نتابع معكم في هذه المحاضرة مع بعض الأمثلة والنماذج لبعض الـ **processes** لكن قبل ذلك يوجد مفهوم يجب أن نضيفه إلى النموذج العام **Generic Process Model** الذي تحدثنا عنه في المحاضرات السابقة وهو مفهوم الـ **Activities Flow** أو تدفق (اتجاه) النشاطات (الإجرائيات) ويدعى أيضاً (**Work Flow, Process Flow**) وهو :

The manner in which the process elements are interrelated to one another. SE-Practitioner Approach 7th Edition p39.

أي هو الوسيلة أو الطريقة التي تتفاعل بها عناصر العملية **Process Elements** مع بعضها البعض, أو بمعنى آخر هو الاتجاه الذي يحدد الانتقال من **Milestone** إلى **Milestone** آخر.

سندرس على مدار محاضرتين مجموعة من الـ**Process Models** والتي يمكن أن تندرج تحت **نمطين وهما:**

- **Agile Processes:**

أي أن هذه الـ**Processes** "رشيقة" أي تدعم التعديل والتغيير المستمر في المتطلبات حسب رغبات الزبون.

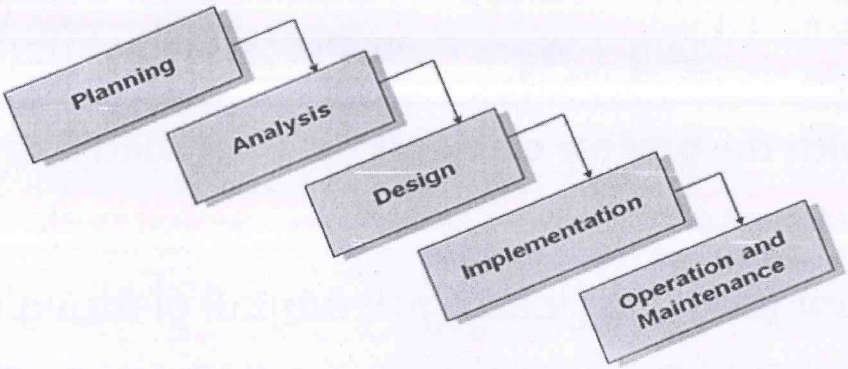
Plan-driven Processes:

أي أن جميع الاطوار في هذه **Processes** مخططة مسبقاً وبالتالي لا نقوم بأي عملية تطوير غير محسوبة ومخططة لها ويطلق عليها أيضا" **Traditional Process Models ومنها:**

❖ Waterfall Process Model (نموذج الشلال):

لدينا هنا نموذجين مقترحين:

النموذج A:



نلاحظ وجود الـ **Milestones** التالية:

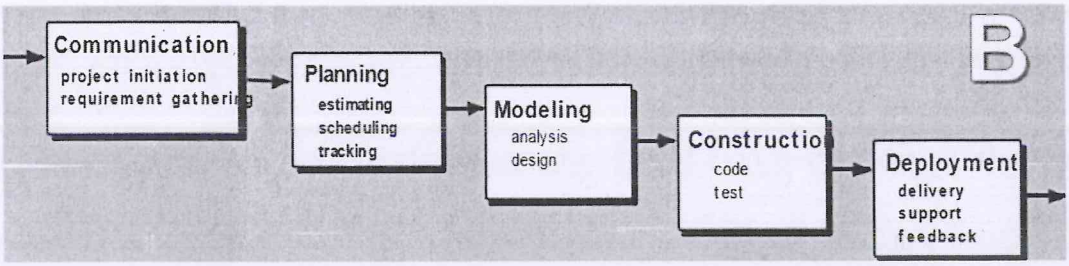
- **Planning**: تحديد الزمن المتوقع والكلفة المتوقعة والتقنيات المستخدمة.
- **Analysis**: وضع الخطوط العريضة (المبدئية) للحل البرمجي ورسم المخططات التحليلية.
- **Design**: وضع المخططات التصميمية (النهائية).
- **Implementation**: عملية تحويل المخططات التصميمية إلى كود.

- **Operation and Maintenance**: المرحلة مابعد تسليم المنتج البرمجي تشمل (أسئلة المستخدم- صيانة المنتج-.....).

