# تكليف طلاب المجموعة الأولى

1/عبدالله محمد عبدالرحمن الشريف

2/حسين ألاشول

3/عبدالله طيب جرشاب

4/علي صالح الوهاشي

5/صالح علي عيظه

تطوير التطبيقات هو عملية تتضمن العديد من الخطوات المختلفة، بدءًا من تحديد متطلبات المشروع، وحتى اختبار الجودة والصيانة بعد الإطلاق. توجد العديد من المنهجيات التي تم تصميمها لتوجيه هذه العملية وتسهيل الإدارة والتنفيذ. هنا نستعرض بعضًا من أهم منهجيات تطوير التطبيقات.

## أهم منهجيات تطوير التطبيقات:

المنهجيات في تطوير التطبيقات تعد أساسية لتحقيق النجاح والإنتاجية. إليك بعض من أهم المنهجيات التي تستخدم في تطوير التطبيقات:

- **منهجية Agile:**
  يتيح التطوير اللين للفرق القدرة على التكيف مع التغييرات أثناء العملية والاستجابة بشكل أسرع لمتطلبات العملاء. تعتمد Agile على التحسين المستمر والتعاون الفريق الوثيق، مع تقسيم العمل إلى مراحل أو "sprints".
- **منهجية Waterfall:**
  هذه هي منهجية تطوير البرمجيات التقليدية، وتتكون من مراحل متعددة تمر بترتيب ثابت، بدءًا من تحديد المتطلبات وحتى الصيانة.
- **منهجية Scrum:**
  منهجية فرعية لـ Agile، تركز على تقسيم العمل إلى أجزاء صغيرة يمكن إدارتها وإكمالها في فترات زمنية قصيرة تسمى "sprints".
- **منهجية Lean:**
  تتركز على تقليل الهدر والزيادة في القيمة التي يتلقاها العملاء، من خلال التركيز على الأنشطة المهمة فقط وتحسين الكفاءة.
- **منهجية DevOps:**
  تتركز على التعاون المستمر بين فرق التطوير والتشغيل، لتحقيق التسليم المستمر والجودة العالية وتحسين الاستجابة للتغيير.
- **التطوير القائم على الخدمات (Service-Oriented Architecture SOA):**
  يتركز على بناء التطبيقات كمجموعة من الخدمات المستقلة التي يمكنها التفاعل مع بعضها البعض.

كل منهجية لها نقاط قوة وضعف خاصة بها، والخيار الأمثل يعتمد على متطلبات المشروع والبيئة العاملة والموارد المتاحة.

## كل ما يخص منهجية Agile

منهجية Agile، أو النهج اللين، هي مجموعة من المبادئ التي تركز على التوصيل السريع للقيمة للعميل، التعاون المكثف، التعلم المستمر، والقدرة على التكيف مع التغيير.
الركيزة الرئيسية للتطوير اللين تتلخص في القيم الأربعة المعرّفة في "دستور التطوير اللين للبرمجيات":

- **الأفراد والتفاعلات أكثر من العمليات والأدوات.**
- **البرمجيات العاملة أكثر من الوثائق التفصيلية.**
- **التعاون مع العميل أكثر من التفاوض على العقود.**
- **الاستجابة للتغيير أكثر من اتباع خطة.**

**هناك العديد من الممارسات الشائعة التي ترتبط بمنهجية Agile، بما في ذلك:**

- **التكامل المستمر (Continuous Integration): تتضمن ممارسة التحديثات المتكررة للبرمجيات، مما يسمح بالاكتشاف المبكر للمشكلات والأخطاء.**
- **التسليم المستمر (Continuous Delivery): تتضمن التحقق الدوري من البرمجيات للتأكد من أنها جاهزة للتسليم في أي وقت.**
- **Scrum: هو إطار عمل تكراري ومتزايد يستخدم في تطوير البرمجيات وإدارة المنتجات.**
- **Kanban: هي منهجية تركز على تحسين الكفاءة عبر التحسين المستمر والبصرية.**
- **تطوير البرمجيات التي تدفعها الاختبارات (Test-Driven Development, TDD): يتضمن كتابة الاختبارات قبل الكود الفعلي.**

**الجدير بالذكر أن منهجية Agile هي أكثر من مجرد مجموعة من الممارسات والأدوات، بل هي ثقافة تعتمد على التعاون، التحسين المستمر، والاستجابة للتغيرات.**

**المبادئ الرئيسية للتطوير اللين تشمل التالي:**

- **التغييرات هي جزء طبيعي ومرحب به: يتكيف نهج Agile مع التغييرات بدلاً من محاولة تجنبها، مما يتيح للفرق الاستجابة بفعالية للتغييرات في المتطلبات أو الظروف.**
- **التسليم المستمر والتحسين المستمر: يتم تقسيم العمل إلى مراحل أو "سبرنتس"، مع السعي لتحقيق التقدم والتحسين مع كل سبرنت.**
- **التعاون والاتصال: التطوير اللين يشجع على التعاون الوثيق بين جميع أعضاء الفريق وأصحاب المصلحة، بما في ذلك العملاء والمستخدمين.**
- **التركيز على القيمة التي يتلقاها العملاء: يهدف Agile إلى تحقيق أقصى قدر من القيمة بأسرع ما يمكن، من خلال التركيز على الأنشطة ذات الأولوية العالية وإيقاف الأعمال غير الضرورية.**

