكل جيد والتي تم الاتفاق عليها وبعد ذلك تضيف ميزات جديدة كما هي مقترحة من قبل الزبون
- Throw-away prototyping: The objective is to understand the system requirements.  
• طريقة طرح النموذجي الأولي: الهدف هو أن يفهم متطلبات النظام.
- We may start with poorly understood requirements, and then build prototypes to clarify what is really needed.  
• يمكن أن نبدأ بالمتطلبات سيئة الفهم ( الغير مفهومة ) ، ومن ثم يبني النماذج لتوضيح ما الذي نحتاجه.

## Evolutionary Models: Prototyping

تصاميم التطوير : النموذج الأولي



## Evolutionary development

التطوير التطوري ( في حالة عدم وضوح متطلبات النظام )

### • Problems

• المشاكل

- Lack of process visibility ( We don't have the complete requirements and hence we are not sure how much resources we may need, or of the project will be a successful one);
  - قلة وضوح العملية (عندما لا نمتلك المتطلبات الكاملة ومن هنا لن نتأكد كم من المصادر قد نحتاج، أو أن المشروع سيكون ناجح)؛
- Systems are often poorly structured (because they evolve and change continuously).
  - الأنظمة تنظم في أغلب الأحيان بشكل سيئ ( تعمل لمدة زمنية قليلة) (لأن التطور والتغير يكون بشكل مستمر).
- Special skills (e.g. in languages for rapid prototyping) may be required.
  - المهارات الخاصة يمكن أن تكون مطلوبة (ومثال على ذلك: - في لغات النمذجة الأولي السريع).

## Evolutionary development

التطوير التطوري

### • Applicability

• التطبيق

- For small or medium-size interactive systems;
  - للأنظمة التفاعلية الصغيرة أو المتوسطة
- For parts of large systems (e.g. the user interface);
  - لأجزاء الأنظمة الكبيرة (ومثال على ذلك: - واجهة المستخدم )
- For short-lifetime systems.
  - للأنظمة مدى الحياة بها قصير
- Prototyping can be used in initial stages as a way of collecting requirements. (Customers may find it easier to make comments on actual product and not just papers or documents).
  - النموذج الأولي يمكن أن يستخدم في المراحل الأولية كالطرق لجمع المتطلبات. (الزبائن يمكن أن يجدوا أنه من السهل إبداء التعليقات على المنتج الفعلي وليس فقط على الأوراق أو الوثائق).

## Process iteration

### المعالجة المكررة

- System requirements ALWAYS evolve in the course of a project. As a result, process iteration where earlier stages are reworked is always part of large systems' processes. (MS Windows & Office continuously have updates, newer versions, batches and bug fixes).
- متطلبات النظام دائماً تتطوّر أثناء عمل المشروع. كنتيجة، تكرر المعالجة حيث أنّ المراحل السابقة مجدّدة دائماً كجزء من عمليات الأنظمة الكبيرة. (مايكروسوفت ويندوز و البرامج المكتبية لهما التجديدات بشكل مستمر، نسخ أحدث، إضافات ومعالجة الأخطاء).
- Iteration can be applied to any of the generic process models.
- التكرار يمكن أن يقدم إلى أي من تصاميم ( نماذج ) المعالجة العامة
- Two (related) approaches
  - نظرتان ( ذات علاقة )
    - Incremental delivery;
    - التسليم التزايدي
    - Spiral development.
    - التطوير الحلزوني ( اللولبي )

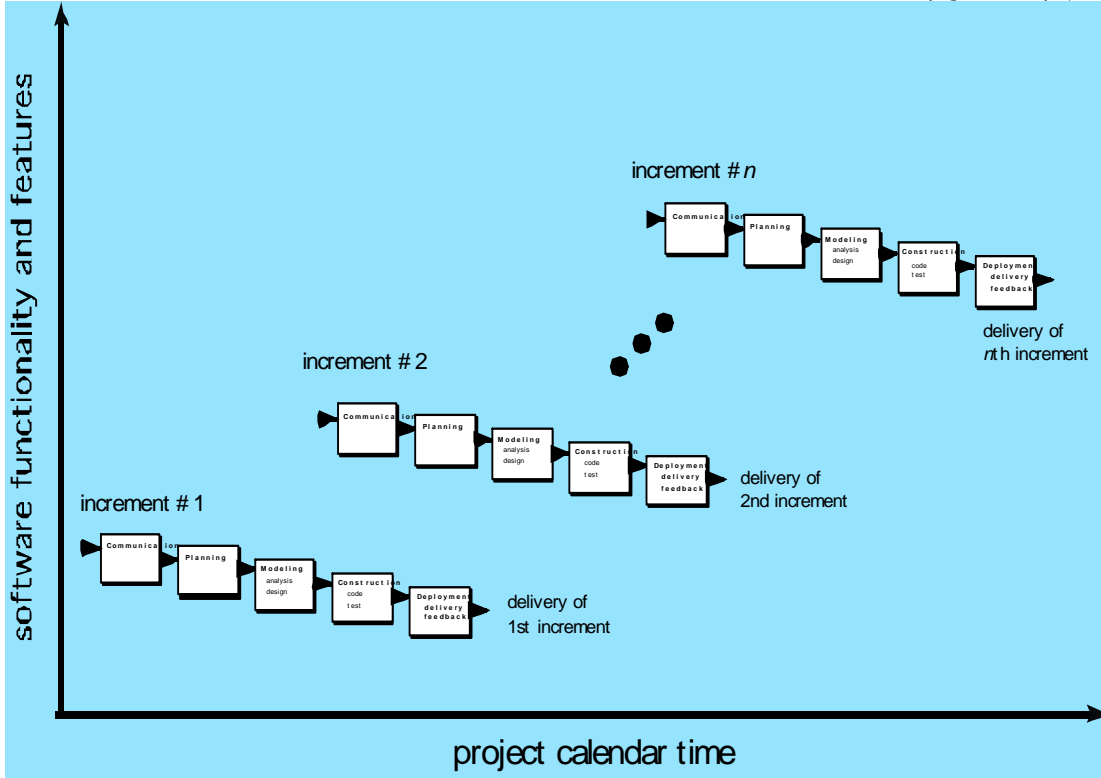
## Incremental delivery

### التسليم التزايدي ( المتزايد )

- Rather than delivering the system as a single delivery, the development and delivery is broken down into increments with each increment delivering part of the required functionality.
- بدلاً من تسليم النظام كتسليم وحيد، التطوير والتسليم الغير مكتمل إلى التزايد ستكون بكل زيادة تسلم جزء من الوظيفة المطلوبة
- User requirements are prioritised and the highest priority requirements are included in early increments.
- الأفضلية لمتطلبات المستخدم والمتطلبات الأولوية الأعلى متضمنة في الزيادات المبكرة
- Once the development of an increment is started, the requirements are frozen though requirements for later increments can continue to evolve.
- عندما يتم البدء بزيادة التطوير ، المتطلبات ستكون مجمّدة مع تلك المتطلبات للزيادات المتأخرة ويمكن أن تواصل التطوّر

## The Incremental Model

التصميم (النموذج) التزايدي



## Incremental development advantages

حسنت التطوير التزايدي

- Customer value can be delivered with each increment so system functionality is available earlier.
  - أهمية الزبون يمكن أن تسلم بكل زيادة لذا وظيفة النظام متوفرة في وقت سابق
- Early increments act as a prototype to help elicit requirements for later increments.
  - تعمل الزيادات المبكرة كنموذج أولي للمساعدة على انتزاع المتطلبات للزيادات المتأخرة
- Lower risk of overall project failure.
  - تقليل المخاطر لفشل المشروع العام
- The highest priority system services tend to receive the most testing.
  - الأولوية الأعلى لخدمات النظام تميل إلى الاستلام الأكثر للاختبار

