**المرحله الرابعه التنفيذ " التنجيز " Implementation**

يجري في هذه المرحلة بناء النظام أو شراؤه (في حالة تقرير شراء حزمة برمجيات جاهزة) ، تكون هذه المرحلة عادة أطول المراحل وأكثرها كلفة، وهي تتألف من ثلاث خطوات:

1. **إنشاء النظام System Construction :** يجري بناء النظام واختباره للتأكد من أدائه العمل كما جرى تصميمه ، وتعتبر عملية الاختبار من أكثر العمليات كلفة.
2. **التثبيت عند الزبن Installation :** وهنا يجري وضع النظام لدى الزبون تدريجيًا أو كليًا حسب إجرائية يتفق عليها ، وتتضمن هذه المرحلة وضع خطة لتدريب المستخدمين Training Plan على النظام الجديد .
3. **وضع خطه لدعم الزبون Support Plan :** تتضمن إجراء مراجعة للنظام في مرحلة الاستثمار، وتحديد التعديلات الصغيرة أو الكبيرة التي يحتاجها النظام.

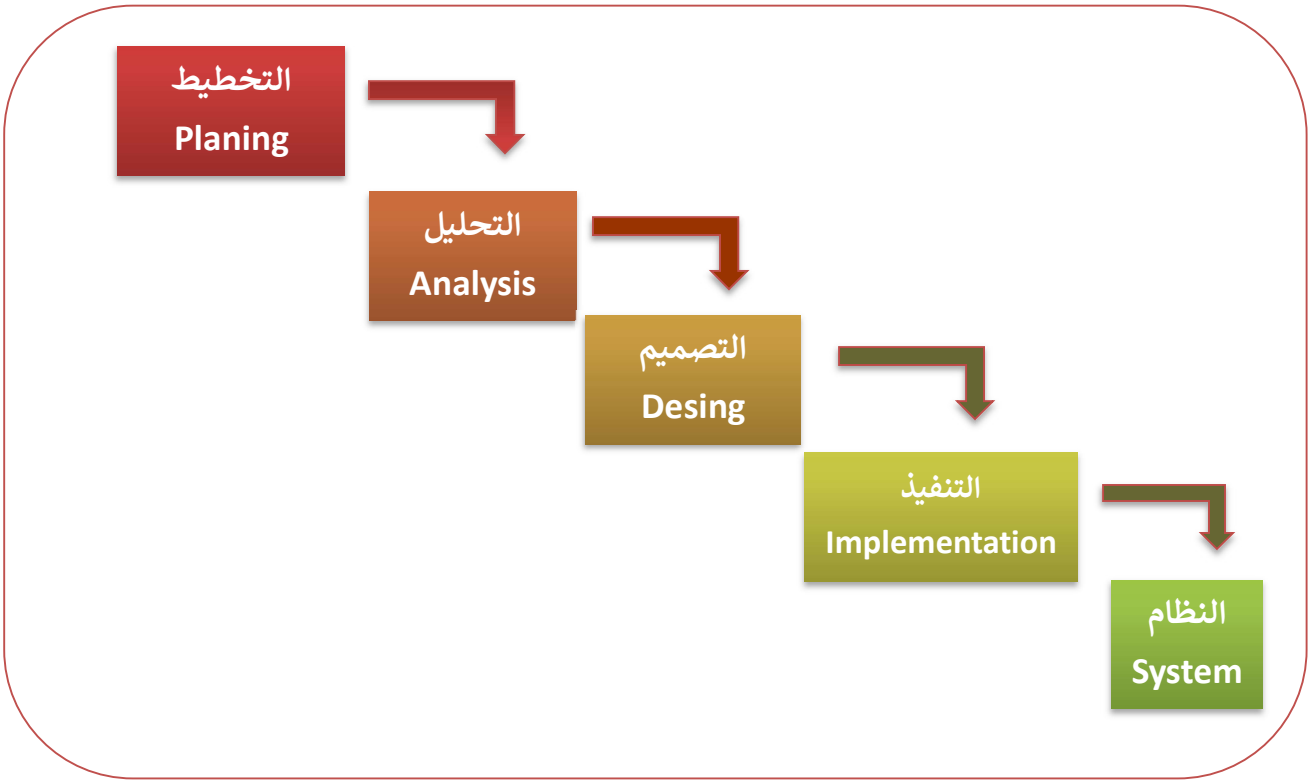
**منهجيات تطوير النظام System Development Methodology**