ويكون الـ**Process Flow**:

بشكل خطي لأنه وبعد عملية الـ**Planning** تأتي عملية الـ**Analysis** وبعدها عملية الـ**Design** ثم الـ**Implementation** وأخيراً مرحلة الـ**Operation and Maintenance**.

النموذج B:



نلاحظ وجود الـ**milestones** والـ**work tasks** التالية:

- **Communication**:
  1) **Project Initiation** تحليل المشروع.
  2) **Requirement Gathering** تجميع المتطلبات.
- **Planning**:
  1) **Estimating** تقدير المتطلبات (الزمن-الكلفة-....).
  2) **Scheduling** وضع جداول للأنتاج.
  3) **Tracking** المتابعة حسب خطة العمل.
- **Modeling**:
  1) **Analysis**.
  2) **Design**.

- **Construction**:
  1) **Code** كتابة الكود للمنتج.
  2) **Test** تجريب هذا الكود والمنتج.
- **Deployment**:
  1) **Delivery**.
  2) **Support**.
  3) **Feedback**.

والـ **Process Flow** هنا يكون خطي أيضاً لأنه وبعد كل مرحلة ينتقل إلى المرحلة التي تليها مباشرة :

Planning→Analysis→Design→Implement→Maintenance

بعد كل مرحلة من هذه المراحل سينتج لدينا منتج مرحلي (**work product**) فبعد مرحلة الـ **Planning** سيكون لدينا مايسمى (**Investigation Report**) وهو تقرير كامل متكامل عن تحقيق الشروط,وبعد الـ**Analysis** سيكون لدينا (**Specifications**) المواصفات الأساسية للمنتج, وبعد الـ**Design** سيكون لدينا (**Design Report**) مجموعة مخططات رئيسية للمنتج وبعد الـ**Implementation** سيكون هو الـ**Application** وبعد الـ**Maintenance** سيكون تقرير صيانة (**Maintenance Report**) عدد المستخدمين-هل المنتج جيد أم ناقص...

<u>نلاحظ من النموذجين التالي:</u>

- كلا النموذجين <u>يحققان شروط **Process model**</u> باحتوائه على عدد من الأطوار **milestones** و المنتجات المرحلية **work products** ومهمات العمل الجزئية **work tasks** و الانتقالات بين الأطوار تكون خطية

process flow مع التسليم بوجود نقاط اختبار جودة بنهاية كل طور من الأطوار.

- النموذج B يختلف عن النموذج A حيث تمت إضافة Phase كامل (Milestone) وهو الـCommunication كما تم دمجAnalysis & Design ضمن Phase واحدة هي Modeling.
- في كلا النموذجين يتم الانتقال من Phase1 إلى Phase2 باتجاه واحد ولا يتم إلا بعد الانتهاء من الـwork tasks وإنتاج Work product للـPhase1 وبعد هذا الانتقال لا نستطيع العودة إلى الـphase1, فهذه النماذج تحمل صفة الشلال الذي يسقط من مكان مرتفع إلى مكان منخفض (من هنا جاءت التسمية).

ومنه كلا النموذجين هو Waterfall Model لكن من هو الأفضل؟

نلاحظ تخصيص طور كامل لمرحلة التواصل مع الزبون في النموذج B لضمان فهم واضح وكامل لمتطلبات الزبون واعطائه الفرصة الكاملة للتعبير والتغيير ما دمنا ضمن هذا الطور.

إيجابيات وسلبيات استخدام الـWaterfall Process Model:

Advantages and Disadvantages of Waterfall:

- **السلبيات Disadvantages :**
  1) غير قابل لتقبل متطلبات الزبائن المتغيرة.
  2) عدم القدرة على الانتقال من مرحلة إلى أخرى إلا عند الانتهاء من المرحلة السابقة (اتجاه واحد).
- **الإيجابيات Advantages:**

1) المحافظة على النقاط الرئيسية في عملية التطوير.
2) التسلسل الخطي لـ**Flow Activities**.
3) تحكم عالي.
4) بنية وخطة واضحة **Plan-driven**.
5) يتم تسليم تقييم المنتج في نهاية عملية التطوير والاختبار.
6) زمن التنفيذ سريع نسبياً.