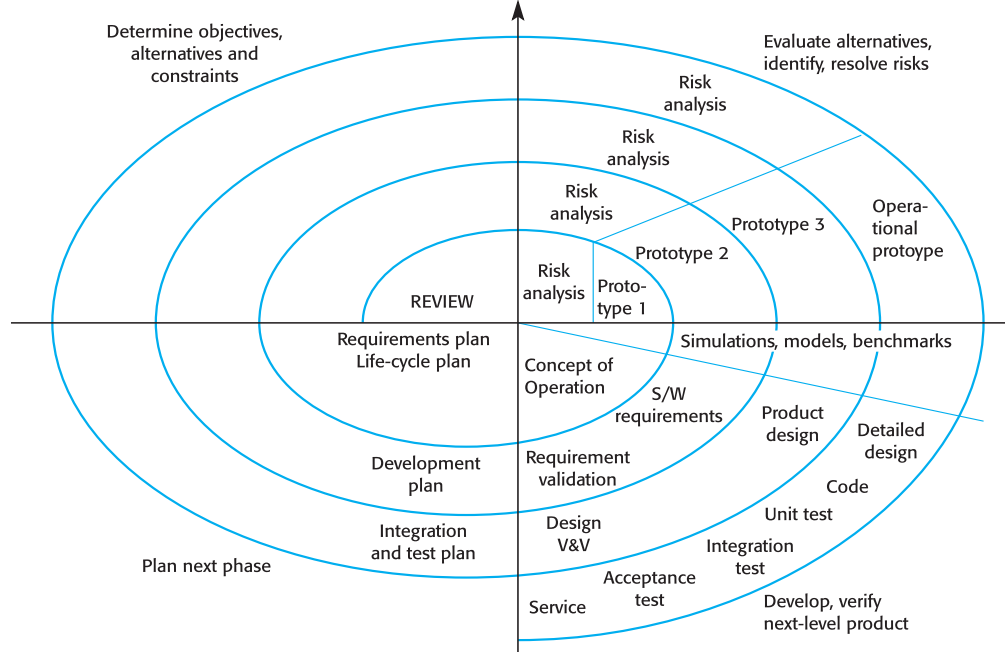
## Spiral development

### التطوير الحلزوني ( اللولبي )

- Process is represented as a spiral rather than as a sequence of activities with backtracking.
  - المعالجة ممثلة كلولب بدلا من سلسلة النشاطات مع عمليات المتابعة العكسية (التراجع)
- Each loop in the spiral represents a phase in the process.
  - كل حلقة في اللولب تمثل مرحلة في المعالجة
- No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required.
  - لا مراحل ثابتة مثل المواصفات أو التصميم - الحلقات في اللولب هو اختيار معتمد على ما هو مطلوب
- Risks are explicitly assessed and resolved throughout the process.  
(This is the main difference from the Incremental model).
  - الأخطار تقيّم بشكل واضح ويتم حلها في كافة أنحاء المعالجة. (هذا الاختلاف الرئيسي من النموذج التزايدى ( المتزايد )).

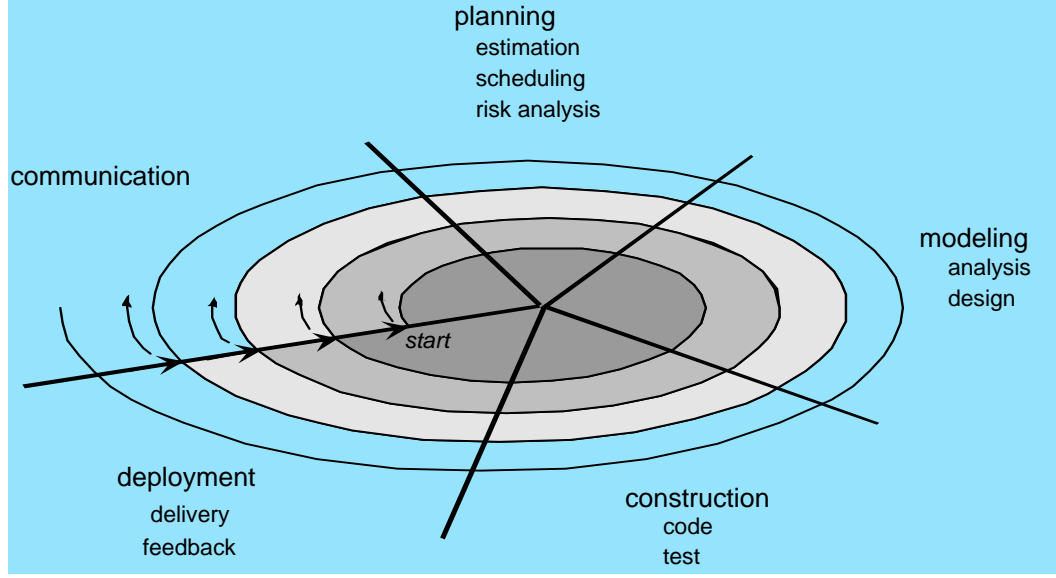
## Spiral model of the software process

### التصميم ( النموذج ) الحلزوني ( اللولبي ) لمعالجة البرمجيات



## Evolutionary Models: The Spiral

تصاميم التطويري : الحلزوني ( اللولبي )



### Spiral model sectors

قطاعات التصميم اللولبي ( الحلزوني )

- Objective setting
  - موضوعية المكان
  - Specific objectives for the phase are identified.
  - الأهداف المعينة للمرحلة تكون معرفة
- Risk assessment and reduction
  - تقييم المخاطر وتخفيضها
  - Risks are assessed and activities put in place to reduce the key risks.
  - المخاطر تم تقييمها والنشاطات توضع في المكان لتقليل هذه المخاطر
- Development and validation
  - التطوير والتحقق
  - A development model for the system is chosen which can be any of the generic models.
  - نموذج التطوير للنظام يتم اختياره عندما يكون أي من النماذج العامة
- Planning
  - التخطيط
  - The project is reviewed and the next phase of the spiral is planned.
  - المشروع يتم مراجعته والمرحلة القادمة للولب مخطط لها.



## General process activities

- Software specification
- Software design and implementation
- Software validation
- Software evolution

## نشاطات المعالجة العامة

- تخصيص البرمجيات
- تصميم وتطبيق البرمجيات
- التحقق من صحة البرمجيات
- تطوير البرمجيات