يستخدم مصطلح منهجية تطوير النظام او البرمجيات ( Software Development Methodology) للتعبير عن إطار العمل المتبع لهيكلة وتخطيط والسيطره على عملية تطوير نظام المعلومات ، وهنالك العديد من منهجيات تطوير البرمجيات نذكر منها ، التطوير الشلالي ، التطوير على التوازي ، التطوير السريع ، التطوير على مراحل ، النمذجه الاولويه ، النمذجه الاوليه مع رمي النموذج ، التطوير الرشيق والبرمجه الحديه ، وسنشرح تباعا كل منهجيه من المنهجيات المذكوره ..

وتعرف المنهجيه بأنها المقاربه المستخدمه أثناء وضع دورة حياة تطوير النظم SDLC بحيث انها تتألف من لائحه من الخطوات والنواتج ، وتختلف المنهجيات بعضها عن بعض بحسب تركيزها على اجرائيات العمل او على المعطيات التي تدعم العمل ، وبحسب ترتيبها وتأكيدها على كل مرحله من مراحل دورة حياة تطوير النظم .

## التطوير الشلالي Waterfall Development

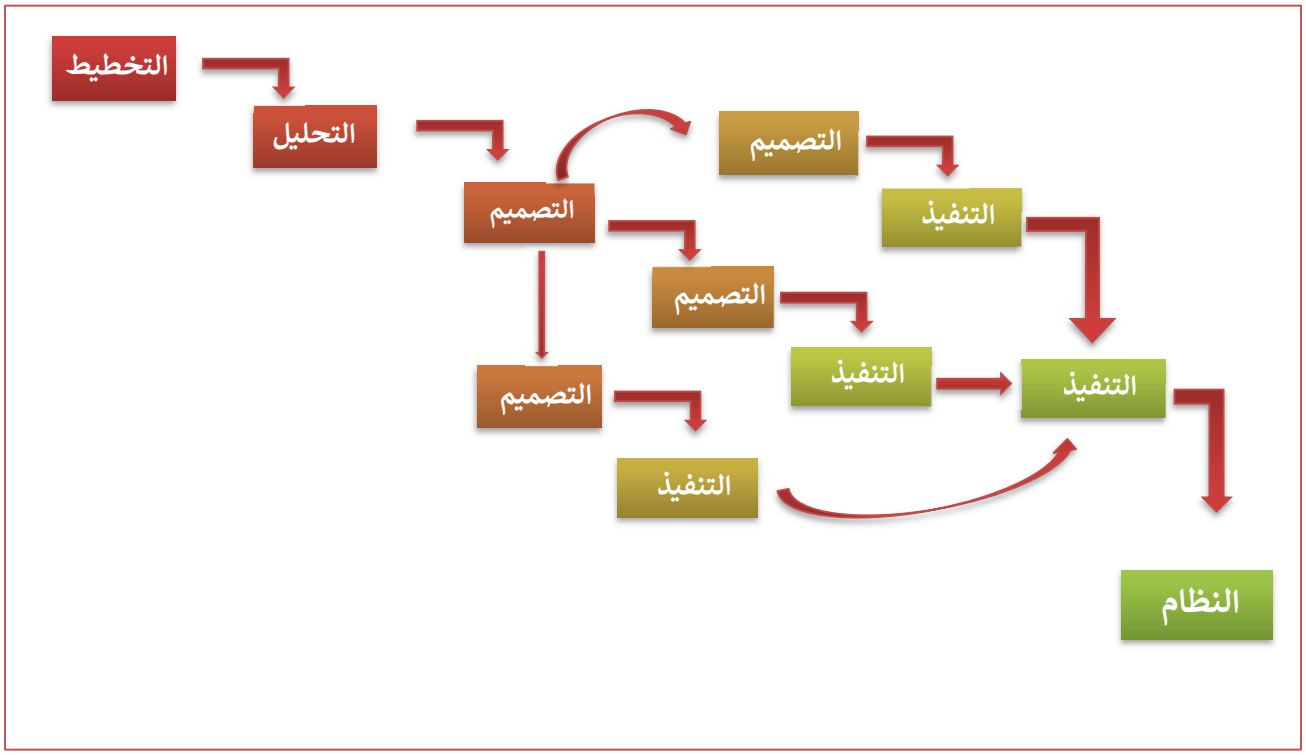
في هذه المنهجيه ينتقل المحللون والمصممون انتقالا تتابعيا من مرحله الى اخرى وتتميز هذه المنهجيه بتحديد متطلبات النظام قبل البدء بالبرمجه بوقت طويل ، ومع تقدم المشروع تضعف إمكانية اجراء التعديلات على المتطلبات . ولها ايضا مساوي تتمثل في انه يجب ان ينتهي التصميم تماما قبل البدء في البرمجه ( مع ملاحظة انها لاتعتبر مساوي في وجهة نظر بعض المحللين والمصممين ) ، كما يمر وقت طويل بين طلب النظام وعملية تسليمه بصوره نهائيه .




