جامعة طرابلس
كلية تقنية المعلومات

مقدمة في هندسة البرمجيات

Introduction to software Engineering

ITGS-213

المحاضرة التاسعة – نماذج إعداد المنظومات

Software Development Models

ربيع2018-

محتويات المحاضرة

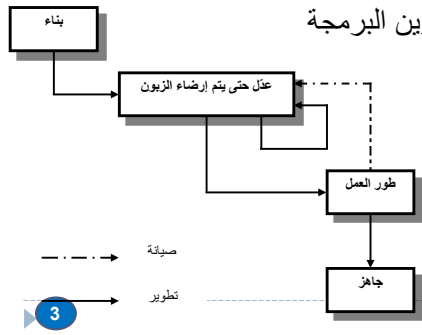
- مراحل دورة حياة المنتج البرمجي(**SDLC** Software Development Life Cycle)
  - نموذج الشلال(Waterfall Model)
  - نموذج التجريبي(Prototype Model)
  - نموذج التزايدي.(Incremental Model)
  - النموذج الحلزوني(Spiral Model)
  - التطوير السريع للبرنامج (Rapid Application Development) (RAD)
  - النموذج الرشيق (Agile Model)

مقدمـــــــــــــــة

سنتناول بعض النماذج المستخدمة في اعداد المنظومات والتي ساهمت في التعامل مع الكم الهائل من المتطلبات الجديدة والمتنوعة والمعقدة والتعامل مع أزمة البرمجيات.

حيث كان في الخمسينات اعداد المنظومات يتم بطريقة بدائية تسمى **طريقة بناء-ثم-إصلاح**. أي اكتب برنامج ثم صحح الخطأ وجربه الى أن تصل الى الحل النهائي.

تعمل هذه الطريقة بصورة جيدة على تمارين البرمجة ذات طول يتراوح من 100 إلى 200 سطر.

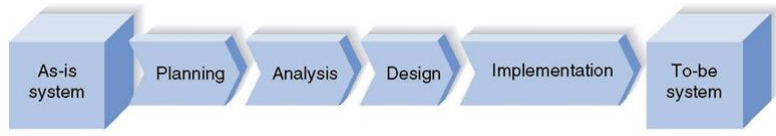


3

مقدمـــــــــــــــة

- ومنذ بداية السبعينات بدأ اعداد المنظومات يتم بطريقة صحيحة وهذه الطريقة تتكون من مجموعة من المراحل عرفت بإسم دورة حياة اعداد المنظومات Software Development Life Cycle **SDLC**.

في دورة حياة اعداد المنظومات يتم استخدام مجموعة نماذج مختلفة لعمليات البرمجة في هندسة البرمجيات . نموذج اعداد البرمجيات هو أسلوب لتنظيم نشاطات عمليات البرمجيات وفقاً لطريقة معينة .



4

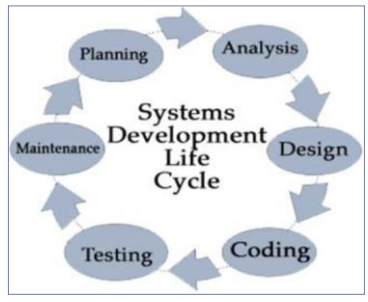
عمليات البرمجيات
Software processes

- كل نموذج هو مجموعة من الأنشطة المرتبة هدفها النهائي تطوير منتج برمجي جديد أو تحسين منتج موجود.
- هذه النشاطات تهدف الي تطوير وتقييم البرمجيات .وتحتوي النشاطات على عمليات تساعد في اداء انشطة هندسة البرمجيات بطريقة منظمة.
- كل عملية من عمليات البرمجيات تتضمن العديد من الخطوات او (المهام) لكل منها مدخلاتها ومخرجاتها, وتأثيرها على جودة المنتج.
- بالتالي يوجد العديد من النماذج و المنهجيات لإدارة هذه العمليات (نماذج اعداد البرمجيات)