## Software specification

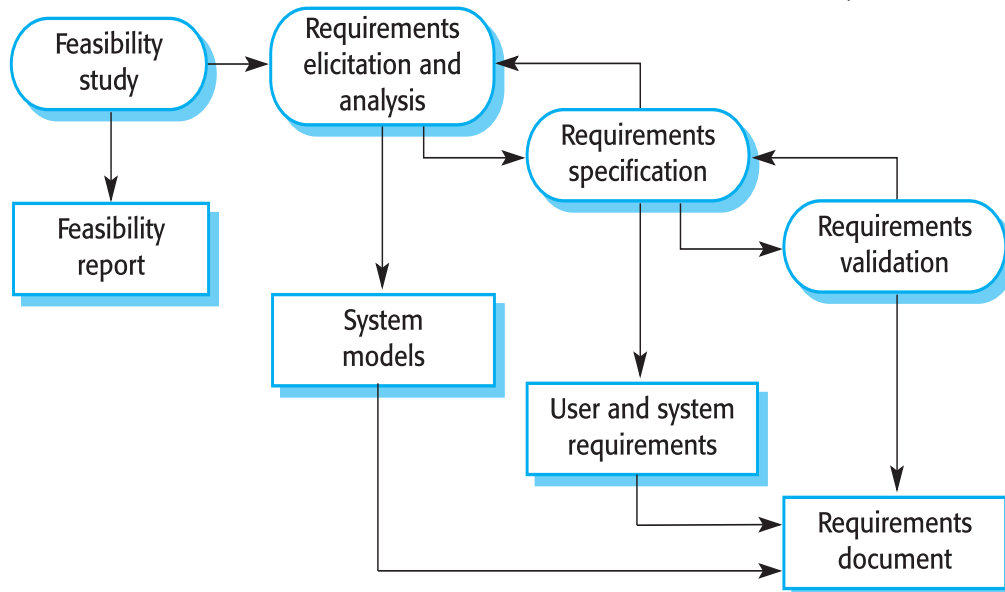
- The process of establishing what services are required and the constraints on the system's operation and development.
  - تأسيس المعالجة لمعرفة ماهية متطلبات الخدمات والقيود على عمليات وتطوير النظام
- Requirements engineering process
  - متطلبات هندسة المعالجة
  - Feasibility study;
    - دراسة الجدوى
  - Requirements elicitation and analysis;
    - متطلبات الأحداث والتحليل
  - Requirements specification;
    - متطلبات التخصيص
  - Requirements validation.
    - متطلبات التحقق من الصحة

## تخصيص البرمجيات

## متطلبات هندسة المعالجة

## The requirements engineering process

## متطلبات هندسة المعالجة



## Software design and implementation

### تصميم وتطبيق البرمجيات

- Software design: To design a software structure or model that realises the specification; The design is the link between the requirements and the code (The blueprint such as the one used in buildings).
- تصميم البرمجيات: لتصميم تركيب البرمجيات أو نموذج إدراك المواصفات؛ التصميم يكون حلقة الوصل بين المتطلبات والشيفرة ( الكود البرمجي ) (المخطط مثل المستخدم في البناء).
- Implementation: The process of converting the system specification into an executable system (i.e. the code or the software).
- التطبيق: المعالجة من خلال تحويل مواصفات النظام إلى نظام قابل للتنفيذ (وبمعنى آخر: الشيفرة ( الكود البرمجي ) أو البرمجيات).
- The activities of design and implementation are closely related and may be inter-leaved.
- نشاطات التصميم والتطبيق بينهما ترابط شديد وقد يكونان بين بعضهما.

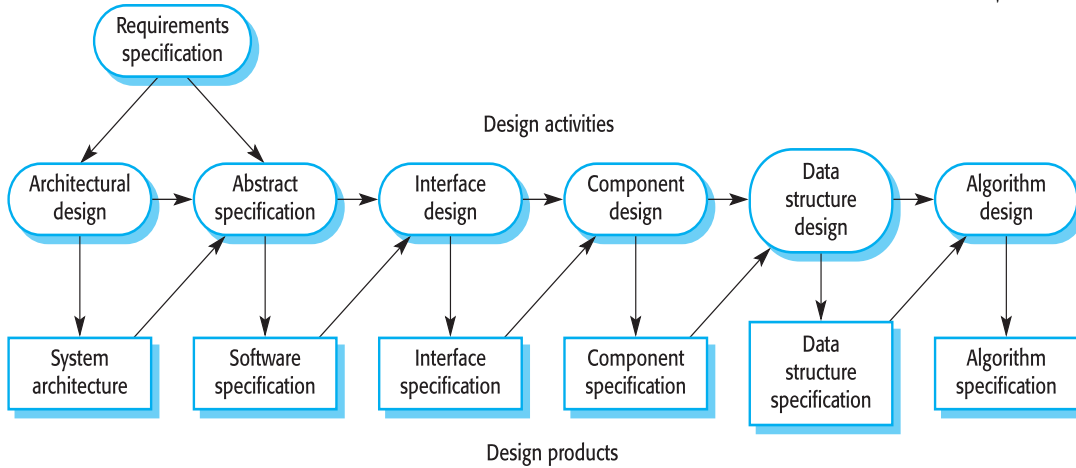
## Design process activities

### نشاطات معالجة التصميم

- Architectural design
- تصميم المعمارية
- Abstract specification
- التخصيص التجريدي
- Interface design
- تصميم الواجهة
- Component or detail design
- تصميم المكونات أو التفاصيل
- Data structure design
- تصميم تركيب البيانات
- Algorithm design
- تصميم الخوارزمية

## The software design process

معالجة تصميم البرمجيات



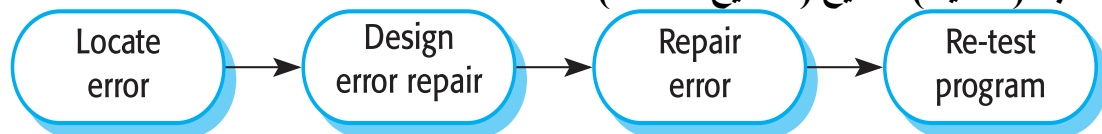
## Programming and debugging

البرمجة وتصحيح الأخطاء

- Translating a design into a program and removing errors from that program.
  - ترجمة التصميم إلى البرنامج وحذف الأخطاء من هذا البرنامج
- We use programming or developing to refer to the implementation stage.
  - نستخدم البرمجة أو التطوير لنشير إلى مرحلة التطبيق
- Programming is a personal activity - there is no generic programming process, but there are coding standards (such as naming conventions).
  - البرمجة هي نشاط الأشخاص - ليس هناك معالجة برمجة عامة، لكن هناك شيفرة معيارية (قياسية) (مثل الاتفاق على التسميات)
- Programmers carry out some program testing to discover faults in the program and remove these faults in the debugging process (i.e. unit testing).
  - ينفذ المبرمجون البعض من برامج الاختبار لاكتشاف العيوب (الأخطاء) في البرنامج ويعمل على إزالة هذه العيوب (الأخطاء) في عملية التنقيح (تصحيح الأخطاء) (وبمعنى آخر وحدة الاختبار).

## The debugging process

معالجة ( عملية ) التنقيح ( تصحيح الأخطاء )



## Software validation

### التحقق من صحة البرمجيات

- Verification and validation (V & V) are intended to show that a system conforms to its specification and meets the requirements of the system customer.
- التحقق من الصحة والتصديق (V & V) المقصود رؤية النظام يتوافق مع مواصفاته ويقابل متطلبات زبون النظام.
- Involves checking and review processes and system testing.
- تتضمن عمليات المراجعة والتدقيق واختبار النظام
- System testing involves executing the system with test cases that are derived from the specification in order to be processed by the system.
- اختبار النظام يتضمن تنفيذ النظام بوضعية الفحص التي تكون مشتقة من المواصفات لكي تكون مصنعة بالنظام
- Validation and verification occurs in the requirement stage while testing occurs within or after the implementation stage.
- يحدث التصديق والتحقق في مرحلة المتطلبات عندما يحدث الاختبار ضمن أو بعد مرحلة التطبيق.