الشكل 2-2 يوضع منهجية التطوير الشلالي

### التطوير على التوازي Parallel Development

تحاول المنهجيات التي تعتمد التطوير على التوازي ان تعالج موضوع الفتره الزمنيه الطويله التي تمر بين طلب النظام (طلب المستخدم) وتسليمه (تنفيذ النظام ) ، فبدلاً من القيام بالتصميم كاملاً ثم الانتقال الى التنفيذ ( كما في التطوير الشلالي ) ، يوضع تصميم عام للنظام ككل ، ثم يقسم المشروع الى عدد من المشاريع الفرعيه المستقله التي يمكن تصميم كل منها وتنفيذه (تنجيزه) على التوازي مع المشاريع الفرعيه الاخرى ثم يقوم في النهايه بتجميع هذه المشاريع الفرعيه و الانظمه الفرعيه لتكوين النظام الكلي .




الشكل 2-3 يوضح منهجيه التطوير على التوازي

### التطوير السريع للتطبيقات Rapid Application Development RAD

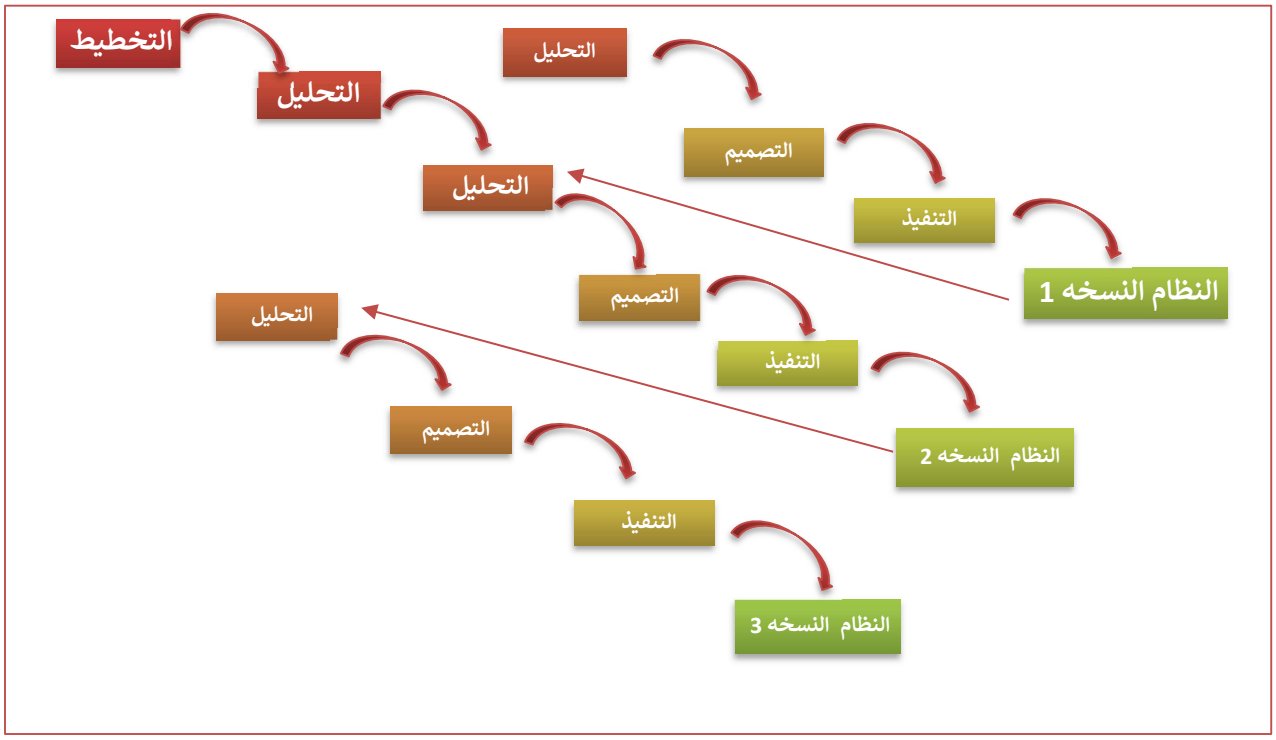
ظهرت المنهجيات المعتمدة على التطوير السريع للتطبيقات للتغلب على نقطتي الضعف المذكورتين آنفًا في منهجيات التصميم البنيوي ( الشلالي ، على التوازي ) . لتحقيق هذا الهدف، تنسق المنهجيات المعتمدة على RAD بين مراحل دورة حياة تطوير النظام للحصول على أجزاء من النظام بسرعة وتضعها بين يدي المستخدم. إن حصول المستخدم على أجزاء من النظام في وقت مبكر يتيح له فهمًا أفضل للنظام مما يجعله يقترح بعض التعديلات التي تجعل النظام أكثر تلبية لاحتياجاته.