**النهج Agile هو ليس فقط مجموعة من الممارسات والأدوات، بل هو أيضًا ثقافة وطريقة تفكير تتطلب التزامًا بالمرونة، التعاون، والتحسين المستمر.**

## كل ما يخص منهجية Waterfall

**منهجية Waterfall، أو المنهجية المتتالية، هي أحد أقدم نماذج تطوير البرمجيات والتي تم البناء عليها في العديد من المنهجيات الأخرى. تتميز بتقديمها الخطي والتسلسلي، حيث يتم إكمال كل مرحلة بالكامل قبل الانتقال إلى المرحلة التالية.**
**تتألف منهجية Waterfall من مراحل متعددة، تشمل عادة الآتي:**

- **جمع المتطلبات: في هذه المرحلة، يتم تحديد متطلبات المشروع وكتابتها في مستند مواصفات مفصل.**

- **تصميم: يتم في هذه المرحلة تصميم البرمجيات تحديد كيف سيتم تنفيذ المتطلبات.**
- **تنفيذ أو البرمجة: في هذه المرحلة، يتم العمل على كتابة الكود وفقاً للتصميم.**
- **اختبار: يتم في هذه المرحلة اختبار البرمجيات للكشف عن الأخطاء والتأكد من أن البرمجيات تعمل كما هو متوقع.**
- **التثبيت: تتم في هذه المرحلة تثبيت البرمجيات على أنظمة المستخدمين.**
- **الصيانة: في هذه المرحلة، يتم القيام بأي تحديثات أو تعديلات ضرورية بناءً على المشاكل أو القضايا التي يمكن أن تواجهها المستخدمين.**

**تعد منهجية Waterfall أسلوباً جيداً للمشاريع التي تتميز بمتطلبات واضحة وثابتة، حيث لا يتوقع حدوث تغييرات كبيرة أثناء العملية. ولكنها قد تكون غير فعالة في بيئات المشروعات الديناميكية حيث التغييرات متكررة، مثل معظم مشاريع تطوير البرمجيات الحديثة.**

## المراحل في منهجية الشلال تتضمن عادةً:

- **التحليل المتطلبات: يتم في هذه المرحلة جمع المتطلبات من العملاء والمستخدمين وتوثيقها بشكل دقيق.**
- **التصميم: يتم تحويل المتطلبات إلى تصميم نظام البرمجيات، والذي يصف كيفية تنفيذ المتطلبات.**
- **التنفيذ: في هذه المرحلة، يتم العمل على بناء النظام بناءً على التصميم باستخدام لغات البرمجة المناسبة.**
- **الاختبار: يتم اختبار النظام للتحقق من أنه يعمل كما هو متوقع و يستجيب للمتطلبات المحددة.**
- **التثبيت والتنفيذ: تتضمن هذه المرحلة تثبيت البرنامج في بيئة العميل والبدء في استخدامه.**
- **الصيانة: في هذه المرحلة، يتم التعامل مع أي مشكلات تظهر بعد التثبيت وتقديم التحديثات اللازمة.**

**يتميز نموذج الشلال بسهولة فهمه وإدارته، لأن كل مرحلة يجب أن تكتمل قبل الانتقال إلى المرحلة التالية. ومع ذلك، فهو قد لا يكون مثاليًا لمشاريع تتطلب التعديلات الكثيرة والسريعة، حيث يتطلب تغيير المتطلبات العودة إلى الوراء لإعادة التصميم والتنفيذ والاختبار، مما يمكن أن يكون عملية مكلفة وبطيئة.**
**منهجيات تطوير البرمجيات هو عمل مهمة توجه فرق التطوير لإنتاج برمجيات ذات جودة عالية وفقًا لجدول زمني محدد. منهجيات مثل الشلال واللين (Agile) هي مثالين بارزين على هذه الأطر، كل منهما يوفر نهجًا فريدًا لتطوير البرمجيات يعتمد على مجموعة من القيم والممارسات.**

**منهجية الشلال تتميز بتسلسلها الخطي والمحدد مسبقا، والذي يمكن أن يكون مثاليًا للمشروعات ذات المتطلبات الثابتة والواضحة، لكنه قد يفشل في التعامل مع المشروعات المعقدة والديناميكية التي تتطلب مرونة أكبر.**
**من ناحية أخرى، منهجية اللين (Agile) تقدم مرونة أكبر وتكيفية مع التغيير، مما يجعلها مثالية للمشروعات التي تتطلب التكيف المستمر مع المتطلبات الجديدة أو المتغيرة. ولكن، يمكن أن يكون تطبيقها تحديا، خاصة في البيئات التي تفتقر إلى الثقافة اللازمة للتعاون والتحسين المستمر.**

**في النهاية، لا يوجد نهج "أفضل" واحد لتطوير البرمجيات. النهج الأمثل يعتمد على مجموعة من العوامل، بما في ذلك طبيعة المشروع، متطلبات العميل، مهارات وخبرات فريق التطوير، و ثقافة الشركة. تحديد المنهجية الأمثل يتطلب تقييمًا دقيقًا لهذه العوامل والتوازن بين المرونة والتحكم، السرعة والجودة، والابتكار والاستقرار.**