نماذج اعداد المنظومات
Software Development Models

- يوجد العديد من النماذج المختلفة لإعداد المنظومات التي تم تطوريها خلال الفترة من السبعينات الي اواخر التسعينات وهي:

1. النموذج التدفقي أو الانحداري Waterfall Model
2. نموذج العرض التجريبي Prototype Model
3. النموذج التزايدي Incremental Model
4. النموذج اللولبي) الحلزوني Spiral Model)
5. نموذج التطوير السريع للبرنامج(RAD)
   Rapid Application Development
6. النموذج الرشيق (Agile Model)

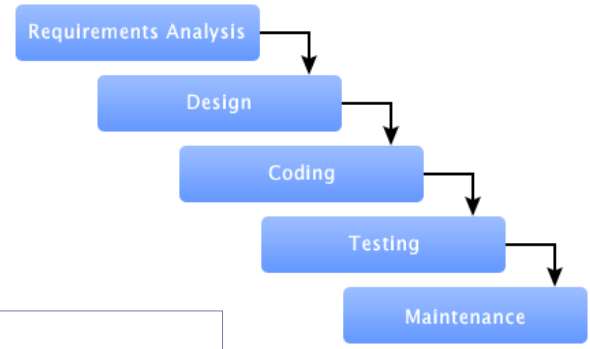


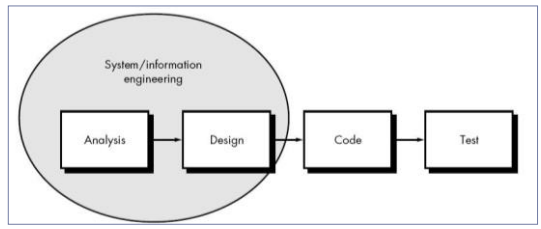
Waterfall Modelالنموذج التدفقي

- يسمي النموذج الشلالي أوالانحداري أو النموذج التتابعي الخطي (Linear sequential) نظرا لتتابع المراحل أو" دورة الحياة التقليدية "هو) اول نموذج تم استخدامه بداية السبعينيات (لهندسة البرمجيات يقترح منهجاً تتابعياً منتظما لتطوير البرمجيات، يبدأ عند مستوى النظام ويتقدم تباعاً إلى التحليل فالتصميم فالشفرة فالاختبار فالصيانة .

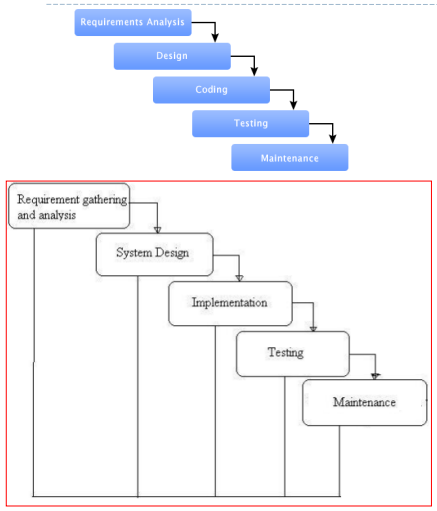
- يوضح دورة حياة البرمجيات بشكل تدرجي ، تبدأ من المرحلة الاولى حتي الاخيرة. كل مرحلة تبدأ بعد الانتهاء من المرحلة السابقة.

Waterfall Modelالنموذج التدفقي





النموذج التدفقي أو الشلالي Waterfall Model



النموذج التدفقي المعدل

- النموذج السابق يفترض تبات المراحل وخاصة في مرحلة جمع المتطلبات وهي من اكثر المراحل تغيرا.
- انه لا يسمح للزبون بمشاهدة النظام الا في المراحل المتأخرة جداً.
- ونتيجة لذلك تم تعديل في هذا النموذج واصبح **النموذج التدفقي المعدل** بحيث يمكن الرجوع للمراحل السابقة في حالة حدوث تغير في المتطلبات

مميزات وعيوب
النموذج التدفقي أو الشلالي Waterfall Model

❑مميزات النموذج التدفقي:

1.انه بسيط وسهل الفهم.