تنصح معظم المنهجيات المعتمدة على التطوير السريع للتطبيقات أن يستخدم المحللون تقنيات مخصصة وأدوات حاسوبية خاصة لتسريع مراحل التحليل والتصميم والتنجيز، مثل أدوات هندسة البرمجيات بمعونة الحاسب CASE وجلسات JAD ( Joint Application Development) " التطوير المشترك للتطبيقات وهي تقنيه تقضي بمشاركة كل من المحلل والزبون في التطوير" ، و لغات البرمجه المرئيه مثل ( Visual Basic ) .

ومع ذلك تبقى هناك في المنهجيات المعتمدة على التطوير السريع للتطبيقات مشكلة خفية تكمن في إدارة توقعات الزبون ، فمع زيادة سرعة تطوير النظام يزداد فهم الزبون للتقانات المستخدمة وتزداد طلباته وتوقعاته من النظام، مما يجعل متطلبات النظام تتضخم بشكل كبير أثناء المشروع.

### التطوير على مراحل Phased Development

تعتمد هذه المنهجيات على تجزئة النظام الكلي الى سلسله من الاصدارات التي يجري تطويرها تتابعياً . ففي مرحلة التحليل يجري تحديد المفهوم الكلي للنظام ، ثم يقوم فريق المشروع والمستخدمون والممول بتصنيف المتطلبات في سلسله من الاصدارات المتتابعه حيث تشكل المتطلبات الاساسيه والاكثر اهميه الاصدار الاول .




الشكل 2-4 يمثل منهجية التطوير على مراحل

ننطلق من مجموعة المتطلبات هذه وبعد تحليلها بدقه الى تصميمها وتنجيزها (تنفيذها) فنحصل على الاصدار الاول من النظام ، نكرر الامر نفسه عدة مرات بحيث اننا بعد الانتهاء من كل اصدار نبدأ العمل على الاصدار التالي الذي ينطلق من مجموعه المتطلبات السابقه مضافاً إليها الافكار الجديده التي يأتي بها المستخدم بعد تجربته مع الاصدار السابق .