استخدام الـ Waterfall Model:

When to use Waterfall Model:

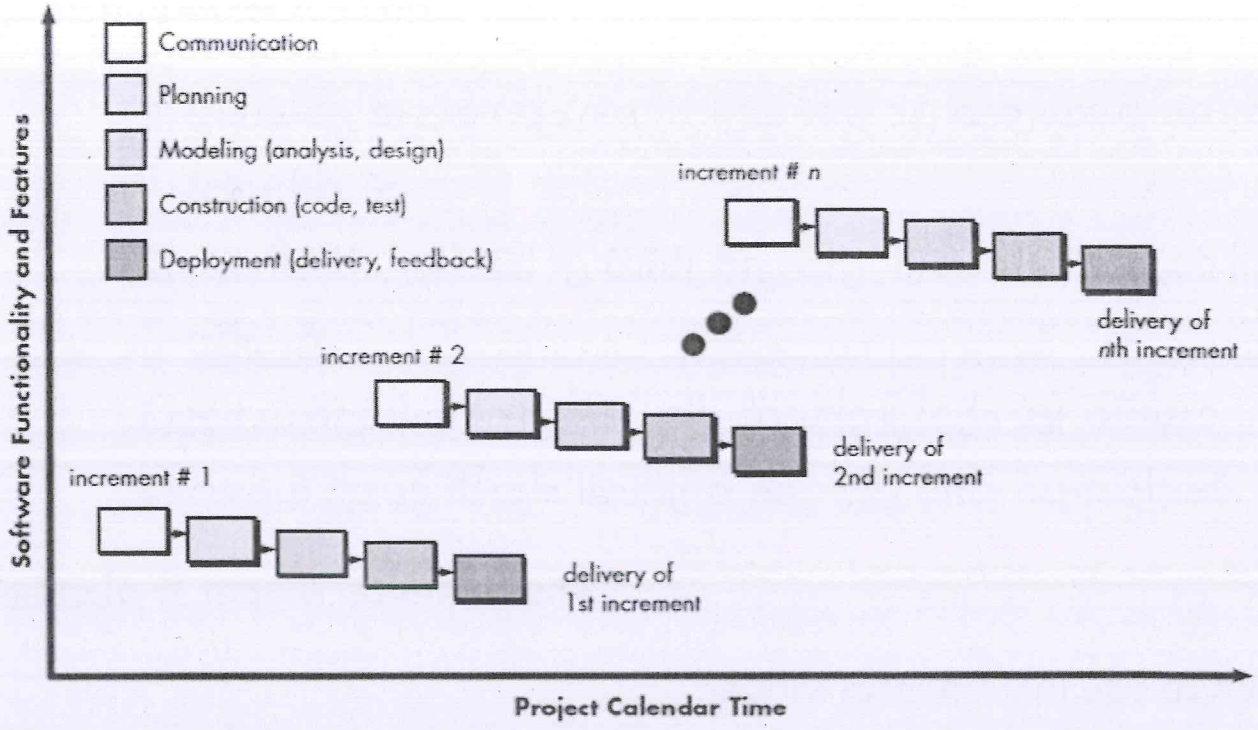
1) عندما تكون المشاريع صغيرة.
2) المتطلبات مفهومة و واضحة و ثابتة.
3) التقنيات المستخدمة مفهومة وليست ديناميكية.
4) التغييرات محدودة إلى حد ما أثناء عملية التطوير.

يوجد تمثيل آخر لنموذج الـWaterfall يدعى **V-model** يمكنكم الاطلاع عليه في الصفحات 39 - 40 من SE: Practitioner Approach 7th edition. وهو نموذج مخصص لتوضيح مراحل وأنواع الاختبار المختلفة وأماكن تصحيح الخطأ إن وجد.

❖ Incremental Process Model: النموذج التزايدي

انطلاقاً من أكبر مشكلة واجهناها عند استخدام نموذج الـ**Waterfall** وهي عدم قابلية التعديل والصيانة المتأخرة تم إنشاء ما يسمى **Incremental Process model** وفيه يتم تقسيم متطلبات المشروع **requirements** إلى **عدة تزايدات** واعتبار كل تزايد مشروع أو منتج

مستقل وبعد الانتهاء منه نقوم بالانتقال لتحضير وتنفيذ التزايد التالي وهكذا...