2.يعمل بشكل جيد في المشاريع الصغيرة والتي تكون فيها المتطلبات واضحة.

❑عيوب النموذج التدفقي

1.ليس نموذج جيد للمشاريع الكبيرة والمعقدة.

2.ليس جيد في المشاريع التي عادة ما تكون فيها تغير المتطلبات عالي جدا.

3.قد يسبب التكرار أو التعديلات ارتباكا) فوضى (مع تقدم فريق المشروع.

متى نستخدم النموذج الشلالي؟

1. المتطلبات واضحة ومعرفة بشكل جيد.
2. هناك تعريف ووصف مستقر للمنتج.
3. عند بناء إصدار جديد من منتج موجود مسبقاً.
4. عند نقل منتج موجود إلى منصة عمل جديدة.

نموذج العرض التجريبي
Prototype Model

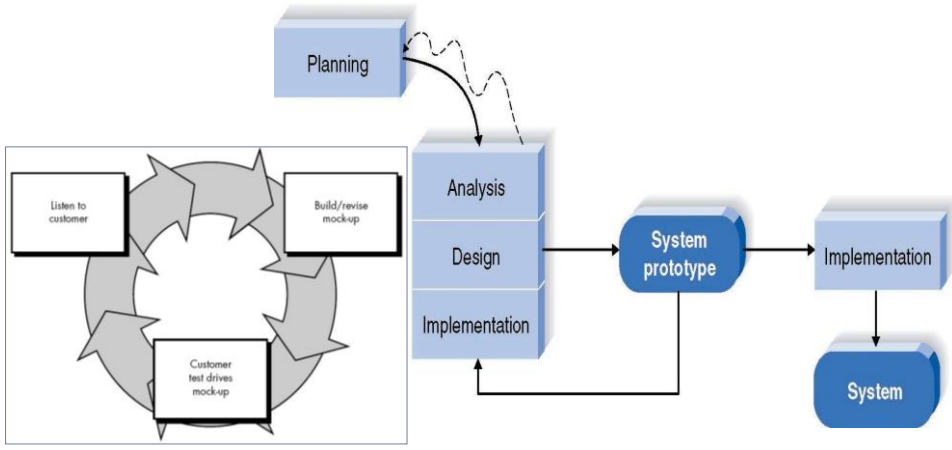
- جاء هذا النموذج لحل القصور في النموذج التدفقي (مشكلة المتطلبات الخاصة بالزبون الغير مدرك للمتطلبات .(غالبا ما يعرف الزبون مجموعة من الأهداف العامة للبرنامج، ولا يحدد بالتفصيل كل المتطلبات.
- يقوم معد المنظومة بإعداد نموذج تجريبي (منظومة تجريبية) كعينة لعرضها على الزبون.
- تستخدم هذه العينة عادة لتوضيح لمساعدة الزبائن على فهم متطلبات النظام ، وكبداية لإصدار عدة نسخ متتالية للوصول للمنظومة النهائية التي ستسلم للزبون فيما بعد.
- هذا النموذج يتم عرض وظائف محددة للمنظومة ابتداء من نسخة اولية.

نموذج العرض التجريبي
Prototype Model

- يبنى النظام كسلسلة من الإصدارات التزايدية تسلم تباعا للزبون حيث

1. يبني المطورون **نموذج أولي** (**Prototype**) خلال مرحلة جمع المتطلبات.
2. يقوم المستخدمون أو الزبون بتقييم النموذج المنجز.
3. يقوم المستخدمون أو الزبون باقتراح الإجراءات التصحيحية .
4. يقوم المطورون بتحسين النموذج وبناء نسخة جديدة.
5. عند بلوغ الزبون درجة مقنعة من الرضى والكفاية، يقوم المطورون بمراجعة شفرة النموذج الأخير لتحسينه من خلال تطبيق المعايير اللازمة لاعتباره منتجاً نهائياً.