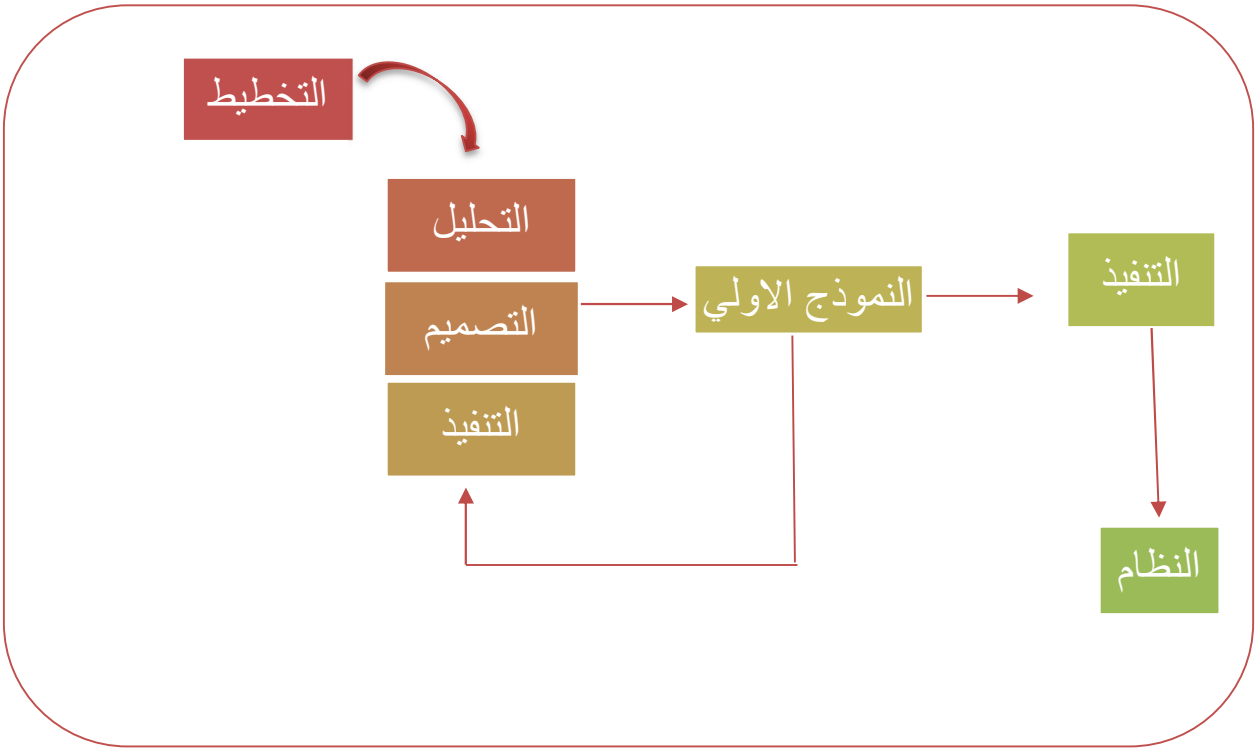
### النمذجه الاوليه Prototyping

في هذه المنهجيات يجري العمل في مراحل التحليل والتصميم والتنفيذ (التنجيز ) بشكل تسايري ، بحيث تؤدي هذه المراحل الثلاثه ضمن حلقه تكراريه الى ان يتم إنجاز كامل النظام .

نقوم باجراء تحليل وتصميم اساسيين ، ثم نقوم مباشره ببناء النموذج الاولى للنظام . ان النموذج الاولى برنامج " سريع وفج " يعتبر إصداراً أولاً مصغرا عن النظام ، ويمتلك عدداً قليلاً من الوظائف والصفات المطلوبه ، يشكل هذا الاصدار الجزء الاول الذي يستخدمه الزبون ، ويعرض عادة على المستخدم والممول

لأخذ ملاحظاتهم وردود افعالهم ، وبناءً على هذه الملاحظات يجري العمل على نموذج فيه المزيد من الوظائف والصفات المطلوبه ، وتكرر العمليه الى حين الوصول الى نظام بكامل مواصفاته .

وتتميز هذه المنهجيه بانها توفر للمستخدم نظاماً يمكنه التفاعل معه وإن لم يكن هذا النظام جاهزاً للاستخدام الفعلي ، اما مساوئ هذه المنهجيات فهي انها تؤدي في غالب الاحيان وبسبب كثرة التعديلات التي تطرأ على النموذج الاولى الى الحصول على تصميم شئ للنظام .