*يمكن استخدام Waterfall model في انتاج كل increment.

الذي ينتج لدينا يسمى **increment** وهو ليس منتج كامل وإنما جزء منه وبالتالي يمكننا تسليم هذا الجزء للمستخدم بخدمات محدودة ليجربه ويقوم بطلب التعديلات (في حال وجودها) وعلى التوازي يمكننا التفكير بعملية تطوير الجزء الثاني من هذا المنتج...

الإيجابيات Advantages:

1) تقليل كلفة متطلبات الزبائن المتغيرة.
2) من السهل إعطاء المستخدم معلومات عن الجزء الذي تم تطويره وبالتالي المستخدم يمكنه أن يعطي رأيه بالعمل المنجز حتى الآن ويرى كيفية التنفيذ.

3) يمكن للمستخدم أن يستفيد من الجزء الذي تم تطويره ويقوم باستخدامه.

السلبيات Disadvantages :

1) ليست كل مرحلة (جزء) تم إنجازها تعبر عن مدى تقدم عملية تطويرالمنتج.
2) بنية النظام يمكنها أن تنهار في حال الإضافات الجديدة الغير مناسبة.
3) إضاعة الوقت والمال في حال تم تطوير أجزاء غير مناسبة.
4) صعوبة تجميع الأجزاء المطورة في حال كانت غير مترابطة.

استخدام الـ Incremental Model :

When to use Incremental Model:

1) في حالة التغيرات المتكررة وليس الإضافات.
2) في حال المشاريع متوسطة الحجم.
3) عندما يكون المستخدم متشوق لرؤية عملية تطوير المنتج أي عندما يكون مطلوب تسليم جزء من المنتج بسرعة.

❖ Prototyping Model: نموذج النمذجة الأولية

يتم وضع نموذج أولي يحوي الخدمات الأساسية ولايتم التركيز إلا على المتطلبات الوظيفية **Functional Requirements** وتهمل الـ **Non-Functional Requirements** وهو تجريبي والفائدة منه إعطاء المستخدم نموذج أولي (مبدئي) عن المشروع حتى يتم فهم المتطلبات بشكل أفضل بالنسبة للزبون وللمطورين.

ويطلق على هذا النموذج الأولي Core Product.

الإيجابيات Advantages:

1) أقرب (واقعياً) إلى متطلبات المستخدم وبالتالي يمكننا من تجنب الجهد المبذول في عملية التطوير.
2) يحسن جودة التصميم و الاستخدام النظام وقابلية الصيانة لاحقاً.
3) يقلل من جهد تطوير المنتج لاحقاً.

السلبيات Disadvantages :

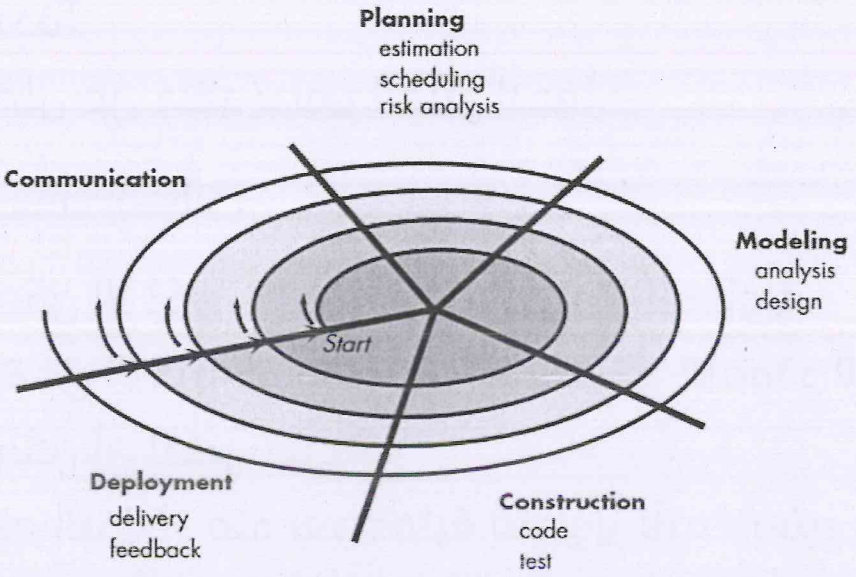
1) يمكن ألا تكون ضمن معايير الجودة المطلوبة.
2) تهمل الاختبارات والتوثيق أثناء تحضير النموذج الاولي.
3) يمكن أن تتغير بسرعة.
4) يتم التخلص منه بعد عملية التطوير لأنه لايعتبر جيداً كـمنتج برمجي, أي الوقت المبذل عليه يعتبر مكلفاً.