نموذج العرض التجريبي
Prototype Model



مميزات و عيوب
نموذج العرض التجريبي

- مميزات النموذج التجريبي:

1. يقلل من مشاكل تحديد المتطلبات.
2. يوفر دعم ومشاركة اكبر من طرف الزبون في المراحل المبكرة للمشروع.
3. يستخدم عندما لا يكون هناك رؤية واضحة للمتطلبات ، بالتالي هو مناسب للبيئة التي يكثر فيها تغير المتطلبات.
4. يمكن استعماله لتدريب المستخدمين قبل التسليم.

مميزات و عيوب
نموذج العرض التجريبي

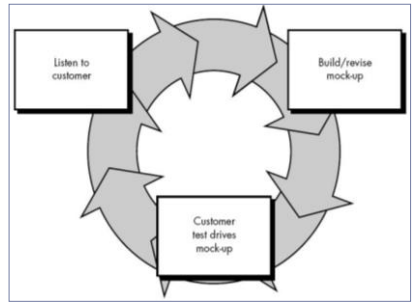
- عيوب النموذج التجريبي:

1. قد يضغط الزبون باتجاه تسليم نموذج غير نهائي.
2. قد تتحول عملية تحسين الطراز إلى حلقة غير منتهية مع اكتشاف المزيد من الاحتياجات والمتطلبات.(scope creep)
3. يجب أن يتمتع فريق التطوير بمهارات عالية.
4. قلة استخدام التوثيق.

نموذج العرض التجريبي
Prototype Model

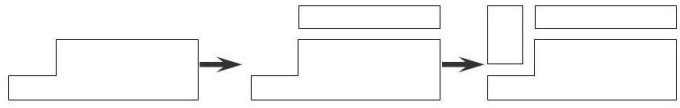
- متي نستخدم العرض التجريبي؟

1. لتوضيح و تحديد وتنقيح المتطلبات.
2. محاولة فهم المسألة المراد حلها.
3. نسخة بدائية للتوصل الي النسخة النهائية او ما يسمي بالإصدار النهائي.



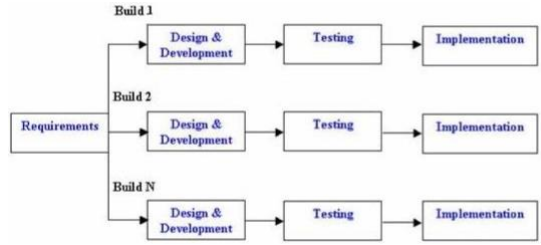
النموذج التزايدي
Incremental Model

- الهدف من هذا النموذج هو **تقليص وقت انتظار الزبون** للحصول على المنظومة النهائية **وخاصة المنظومات الكبيرة والمعقدة** والتي تستغرق مدة طويلة لإعدادها.
- وبالتالي تسليم المنظومة علي دفعات يكون الحل. حيث يتم تقسيم المنظومة الي اجزاء أو بناء Build يتم تسليمها للزبون تباعاً في فترات زمنية متلاحقة.
- كل جزء تمر عليه جميع مراحل تطوير البرمجيات ليتم تشغيله فيما بعد في موقع الزبون.
- **النموذج التزايدي هو عبارة عن خليط من النموذج التدفقي و نموذج العرض التجريبي.**