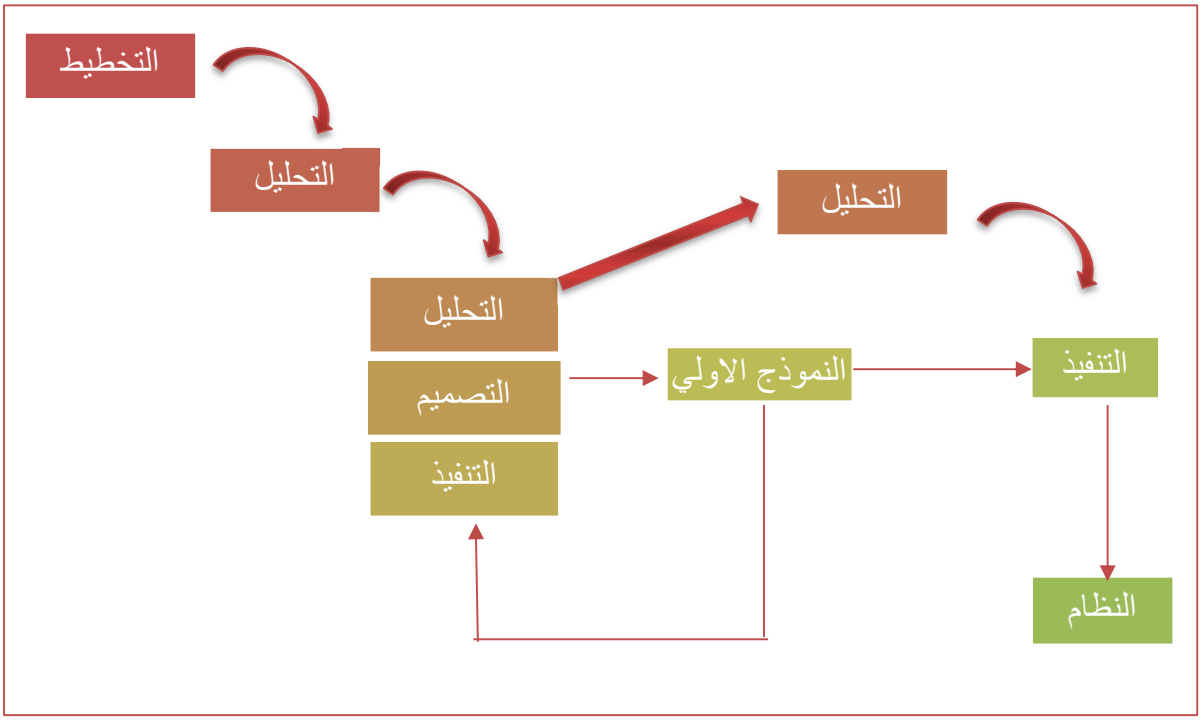



الشكل 2-5 يمثل منهجية التطوير المعتمده على النمذجه الاوليه

### النمذجه الاوليه مع رمي النموذج Throwaway Prototyping

تشبه هذه المنهجيات تلك التي اوردناها في الفقره السابقه (المنهجيات المعتمده على النموذج الاولى ) في انها تعتمد على صنع نماذج أوليه ، غير أنها تختلف عنها في أنه يجري صنع النماذج الأوليه في موضع مختلف من دورة الحياة .

تقوم هذه النماذج بدور مختلف عن مثيلاتها التي اوردناها في الفقره السابقه (المنهجيات المعتمده على النموذج الاولى ) كما يكون مظهرها مختلفاً كليا . حيث يقوم المحللون في هذه المنهجيات بإجراء تحليل عميق نسبياً يتم من خلاله جمع المعلومات وتطوير افكار حول مفهوم النظام . قد تكون بعض خصائص النظام التي يطلبها المستخدم غير واضحه او خياليه او تمثل تخدياً تقانياً ، وعليه يجري كل من هذه الطروحات عبر بناء نموذج اولي تصميمي Design Prototype . حيث لايعتبر هذا النموذج نظاما لانه في الحقيقه يمثل جزء من النظام الذي يحتاج الى تفصيل .