استخدام الـ Prototype Model :

1) عندما تكون المتطلبات غير واضحة أو غير محددة.
2) عندما يكون مطلوب نسخة سريعة عن المشروع ككل.
3) لشرح خيارات التصميم.
4) للمساعدة في عملية التحقق من متطلبات المنتج قبل البدء بعملية تطويره.
5) للقيام بعملية الاختبار في وقت مسبق.

❖ The Spiral Model: نموذج اللولب

يعتبر هذا النموذج من النماذج المتزايدة **Incremental** وتعتمد على مبدأ الـ**Slices** ولكن بطريقة تزايدية وهذا النموذج **Risk-Based Process Model** أي أننا نعتمد على هذا النموذج في تطوير البرمجيات ذات الخطورة العالية.



ما هو أول شيء نبدأ بدراسته (تطويره) عند استخدام نموذج الـSpiral ؟

معالجة (تطوير) المتطلبات المعرضة للفشل بشكل أكبر لأننا سنعالجها أكثر من مرة عند كل **Iteration** .

الإيجابيات Advantages:

1) تحليل كمية كبيرة من الخطر ولذلك سيتم تجاهل الخطر بشكل ملحوظ.

2) جيدة ومثالية في حالة المشاريع العالية الخطورة وذات المتطلبات المتغيرة.
3) إمكانية إضافة وظائف إضافية في وقت لاحق.
4) جزء من المنتج يتم إنتاجه في وقت مبكر من حياة المنتج.

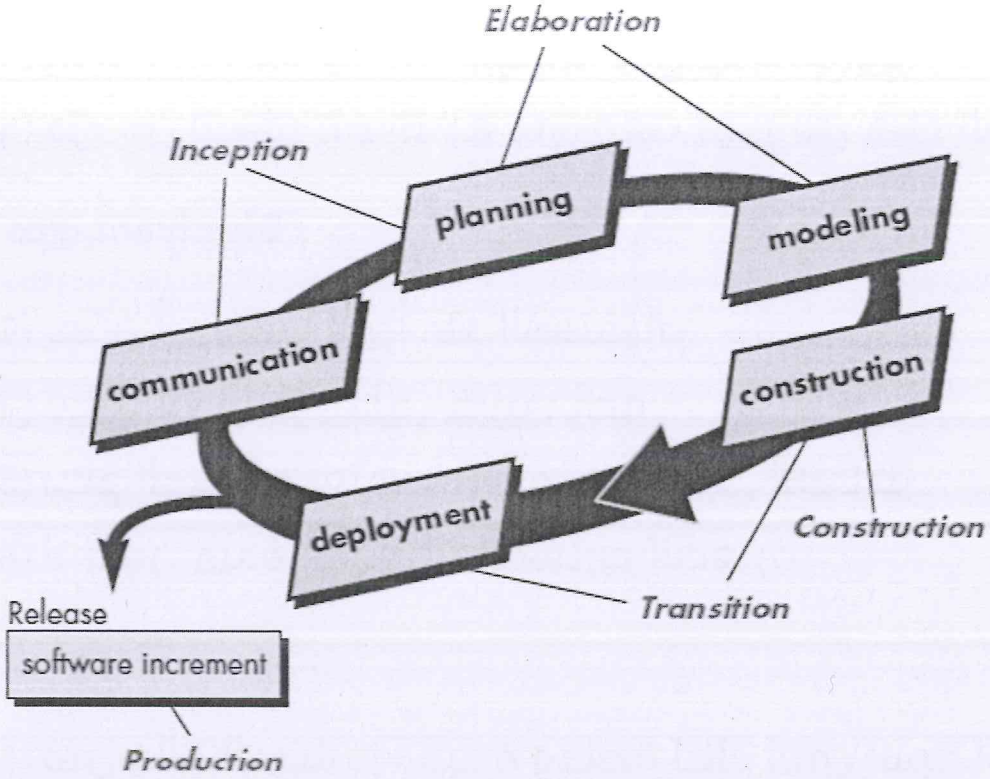
السلبيات Disadvantages :