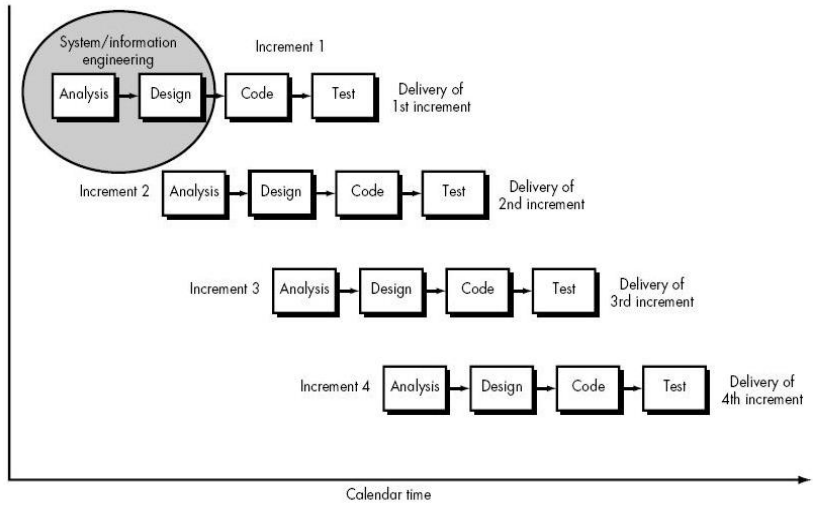


النموذج التزايدي
Incremental Model

- فمثلا قد تنجز برنامج معالجة النصوص باستخدام النموذج ألتزايدي كإنشاء الملفات والتحرير في التزايد الأول، وتقدم إمكانيات أكثر تطورا (تعقيدا وتقدما) كتحرير والطباعة وادراج الجداول التزايد الثاني، وتقدم التصحيح الإملائي والقواعد في التزايد الثالث، وتقدم إمكانات متقدمة لضبط تنسيق الصفحة (page layout) في التزايد الرابع.



النموذج التزايدي
Incremental Model



خصائص النموذج التزايدي
Incremental Model

1. يتكون المنتوج عادة من اجزاء أو بناء.Build
2. كل جزء) بناء (يتم تصميمه وكتابته ثم تسليمه للزبون.
3. يعطي المنتوج للزبون تدرجيا علي شكل دفعات جزء بعد جزء.

مميزات وعيوب النموذج التزايدي
Incremental Model

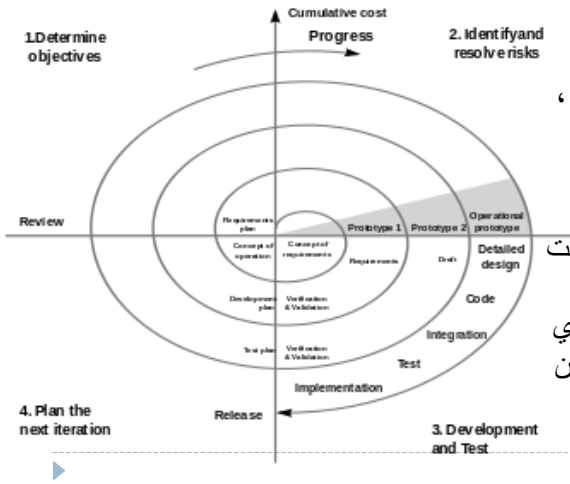
- **مميزات النموذج التزايدي**

1. لا يحتاج الي الكثير من أعضاء فريق العمل مما يودي الي توفير التكلفة.
2. يستطيع الزبون أن يرى منظومة قابلة للعمل في وقت مبكر.
3. يمكنك تسليم الأجزاء البرمجية العاجلة ، و تأجيل الباقي فيما بعد.

- **عيوب النموذج التزايدي**

1. يحتاج الي تخطيط وتصميم جيد.
2. التكلفة أكثر من النموذج الانحداري.

النموذج الحلزوني
Spiral Model

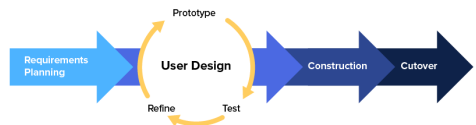


- تسمح المنهجية اللولبية للفرق باعتماد نماذج SDLCمتعددة بناءً على أنماط المخاطر للمشروع المحدد . مزيج من النهج التكراري والشلال ، التحدي مع النموذج الحلزوني هو معرفة متى تكون اللحظة المناسبة للانتقال إلى المرحلة التالية
- .يمكن أن تستفيد الشركات التي ليست متأكدة من متطلباتها أو تتوقع تعديلات كبيرة خلال مشروعها الذي يتسم بمخاطر متوسطة إلى عالية من قابلية التوسع في هذه المنهجية.