الشكل 2-6 يمثل المنهجيه المعتمده على النمذجه الاوليه مع رمي النموذج

### التطوير الرشيق Agile Development

أخذت هذه المنهجيات بالظهور حديثاً وهي تركز بشكل كبير على عملية البرمجه وتمتلك عددا قليلا من القواعد والممارسات مما يجعل اتباعها سهلاً ، وتهدف هذه المنهجيه الى الانسياب عبر دورة حياة تطوير

النظام البرمجي والغاء الكثير من الحمل الاضافي الذي ينتج عن عمليات النمذجه والتوثيق مما يؤدي الى توفير الوقت الذي تستغرقه هذه العمليات ، وبدلاً عن ذلك تركز المشاريع على تطوير بسيط وتكراري للتطبيقات ، ومن امثلة هذه المنهجيات نورد البرمجه الحديه XP .

**البرمجه الحديه Extreme Programming (XP)**

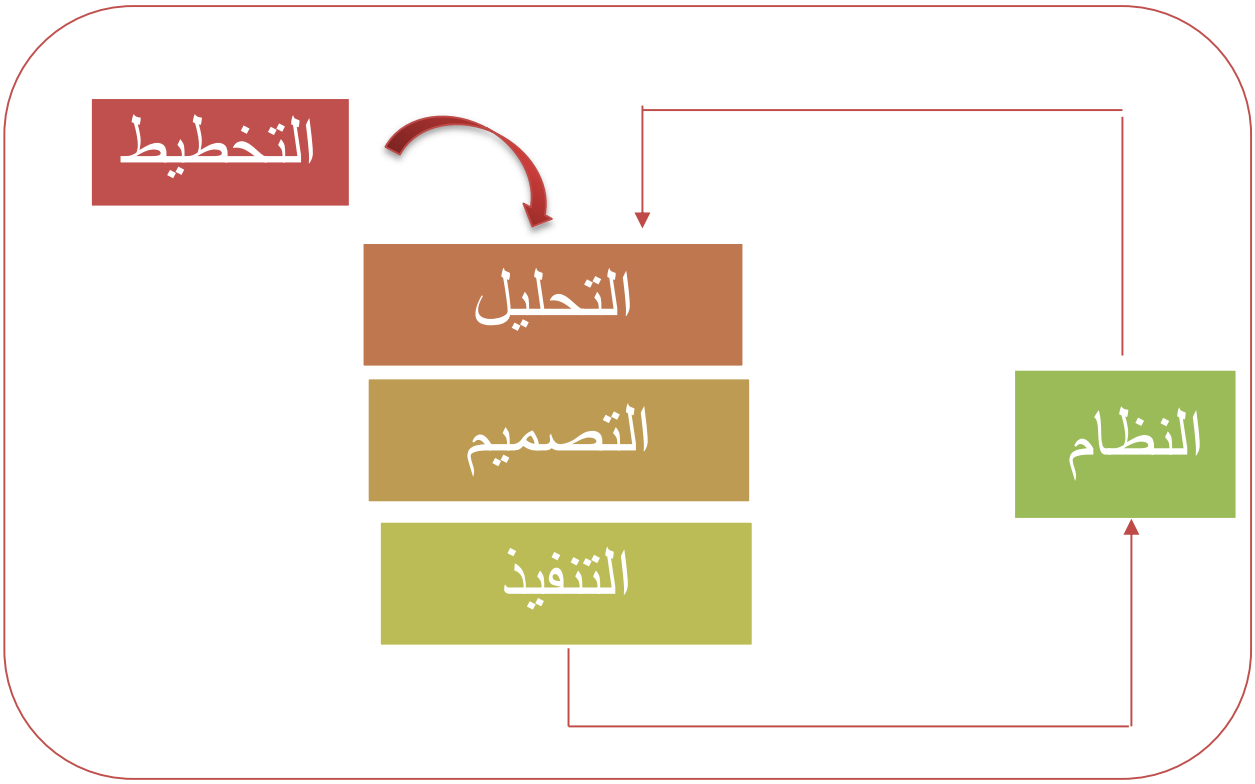
تقوم هذه المنهجيه على القيم الاربعه التايه :

- التواصل : يجب ان يقدم المطورون رد فعل سريع على طلبات الزبون بشكل دائم .
- البساطه : يجب أن يحافظ المطورون على المبدأ Keep It Simple and Stupid KISS.
- رد الفعل : يجب ان يقوم المطورون بتعديلات تزايديه incremental حتى يكبر النظام تدريجيا ، كما يجب ان يستوعبوا التعديلات ويعملوا على احتوائها لا مجرد ان يقبلو بها .
- الشجاعه : التي يجب ان يتحلى بها المطورون .