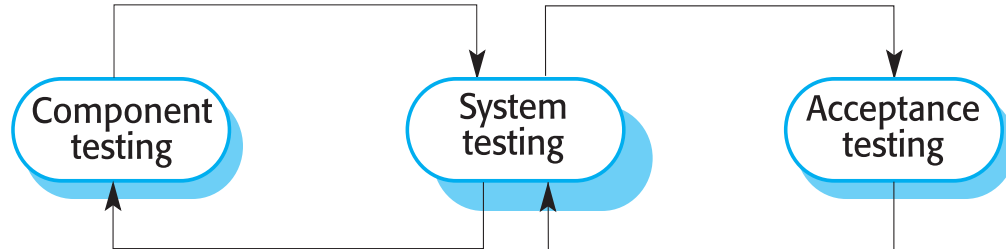
## Testing stages

### مراحل الفحص

- Component or unit testing : Individual components are tested independently;
  - المكونات أو وحدة الفحص: المكونات الفردية يتم اختبارها ( تجربتها ) بشكل مستقل
  - Components may be functions or objects or coherent groupings of these entities. Usually done by developers.
  - المكونات يمكن أن تكون وظائف أو أهداف أو تجمعات متماسكة لهذه الكيانات. عادةً يتم إنجازها من خلال المطورين
- System testing: Testing of the system as a whole. Testing of emergent properties is particularly important. Usually done by local company testers.
  - فحص النظام: اختبار النظام ككل. اختبار الخصائص المنبثقة المهمة جداً. عادةً يتم إنجازها بفاحصات الشركة المحلية.
- Acceptance testing: Testing with customer data to check that the system meets the customer's needs. Usually done by customers or independent testers.
  - اختبار القبول: اختبار ببيانات الزبون للتدقيق بأن هذا النظام يقابل حاجات الزبون. عادةً يتم إنجازها من قبل الزبائن أو الفاحصات المستقلة.

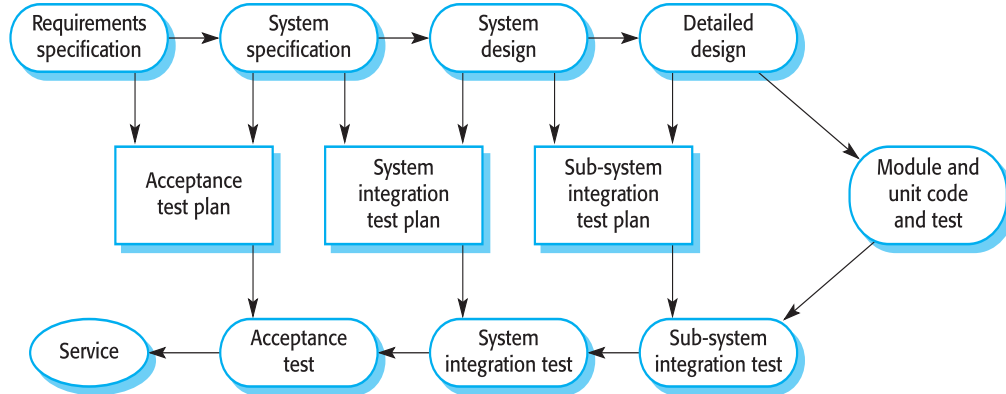
## The testing process

### عملية الفحص



## Testing phases

### مراحل الفحص



## Other types of testing

### أنواع أخرى من الفحوصات

- Unit or white box testing (i.e. component testing).  
وحدة أو اختبار الصندوق الأبيض (وبمعنى آخر: . الاختبار للمكونات).
- Black box testing (can be part of system testing).  
اختبار الصندوق الأسود (يمكن أن يكون جزء من اختبار النظام).
- Integration testing (related to system testing).  
الاختبار (الفحص) المتكامل (مرتبط مع اختبار النظام).
- User Interface testing (can be part of system testing).  
اختبار (فحص) واجهة المستخدم (يمكن أن يكون جزء من اختبار النظام).
- Alpha and Beta testing (can be part of acceptance testing).  
اختبار ألفا وبيتا (يمكن أن يكون جزء من اختبار القبول).

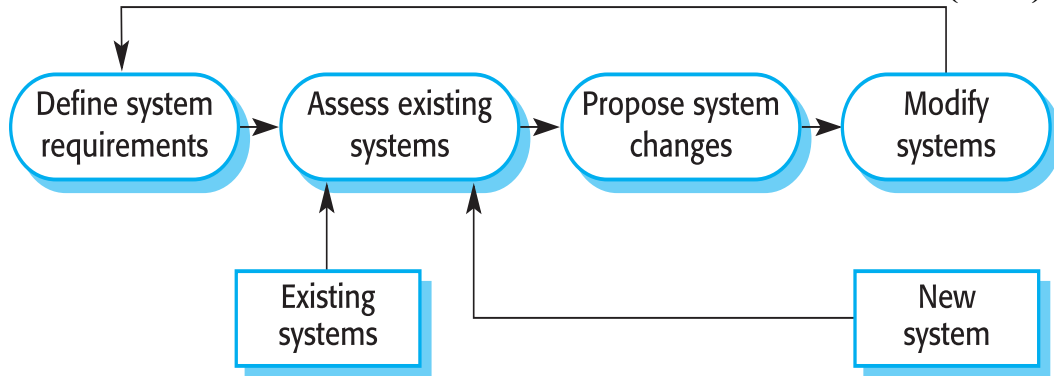
## Software evolution or maintenance

### تطوير أو صيانة البرمجيات

- Software is inherently flexible and can change.  
البرمجيات أصلها مرن ويمكن أن تتغير.
- As requirements change, through changing business circumstances, the software that supports the business must also evolve and change.  
تغير المتطلبات، خلال تغيير ظروف العمل (الشركة)، البرمجيات التي تدعم العمل (الشركة) يجب أيضاً أن تتطور وتتغير.
- Although there has been a mixing between development and evolution (maintenance) this is increasingly irrelevant as fewer and fewer systems are completely new.  
بالرغم من أن هناك خلط بين التطوير والتطور (الصيانة) هذه ليست علاقة بين الأنظمة المطورة بفعل الصيانة (من نظام موجود مسبقاً) والأنظمة المطورة بشكل جديد.

## System evolution

### تطور (صيانة) البرمجيات

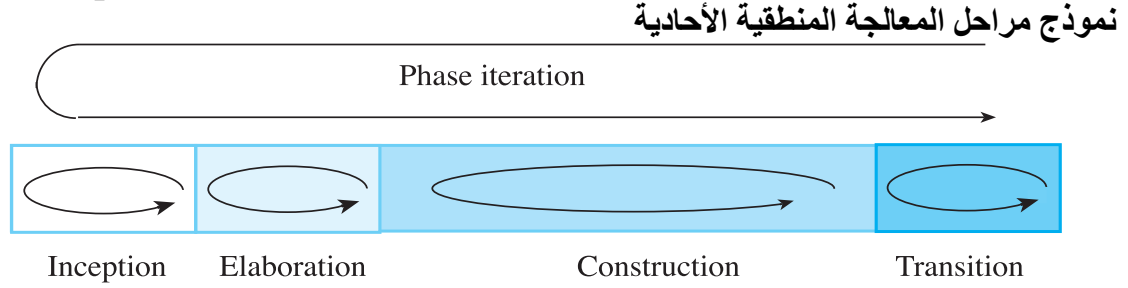


## The Rational Unified Process

### المعالجة المنطقية الموحدة

- A modern process model derived from the work on the UML and associated process.
  - نموذج المعالجة الحديثة مشتق من العمل على UML والمعالجة التشاركية
- Normally described from 3 perspectives
  - وصفت عادة من ثلاثة منظورات
  - A dynamic perspective that shows phases over time;
    - المنظور المتحرك تشاهد من خلالها المراحل مع مرور الوقت
  - A static perspective that shows process activities;
    - المنظور الساكن تشاهد من خلالها نشاطات المعالجة
  - A proactive perspective that suggests good practices.
    - المنظور التفاعلي الذي يقترح من خلاله الممارسات الجيدة