1) يمكن أن تكلف كثيراً عند الاستخدام.
2) تحليل المخاطر يتطلب خبرة خاصة.
3) نجاح المشروع يعتمد على مرحلة تحليل المخاطر.
4) لا تكون جيدة في حالة المشاريع الصغيرة.

استخدام الـ Spiral Model :

1) في حالة المشاريع الكبيرة وخاصة لتلك التي نسبة الفشل فيها عالية.
2) عندما تكون عملية التحقق من المخاطر والكلفة مهمة.
3) في حالة المشاريع المتوسطة إلى العالية الخطورة.
4) الالتزام بالمشاريع طويلة الأمد غير جديرة بالثقة بسبب التغيرات المتوقعة للأولويات الاقتصادية.
5) المستخدمون غير متأكدون من حاجاتهم.
6) عندما تكون المتطلبات معقدة.
7) في حال وجود خط انتاج جديد.
8) التغيرات العلمية المتوقعة (كالبحوث والاستكشافات).

❖ النموذج الموحد :The Unified Process Model



تم توحيد جميع مزايا الـProcess Model السابقة وجمعها في Process Model واحدة هي الـ Unified Model التي تعتمد على الـUML وهدفها هو إنتاج منتج بجودة عالية.

يعتمد الـUP على الـIncremental product فهو تزايدي incremental ودوري iterative.

أساس الدورية في الـUnified Model هو الـworkflows والتي هي على التمثيل:

(Planning-Analysis-Design-Testing-Support)

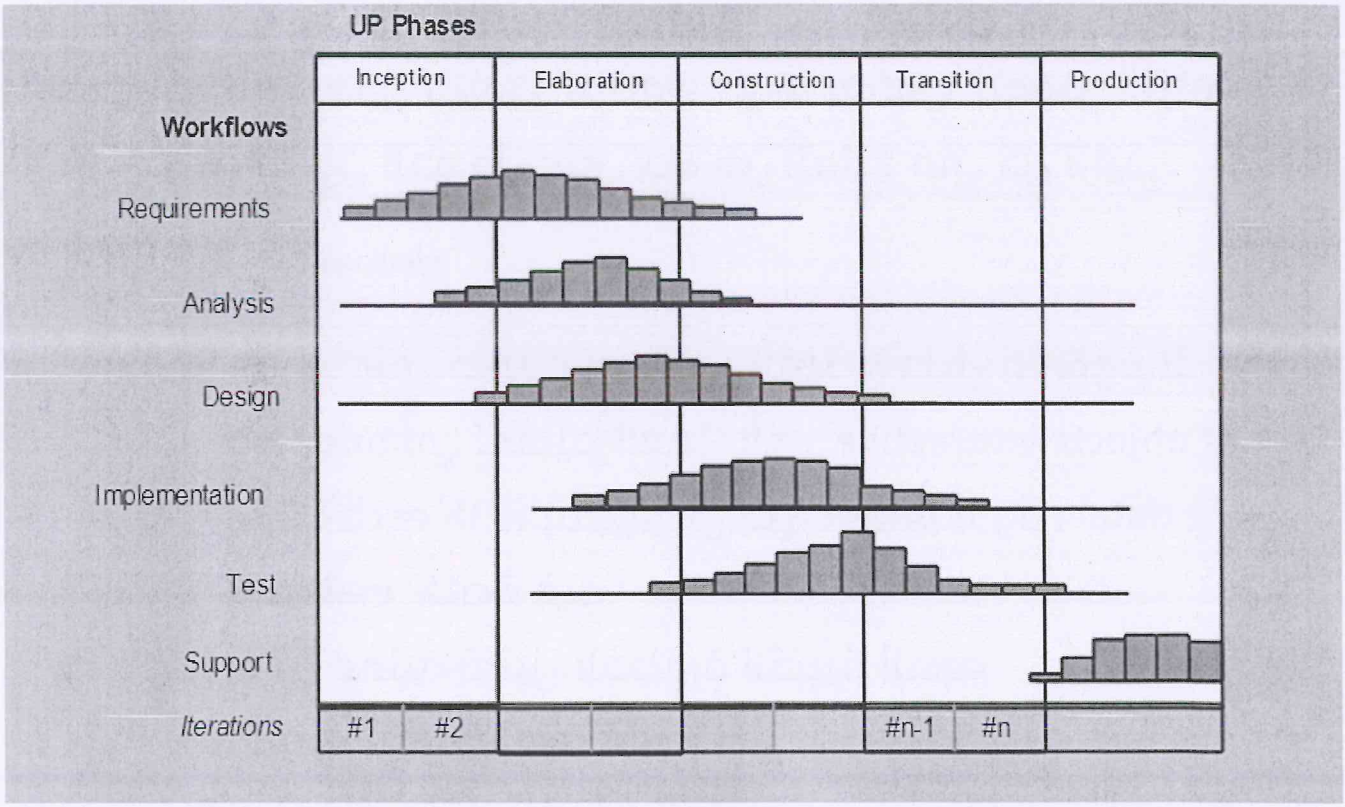
وحتى ننتهي من **milestone** واحدة يمكننا أن نقوم بالدوران على الـ **workflows** عدة مرات في كل طور من أطوار نموذج الاجرائيات الموحد **milestones** والتي هي:

Inception, Elaboration, Construction and Transition

الطور البدائي, الطور التوضيحي, الطور البنائي و الطور التوظيفي.

بحيث تحدد عدد الدورات بحسب حجم المشروع.

والسؤال المهم هنا ما العلاقة الرابطة بين UP work flows وبين UP (Milestones) phases ؟



نلاحظ من الجدول أنه على الرغم من أن كل طور يتضمن عدة دورات من **workflows** ليتم إنجازه ولكن في الحقيقة بعض الاعمال المتضمنة في

workflows لا يتم استدعاءها أو إنجازها حسب الطور الموجودة ضمنه.
فعلى سبيل المثال يحتاج طور inception لدورتين كاملتين من workflows ليتم إنجازه ولكن مراحل العمل المنجزة في كل دورة هي فقط requirements والقليل من الـanalysis والـdesign والباقي يهمل...

ومثلاً عند الـelaboration نقوم بالـrequirements ونكثر من analysis والـdesign ونبدأ بالـ implementation و الـtesting...

أي تأثير الـWorkflow نسبي على الـMilestones.

وبالتالي نحن هنا بحاجة الى عدة دورات قبل الحصول على احد الإصدارات التي يمكن للزبون الحصول عليها بعكس النموذج التزايدي الذي يضمن للزبون الحصول على جزء من المنتج في كل تزايد.

الإيجابيات Advantages:

1) يعتمد على طريقة تصميم وتنفيذ وتحليل مفهوم الـ object oriented أي تقسيم العمل إلى set of objects interactive with each other بمعنى أن كل object يحوي data و methods خاصة فيه.
2) يدعم عملية تحليل الخطورة الكبيرة الحجم.
3) يضمن النتائج بجودة عالية.
4) استخدام فعال للموارد المتاحة.
5) القدرة على اكتشاف القضايا في وقت باكر من المشروع وبالتالي يمكن تسليم أجزاء من المشروع للمستخدم بانتظام أي يبقى أصحاب المصحلة على تواصل دائم.

6) يمكن أن تتغير لاستيعاب الحالات المختلفة.

السلبيات Disadvantages :

1) تكلفة عالية و زمن كبير لأنها عملية ضخمة.
2) لايمكن أن ينفذ في حالة المشاريع الصغير والغير **object oriented**.
3) يمكن أن تكون عملية معقدة جداً.
4) عدم القدرة على التحكم بعملية التطوير.
5) تحتاج إلى خبرة جيدة للتعامل معها.

استخدام الـ Unified Model:

1) في حالة المشاريع الكبيرة.
2) مناسبة في حالة المشاريع التي تدعم OOP.
3) نستطيع استخدامه في جميع أنواع المشاريع.

(embedded-web development-desktop application...)

- النهاية -