**كما تعتمد XP المبادئ الاساسيه التاليه لصنع أنظمه ناجحه :**

- الاختبار المستمر .
- الترميز البسيط الذي يقوم به زوجان من المبرمجين .
- التواصل الوثيق مع المستخدم لبناء النظم بناءً سريعاً.

وبعد عمليه تخطيط سريعه تبدأ فرق العمل بإجراء التحليل والتصميم والتنفيذ بشكل تكراري.




الشكل 2-0 يمثل المنهجيه المعتمده على البرمجه الحديه

**اختيار منهجيه التطوير المناسبه**

ليس من السهل اختيار المنهجية المناسبة للتطوير، إذ لا توجد منهجية أجمع المطورون على أنها هي الفضلى! كما أن لكل شركة تطوير معاييرها ومقاييسها النموذجية. سنسلط الضوء في هذه الفقرة على بعض النقاط التي يمكننا استخدامها كمعايير للمقارنة بين المنهجيات.

### وضوح متطلبات المستخدم

عندما يقدم لك المستخدم متطلبات غير واضحة عن النظام وعما يجب أن يفعله، يكون من الصعب فهم هذه المتطلبات بالحديث عنها أو بكتابة تقرير حولها .يحتاج المستخدم في هذه الحالة إلى التفاعل مع التقانة لفهم ما سيفعله النظام وكيف يمكن تطويع هذه التقانة لاحتياجات الزبون.

في مثل هذه الحالات تكون المنهجيات المعتمدة على النمذجة الأولية وعلى النمذجة الأولية مع رمي النموذج هي الأكثر مناسبة لأنها تقدم نماذج أولية للمستخدمين تمكنهم من التفاعل معها في مرحلة مبكرة من دورة الحياة.

### التألف مع التكنلوجيا

إذا كان النظام مصممًا دون أن يكون فريق المشروع متآلفًا مع التكنولوجيا الأساسية فيه، تزداد المخاطرة لأن الأدوات قد لا تكون قادرة على فعل المطلوب منها .في هذه الحالة يكون استخدام المنهجيات المعتمدة على النمذجة الأولية مع رمي النموذج هو الخيار الأفضل، في حين لايكون استخدام النمذجة الأولية مناسبًا.

### تعقيد النظام

تحتاج النظم المعقدة إلى تحليل وتصميم دقيقان .يمكن في هذه النظم استخدام النمذجة الأولية مع رمي النموذج، أو استخدام المنهجيات المعتمدة على التصميم البنيوي .أما استخدام التطوير على مراحل، فقد بينت التجربة أن فرق العمل الذي اعتمدته كانت تولي تحليل النظام المعقد اهتمامًا أقل مما لو استخدمت منهجيات أخرى.

### موثوقية النظام

تعتبر موثوقية النظام عاملاً هامًا في تطوير النظم .وتشكل المنهجيات المعتمدة على النمذجة الأولية مع رمي النموذج الخيار الأفضل عندما تكون الموثوقية ذات أولوية عالية .أما استخدام النمذجة الأولية فلا ينصح به هنا لأنه تنقصه الدقة والتأني في مرحلتي التحليل والتصميم.

**الخطه الزمنيه القصيره**

تناسب هذه المشاريع المنهجيات المعتمدة على التطوير السريع للتطبيقات ، كما أن النمذجة الأولية والتطوير على مراحل يشكلان خيارين ممتازين لمثل هذه المشاريع .أما التطوير الشلالي فهو الخيار الأسوأ ويجب الابتعاد عنه.

**متابعة الخطه الزمنيه**

لا توفر جميع المنهجيات القدرة على متابعة الخطط الزمنية والتحقق من مدى التقيد بها بدرجة واحدة . ونظرًا إلى أن التصميم البنيوي يترك التصميم والتنجيز للمراحل الأخيرة، فإنه يخشى من عدم التمكن من المتابعة .تنقل منهجيات التطوير السريع للتطبيقات الكثير من قرارات التصميم إلى البدايات مما يسمح بالتعرف إلى مواطن المخاطرة العالية والتصدي لها مبكرًا.