## RUP phase model



## RUP phases

### مراحل المعالجة المنطقية الأحادية

- Inception
  - الإبتداء
  - Establish the business case for the system.
    - تأسيس وضعية الشركة من النظام
- Elaboration
  - الدراسة
  - Develop an understanding of the problem domain and the system architecture.
    - تطوير الفهم لمجال المشكلة وهيكلية النظام
- Construction
  - البناء
  - System design, programming and testing.
    - تصميم النظام، البرمجة والفحص ( الاختبار )
- Transition
  - الانتقال (التحول)
  - Deploy the system in its operating environment.
    - نشر النظام في العمليات البيئية

## RUP good practice

### الممارسات الجيدة للمعالجة الأحادية العلائقية

- Develop software iteratively
  - تطوير البرمجيات بشكل تكراري ( روتيني )
- Manage requirements
  - إدارة المتطلبات
- Use component-based architectures
  - استخدام المكونات المعتمدة على المعمارية
- Visually model software
  - نموذج البرمجيات يكون بشكل مرئي
- Verify software quality
  - تحقق نوعية البرمجيات
- Control changes to software
  - تغيير السيطرة على البرمجيات



## Static workflows

## تدفق العمل الساكن

Workflow تدفق العمل	Description الوصف
Business modeling وضع الشركة	The business processes are modelled using business use cases. عمليات الشركة تستخدم نموذج العمل عند حدوث مشكلة
Requirements المتطلبات	Actors who interact with the system are identified and use cases are developed to model the system requirements. الممثلون الذين يتفاعلون بالنظام يحددون ويستخدمون المشكلات التي تطور لتشكيل متطلبات النظام
Analysis and design التحليل والتصميم	A design model is created and documented using architectural models, component models, object models and sequence models. نموذج التصميم ينشأ ويوثق استخدام النماذج المعمارية، نماذج المكونات ونماذج الأهداف ونماذج المتسلسلة
Implementation التطبيق	The components in the system are implemented and structured into implementation sub-systems. Automatic code generation from design models helps accelerate this process. المكونات في النظام تطبق وتنظم إلى تطبيق الأنظمة الفرعية. جيل الشيفرة الآلية من نماذج التصميم تساعد على تسريع هذه العملية.
Test الفحص (الاختبار)	Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation. الاختبار عملية (معالجة) تكرارية تنفذ من خلال الارتباط مع التطبيق. اختبار النظام يتبعه اكتمال التطبيق
Deployment الانتشار	A product release is created, distributed to users and installed in their workplace. تعني إطلاق المنتج الذي تم إنشائه، ومن ثم توزيعه إلى المستخدمين وتركيبه في موقع عملهم
Configuration and change management إدارة التغيير والترتيب	This supporting workflow managed changes to the system (see Chapter 29). هذا يدعم إدارة تغيير تدفق العمل للنظام (أنظر الفصل التاسع والعشرون)
Project management إدارة المشروع	This supporting workflow manages the system development (see Chapter 5). هذا يدعم إدارة تدفق العمل لتطوير النظام (أنظر الفصل الخامس)
Environment البيئة	This workflow is concerned with making appropriate software tools available to the software development team. هذه تهتم بتدفق العمل مع صنع (تقديم) أدوات البرمجيات المتوفرة الملائمة إلى فريق تطوير البرمجيات.

## Computer-aided software engineering CASE

### هندسة البرمجيات المعتمدة على الحاسوب

- Computer-aided software engineering (CASE) is a software to support software development and evolution processes (Example, .NET IDE).
- هندسة البرمجيات المعتمدة على الحاسوب هي برمجيات لدعم تطوير البرمجيات وتطور العمليات ( المعالجة ) (ومثال عليها .NET IDE).
- Activity automation
  - أنتمة النشاط
  - Graphical editors for system model development (such as UML, .NET, Eclipse, etc);
    - المحرر الرسومي (التخطيطي) لتطوير تصميم النظام ( مثل .NET, UML, Eclipse)
  - Data dictionary to manage design entities (OO design tools);
    - قاموس البيانات لإدارة كيانات التصميم (أدوات تصميم الكائنات الموجهة)؛
  - Graphical UI builder for user interface construction (.NET, Eclipse, etc.);
    - واجهة المستخدم الرسومية تم بنائها لبناء واجهة المستخدم (.NET, Eclipse)
  - Debuggers to support program fault finding;
    - التنقيح ( تصحيح الأخطاء) لدعم إيجاد أخطاء البرنامج
  - Automated translators to generate new versions of a program (such as Installers).
    - المترجمات الآلية لتوليد نسخ جديدة من البرنامج (مثل عملية التركيب).

## CASE technology

### تقنيات هندسة البرمجيات المعتمدة على الحاسوب

- CASE technology has led to significant improvements in the software process. However, these are not the order of magnitude improvements that were once predicted
  - تقنيات هندسة البرمجيات المعتمدة على الحاسوب أدت إلى التحسينات الهامة في معالجة البرمجيات. على أية حال، هذه ليست طلب تحسينات ذو أهمية التي تكون متوقعة
- Software engineering requires creative thought - this is not readily automated;
  - هندسة البرمجيات تتطلب فكر مبدع - هذا لا يتم أنتمته بسهولة
- Software engineering is a team activity and, for large projects, much time is spent in team interactions. CASE technology does not really support these. (i.e. the personal activities, Some tools can be used to work as a communication media between project team members, project or defect tracking tools).
- هندسة البرمجيات هو نشاط فريق، وللمشاريع الكبيرة، ووقت كثير يصرف في تفاعلات الفريق. تقنية هندسة البرمجيات المعتمدة على الحاسوب لا تدعم هذه حقا. (وبمعنى آخر: النشاطات الشخصية، بعض الأدوات يمكن أن تستخدم للعمل كوسائط اتصال المشروع بين أعضاء فريق المشروع أو الخلل الذي يتعقب الأدوات).

## CASE classification

### تصنيف هندسة البرمجيات المعتمدة على الحاسوب

- Classification helps us understand the different types of CASE tools and their support for process activities.
  - التصنيف يساعد على فهم الفرق بين أنواع أدوات هندسة البرمجيات المعتمدة على الحاسوب والتي تدعم نشاطات المعالجة
- Functional perspective
  - المنظور الوظيفي
  - Tools are classified according to their specific function.
    - الأدوات تصنف طبقا لوظائفها المخصصة (المعينة)
- Process perspective
  - منظور المعالجة (العملية)
  - Tools are classified according to process activities that are supported.
    - الأدوات تصنف طبقا لدعم نشاطات المعالجة
- Integration perspective
  - المنظور التكاملي
  - Tools are classified according to their organisation into integrated units.
    - الأدوات تصنف طبقا للمنظمة لتكامل وحداتها