التطوير السريع للبرنامج (RAD) (Rapid Application Development)

- التطوير السريع للتطبيق هو منهجية تطوير البرمجيات التي تتضمن أساليب مثل التطوير التكراري ووضع النماذج الأولية للبرمجيات
- يستند نموذج RAD (Rapid Application Development) Modelعلي النماذج والتطوير التكراري مع عدم وجود تخطيط محدد المعنية. عمليه كتابه البرنامج نفسه ينطوي علي التخطيط المطلوب لتطوير المنتج.
- يركز تطوير التطبيقات السريعة علي جمع متطلبات العملاء من خلال ورش العمل أو مجموعات التركيز ، والاختبار المبكر للنماذج الاوليه من قبل العميل باستخدام المفهوم التكراري ، وأعاده استخدام النماذج الحالية (المكونات) ، والتكامل المستمر التسليم السريع.

Rapid Application Development (RAD)



مميزات نموذج تطوير التطبيقات السريع RAD

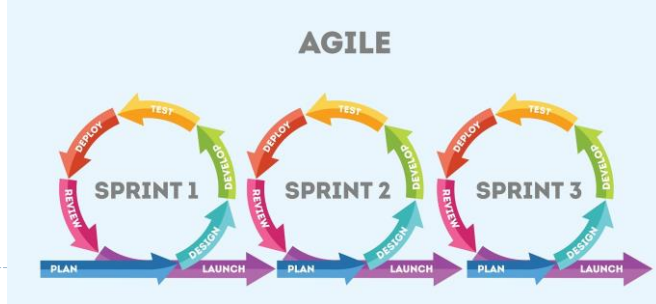
1. يمكن استيعاب الاحتياجات المتغيرة.
2. وقت التكرار يمكن ان تكون قصيرة مع استخدام أدوات راد قويه.
3. الانتاجيه مع عدد اقل من الناس في وقت قصير.
4. تقليل وقت التطوير.
5. تشجيع ملاحظات العملاء.

عيوب نموذج تطوير التطبيقات السريع RAD

1. يتطلب المطورين ذوي المهارات العالية/المصممين.
2. الاعتماد العالي علي مهارات النمذجة.
3. غير قابله للتطبيق علي مشاريع أرخص كتكلفه النمذجة وغيرها.
4. يتطلب مشاركه المستخدم طوال دوره الحياة.
5. درجة تعقيد عالية

النموذج الرشيق
Agile Model

منهجية أجايل هي عكس نهج الشلال .بدلاً من التعامل مع المتطلبات والتصميم والاختبار كخطوات متسلسلة كبيرة ، فإن النموذج الرشيق يجعلها جميع العمليات الجارية التي تتطلب مشاركة المطورين والإدارة والعملاء
.يتم تقسيم العمل عادةً إلى شرائح من 2 إلى 4 أسابيع تُعرف باسم "سباقات السرعة "Sprint، حيث تقوم الفرق المسؤولة بمعالجة الاحتياجات الرئيسية لعملائها وإجراء الاختبارات أثناء ذهابهم .تميل Agileإلى العمل بشكل جيد في المؤسسات الصغيرة ، وخاصة الشركات الناشئة ، حيث تكون السرعة والمرونة ضرورية



عند اختيار أفضل نهجSDLC، من المهم أن تتذكر أن أحد الحلول قد لا يناسب كل سيناريو أو عمل. قد تعمل بعض المشاريع بشكل أفضل مع نهج الانحدار ، بينما سيستفيد البعض الآخر من المرونة في النماذج الرشيقة أو التكرارية