## Functional tool classification

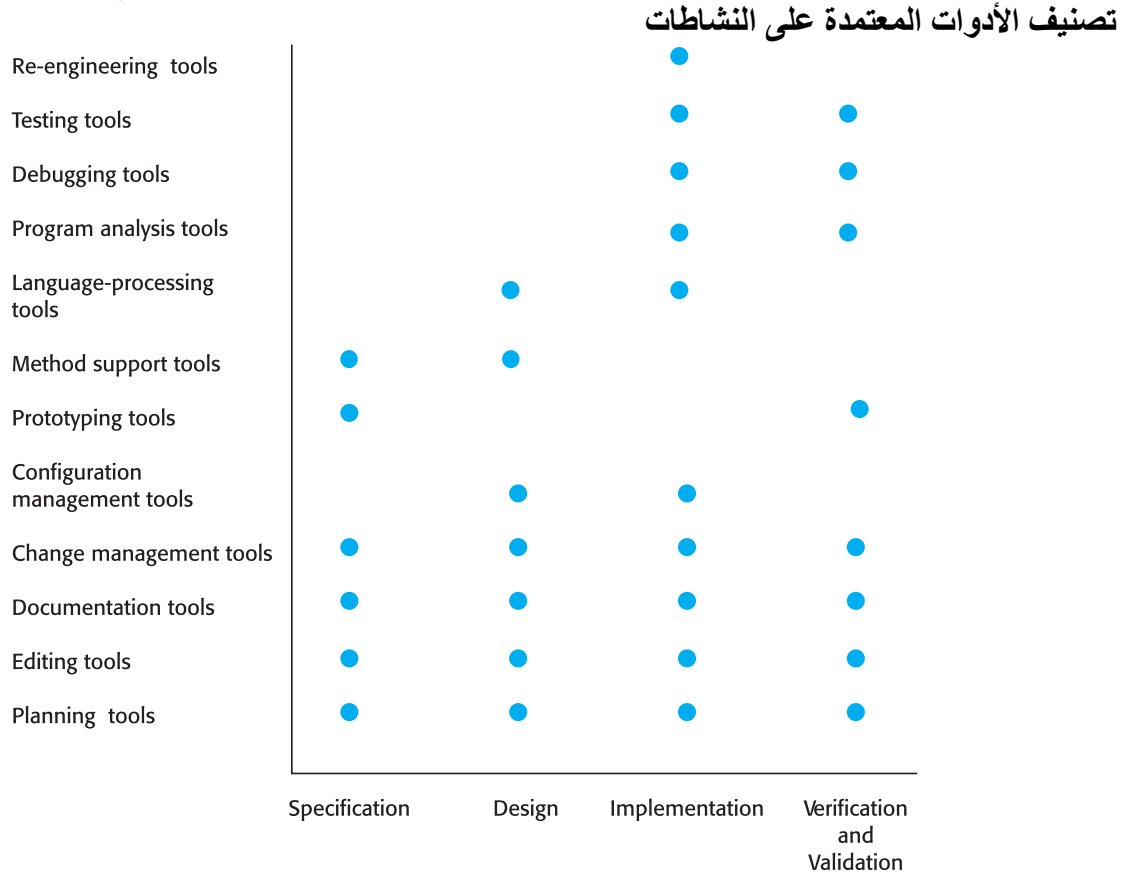
## تصنيف الأدوات حسب الوظيفة

---

<b>Tool type</b> نوع الأداة	<b>Examples</b> أمثلة
Planning tools أدوات التخطيط	PERT tools, estimation tools, spreadsheets
Editing tools أدوات التحرير	Text editors, diagram editors, word processors
Change management tools أدوات إدارة التغيير	Requirements traceability tools, change control systems
Configuration management tools أدوات إدارة الترتيب	Version management systems, system building tools
Prototyping tools أدوات النماذج الأولية	Very high-level languages, user interface generators
Method-support tools أدوات دعم - الطرق	Design editors, data dictionaries, code generators
Language-processing tools أدوات معالج - اللغات	Compilers, interpreters
Program analysis tools أدوات تحليل البرنامج	Cross reference generators, static analysers, dynamic analysers
Testing tools أدوات الاختبار ( الفحص )	Test data generators, file comparators
Debugging tools أدوات تصحيح الأخطاء	Interactive debugging systems
Documentation tools أدوات التوثيق	Page layout programs, image editors
Re-engineering tools أدوات إعادة الهندسة	Cross-reference systems, program re- structuring systems

---

## Activity-based tool classification



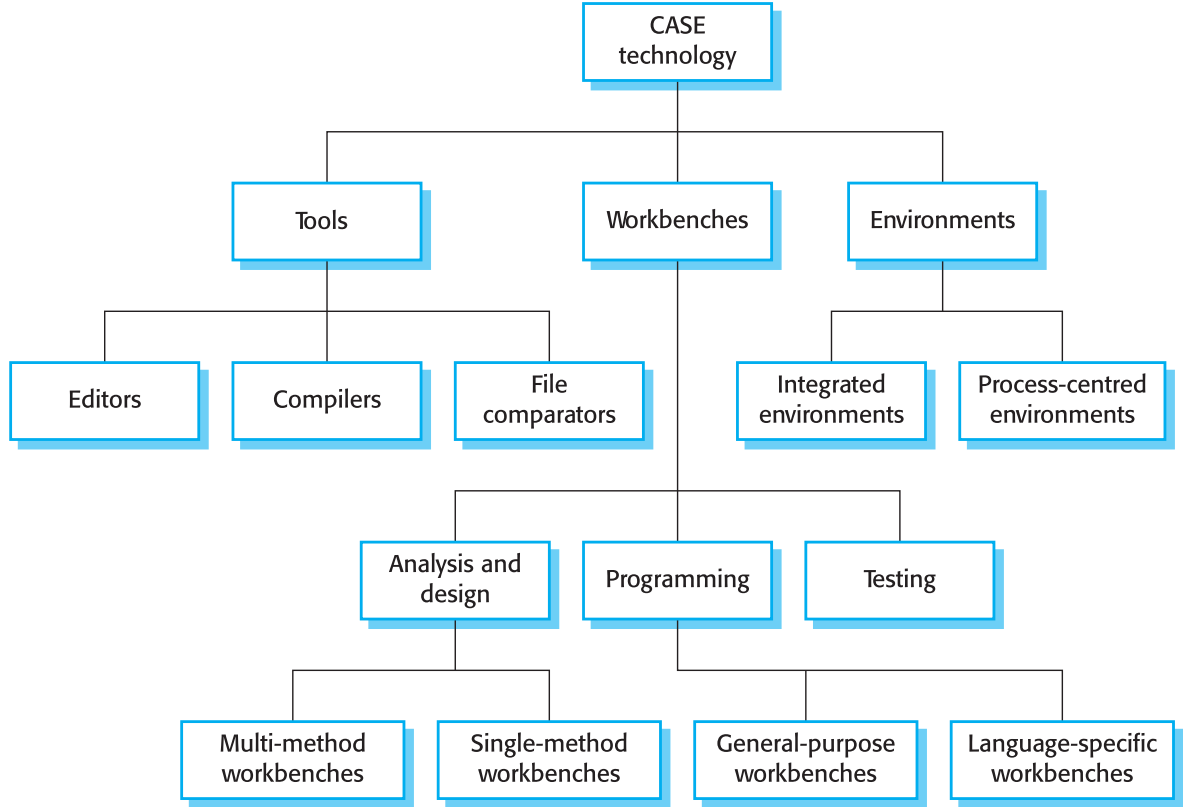
## CASE integration

تكامل هندسة البرمجيات المعتمدة على الحاسوب

- Tools
  - الأدوات
  - Support individual process tasks such as design consistency checking, text editing, etc.
  - تدعم مهام المعالجة الفردية مثل تدقيق توافق التصميم، تحرير النص، الخ.
- Workbenches
  - مكان العمل
  - Support a process phase such as specification or design, Normally include a number of integrated tools.
  - تدعم مرحلة المعالجة مثل التخصيص أو التصميم، عادة يتضمن عدد من الأدوات المتكاملة
- Environments
  - البيئات
  - Support all or a substantial part of an entire software process. Normally include several integrated workbenches.
  - يدعم كل أو الجزء الأكبر من عملية (معالجة) البرمجيات. عادة يتضمن عدة أماكن عمل متكاملة

## Tools, workbenches, environments

الأدوات، أماكن العمل، البيئات.



## The Manifesto for Agile Software Development

البيان العام لتطوير البرمجيات السريع

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

”نحن نكشف طرق أفضل من تطوير البرمجيات بعملها ومساعدة الآخرين بالعمل. خلال هذا العمل جننا للتقييم:

- *Individuals and interactions* over processes and tools  
• الأفراد والتفاعلات على العمليات والأدوات
- *Working software* over comprehensive documentation  
• عمل البرمجيات على التوثيق الشامل
- *Customer collaboration* over contract negotiation  
• تعاون الزبائن على مفاوضات العقد
- *Responding to change* over following a plan  
• الاستجابة للتغيير على الخطط المتبعة

That is, while there is value in the items on the right, we value the items on the left more.”

ذلك، بينما هناك قيمة في المواد على الجهة اليمنى، نقيم المواد على الجهة اليسرى أكثر.”

## What is "Agility"?

ماذا تعني " خفة الحركة، الرشاقة " ؟

- Effective (rapid and adaptive) response to change  
الفعالية (السرعة والتكيف) يستجيب للتغيير
- Effective communication among all stakeholders  
الاتصال الفعال بين كل أصحاب الحصص (المالكين)
- Drawing the customer onto the team  
يسحب (يجعل) الزبون في الفريق
- Organizing a team so that it is in control of the work performed  
Yielding...  
تنظيم الفريق لكي يكون هو المسيطر على تأدية العمل، الناتج ...
- Rapid, incremental delivery of software  
سريع، التسليم التزايدى للبرمجيات

## An Agile Process

المعالجة الخفيفة الحركة، الرشاقة

- Is driven by customer descriptions of what is required (scenarios)  
تتم القيادة من خلال وصف متطلبات الزبون (السيناريوهات)
- Recognizes that plans are short-lived  
يعرف الخطط القصيرة الأجل
- Develops software iteratively with a heavy emphasis on construction activities  
تطوير البرمجيات الروتيني (التكراري) مع التأكيد على نشاطات البناء
- Delivers multiple 'software increments'  
التسليم المتعدد " تزايد البرمجيات"
- Adapts as changes occur  
التكيف مع التغيرات التي تحدث.

## Extreme Programming (XP)

### البرمجة النهائية

- The most widely used agile process, originally proposed by Kent Beck العملية (المعالجة) السريعة المستخدمة على نحو واسع، اقترحت بالأصل من خلال كنت بك
- XP Planning

### مخطط البرمجة النهائية

- Begins with the creation of “user stories” يبدأ مع إنشاء "قصص المستخدمين"
- Agile team assesses each story and assigns a cost الفريق السريع يقيم كل قصة ويخصص لها التكلفة
- Stories are grouped to for a deliverable increment القصص يتم جمعها للتسليم القابلة للزيادة
- A commitment is made on delivery date التعهد عند تاريخ التسليم
- After the first increment “project velocity” is used to help define subsequent delivery dates for other increments بعد الزيادة الأولى "سرعة المشروع" تستخدم للمساعدة على تعريف تاريخ التسليم اللاحق للزيادات الأخرى

- XP Design

### تصميم البرمجة النهائية

- Follows the KIS principle اتبع مبادئ KIS
- Encourage the use of CRC cards. يساعد (يشجع) على استخدام بطاقات CRC
- For difficult design problems, suggests the creation of “spike solutions”—a design prototype لمشاكل التصميم الصعبة، يقترح إنشاء "حلول المسمار" – تصميم النموذج الأولي.
- Encourages “refactoring”—an iterative refinement of the internal program design يساعد (يشجع) "إعادة التحليل العوامل" – للنقاء التكراري للتصميم الداخلي للبرنامج

- XP Coding

### شيفرة البرمجة النهائية

- Recommends the construction of a unit test for a store *before* coding commences (Test driven development). يوصي على بناء وحدة اختبار للتخزين قبل البدء في التشفير (تطوير قيادة الاختبار).
- Encourages “pair programming” يساعد (يشجع) " البرمجة الزوجية"



- XP Testing

- فحص البرمجة النهائية

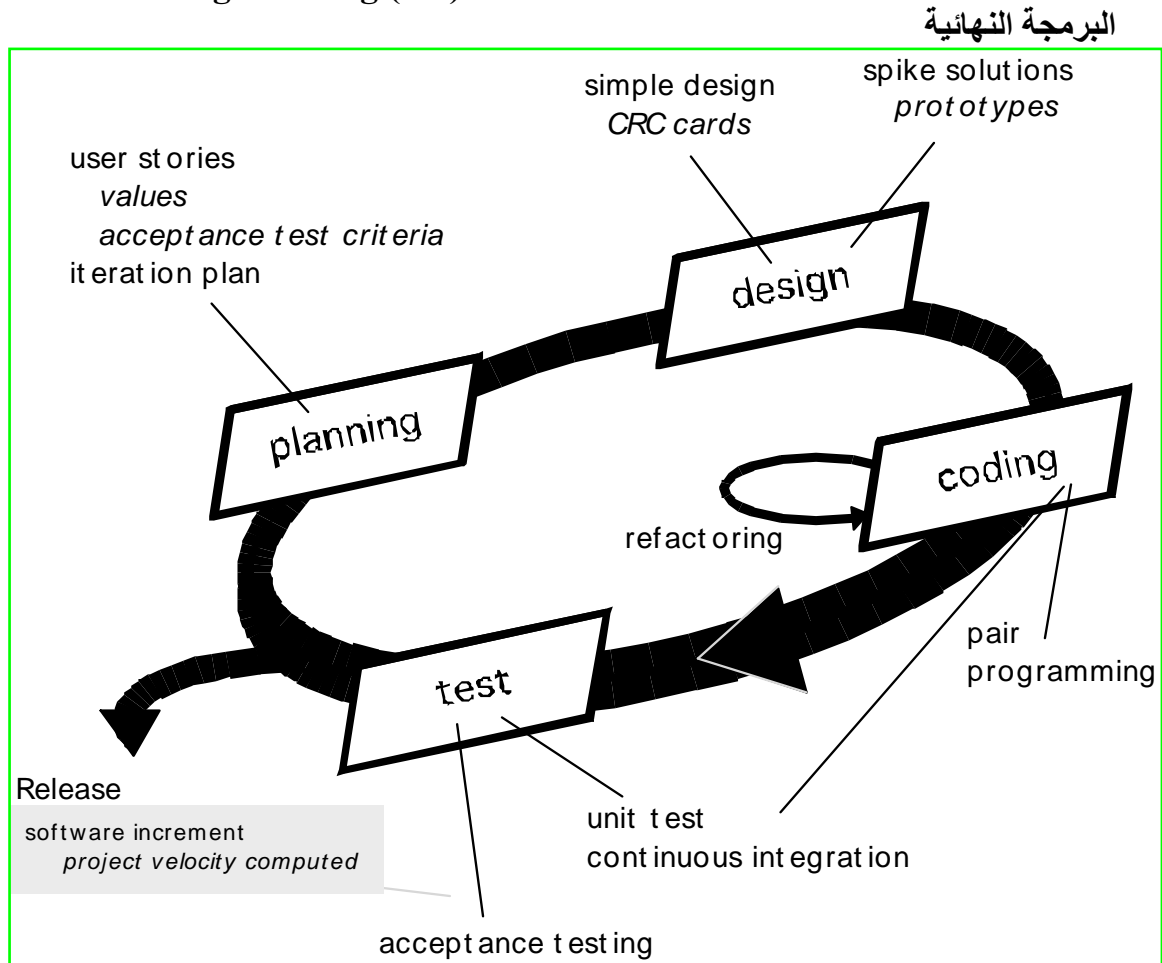
- All unit tests are executed daily

- كل وحدات الفحص تنفذ يوميا

- “Acceptance tests” are defined by the customer and executed to assess customer visible functionality

- "اختبارات القبول" معرفة من قبل الزبون ونفذت لتقييم وظيفة الزبون المرئية

## Extreme Programming (XP)



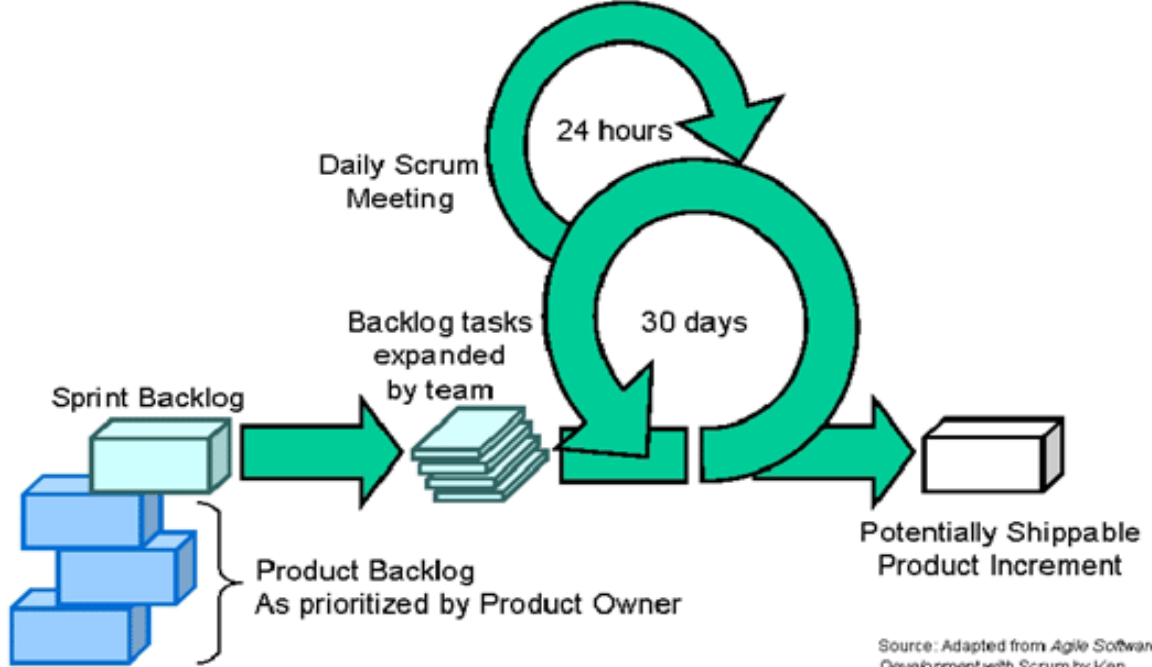
## Scrum

### المزامنة (التزام)

- Originally proposed by Schwaber and Beedle  
• اقترحت بالأصل من خلال سكوبير وبيدل
- Scrum—distinguishing features  
• الازدحام – الميزات التي تميزه
- Development work is partitioned into “packets”  
• عمل التطوير مقسم إلى "رزم"
- Testing and documentation are on-going as the product is constructed  
• الاختبار والتوثيق مستمر كبناء المنتج
- Work occurs in “sprints” and is derived from a “backlog” of existing requirements  
• أحداث العمل في "الإسراع" ومشتق من "تراكم" المتطلبات الموجودة
- Meetings are very short and sometimes conducted without chairs  
• الاجتماعات قصيرة جداً أحياناً تجري بدون وجود كراسي
- “demos” are delivered to the customer with the time-box allocated  
• "العينات" تسلّم إلى الزبون بالصندوق المخصص لها

## Scrum

### التزام



## Crystal

## البلور

- Proposed by Cockburn and Highsmith
  - Crystal—distinguishing features
    - Actually a family of process models that allow “maneuverability” based on problem characteristics
    - Face-to-face communication is emphasized
    - Suggests the use of “reflection workshops” to review the work habits of the team
- مقترح من خلال كوكبورن و وهايسميث
- البلور – الميزات التي تميزه
- في الحقيقة عائلة نماذج المعالجة التي تسمح ل"عمليات المناورة" مستندة على خصائص المشكلة
- الاتصال وجها لوجه بشكل مؤكد
- يقترح استخدام "الورشات المنعكسة" لمراجعة عادات عمل الفريق

## Key points

### النقاط الرئيسية

- Software processes are the activities involved in producing and evolving a software system.  
• معالجة البرمجيات هي اشترك النشاطات في الإنتاج والتطور (الصيانة) لنظام البرمجيات.
- Software process models are abstract representations of these processes.  
• نماذج معالجة البرمجيات اعتراضات مجردة لهذه العمليات ( المعالجة)
- General activities are specification, design and implementation, validation and evolution.  
• النشاطات العامة مواصفات (تخصيص) وتصميم وتطبيق وتصديق وتطور
- Generic process models describe the organisation of software processes. Examples include the waterfall model, evolutionary development and agile development.  
• نماذج المعالجة العامة تصف منظمة معالجة البرمجيات. تتضمن الأمثلة نموذج الشلال، تطوير التطوري ( تطوير الصيانة) والتطوير السريع.
- Iterative process models describe the software process as a cycle of activities.  
• نماذج المعالجة التكرارية تصف معالجة البرمجيات كدورة النشاطات
- Requirements engineering is the process of developing a software specification.  
• هندسة المتطلبات عملية (معالجة) التطوير لمواصفات البرمجيات
- Design and implementation processes transform the specification to an executable program.  
• معالجة (عمليات) التطبيق والتصميم تحول المواصفات إلى برنامج قابل للتنفيذ (تنفيذي)
- Validation involves checking that the system meets to its specification and user needs.  
• التصديق يتضمن التدقيق في اجتماع النظام على المواصفات والتي يحتاجها المستخدم
- Evolution is concerned with modifying the system after it is in use.  
• التطور (الصيانة) مهتمة بتعديل النظام بعدما يكون قيد الاستخدام
- The Rational Unified Process is a generic process model that separates activities from phases.  
• المعالجة المنطقية الموحدة نموذج المعالجة العامة الذي يفصل النشاطات من المراحل
- CASE technology supports software process activities.  
• تقنية هندسة البرمجيات بمساعدة الحاسوب تدعم برمجيات معالجة النشاطات
- Agile development methodologies are suggested to deal with the continuously evolving requirements and the need of company to deliver software in a reasonable time.  
• منهجيات التطوير السريعة تقترح للتعامل مع المتطلبات الناشئة بشكل مستمر وحاجة الشركة لتسليم البرمجيات في مدة معقولة.

- Agile development accept changes on requirements and deal with them continuously. They focus on the product rather than the documents. They focus on customer satisfaction rather than contracts.
- يقبل التطوير السريع التغييرات على المتطلبات وتتعامل معهم بشكل مستمر. يركزون على المنتج بدلا من الوثائق. يركزون على إرضاء الزبون بدلا من العقود.
- The 2 factors that affect the software process is the stability of the requirements and the flexibility to change requirements. We need to keep a balance between those 2 factors because instable requirements causes a lot of troubles for the project management, while inflexible project may have problems with customers and come up to a complete failure.
- العاملان اللذان يؤثران على معالجة البرمجيات استقرار المتطلبات والمرونة لتغيير المتطلبات. نحتاج لإبقاء موازنة بين هذان العاملين لأن متطلبات التغيير تسبب الكثير من المشاكل لإدارة المشروع، بينما المشروع الغير مرن ربما يكون له مشاكل مع الزبائن وجاءت لاكتمال الفشل