

الكلية التطبيقية

السنة الرابعة- تقنيات

إدارة مشاريع نظم المعلومات القسم الأول

مدرس المقرر:
أيهم محمود الحميد

المرجع: الحسين، جمال، (2018)، منشورات الجامعة الافتراضية السورية.

الفهرس

١	مقدمة في إدارة المشاريع
٢	تعريف المشروع
٣	مفاهيم أساسية
٤	خصوصية مشاريع تطوير البرمجيات
٤	عوامل نجاح وأسباب فشل المشاريع
٥	مهارات إدارة المشاريع
٦	العلاقة بين الجودة والكلفة والتسليم
٧	إطار عمل إدارة المشاريع
٨	مجالات المعرفة المتعلقة بإدارة المشاريع
٩	أطوار إدارة مشاريع تطوير البرمجيات
١١	مفاهيم لإدارة المشروع
١٢	دورة حياة المشروع ودورة حياة المنتج
١٣	تدريبات
١٤	الإجابة الصحيحة
١٥	دورة حياة المنتج البرمجي
١٧	دورة حياة المنتج البرمجي
١٧	منهجية الشلال
١٩	منهجية V
٢١	منهجية النمذجة المتعاقبة
٢٢	منهجية التطبيق السريع
٢٤	دورة حياة المنتج التراكمية
٢٦	النموذج الحلزوني
٢٧	منهجية آجيل
٢٨	تدريبات
٣٠	الإجابة الصحيحة
٣٠	المراجع
٣١	منهجية التطوير سكرم (١)
٣٣	مقدمة

٣٤	سكرم بديل عن المنهجيات التقليدية في إدارة مشاريع تطوير البرمجيات
٣٥	عناصر السكرم
٤٢	مقارنة بين سكرم والمنهجيات التقليدية لتطوير البرمجيات
٤٤	تدريبات
٤٥	الإجابة الصحيحة
٤٥	المراجع
٤٦	منهجية التطوير سكرم (٢)
٤٨	مقدمة
٤٨	اجتماعات سكرم
٥٨	ورقيات سكرم
٦١	سكرم مع عدة فرق تطوير
٦٢	منهجيات أخرى معتمدة في آجايل
٦٢	المراجع
٦٣	تدريبات
٧٦	الجدولة الزمنية للمشاريع
٧٨	دورة PDCA في إدارة المشاريع
٧٩	الجدولة الزمنية للمشاريع
٨٠	إدارة الجدولة الزمنية للمشروع
٨٠	المخطط الشبكي للمشروع
٩١	الجدولة الزمنية للمشروع في آجايل
٩٢	تدريبات
٩٤	الإجابة الصحيحة
٩٥	إدارة الكلفة
٩٧	تكاليف المشاريع
٩٧	تصنيف الكلف في المشاريع البرمجية
٩٨	القياسات الممكنة في مشاريع تطوير البرمجيات
١٠٢	الطرق الأساسية لتقدير الكلفة في مشاريع تطوير البرمجيات
١٠٣	دقة تقدير كلفة البرمجيات
١٠٤	إدارة الكلفة
١٠٥	أساسيات نموذج COCOMO
١٠٧	نمذجة الجهد

١٠٩	تدريبات
١١٠	الإجابة الصحيحة
١١٠	المراجع
١١١	إدارة المخاطر
١١٣	إدارة المخاطر
١١٣	مفاهيم أساسية
١١٦	خطة إدارة المخاطر
١١٧	تحديد المخاطر
١٢٣	تقييم المخاطر
١٢٩	وضع الاجراءات المضادة ومراقبة المخاطر
١٣١	تدريبات
١٣٢	الإجابة الصحيحة
١٣٣	التوثيق – ادارة الجودة - الدروس المستفادة
١٣٥	التوثيق
١٣٨	جودة المنتج
١٤٠	الدروس المستفادة
١٤١	تدريبات
١٤٢	الإجابة الصحيحة
١٤٢	المراجع



الفصل الأول: مقدمة في إدارة المشاريع

كلمات المفتاحية:

متطلبات الزبون، مدير المشروع، مشروع، منهجيات إدارة المشاريع، دورة حياة المنتج البرمجي، أصحاب المنفعة من المشروع، إطلاق المشروع، إدارة الجودة، إدارة المخاطر، إدارة الموارد، إدارة المشتريات، مهارات إدارة المشاريع، مجالات المعرفة في إدارة المشاريع، إجراءات إدارة المشاريع.

User requirements, Project manager, Project, project management methodologies, cost management, quality management, risk management, SDLC, project initiation, stakeholders

ملخص:

في هذا الفصل تعريف للمشروع، ومفاهيم إدارة المشاريع بشكل عام، والبحث في خصوصية مشاريع تطوير البرمجيات، وطرق إدارتها.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- شرح مفهوم المشروع وإدارة المشروع
- شرح العوامل الأساسية لنجاح وفشل المشاريع
- توضيح أهمية الإدارة في المشاريع البرمجية
- شرح إطار عمل إدارة المشاريع
- شرح مفهوم دورة حياة المشروع
- شرح مفهوم إجراءات إدارة المشاريع وبنية هذه الإجراءات

1. تعريف المشروع:

المشروع هو "جهد فردي أو عمل جماعي مخطط له مسبقاً لتحقيق هدف معين في اطار زمني محدد".

أهم خصائص المشروع هي:

- له مدة زمنية محددة
 - غايته منتج مميز ويختلف عن غيره من المنتجات
- ومن خصائصه:

- له تاريخ بدء وتاريخ نهاية
- له معايير جودة يجب أن يحققها
- له ميزانية ويجب انجازها ضمن حدود هذه الميزانية
- مخطط له مسبقاً

2. مفاهيم أساسية:

إدارة المشاريع:

هي تطبيق المعارف والمهارات والتقنيات والأدوات المتاحة، على أنشطة المشروع لتحقيق غايته المعرفة من قبل أصحاب المصلحة (Stakeholders).

مدير المشروع (Project manager):

هو الشخص المسؤول عن توجيه نشاطات المشروع، والاستفادة من كافة الموارد المالية والبشرية، لتحقيق غاية المشروع ضمن الشروط الزمنية وحدود الكلفة المقدرة سلفاً.
يبني هدف المشروع على متطلبات المستخدم (User requirements).

المهتمون بالمشروع (Project Stakeholders)

هم أشخاص يهتمون أو يتأثرون بنشاطات المشروع (الإدارة، راعي المشروع، فريق العمل، كادر الدعم الفني، الزبائن، المستخدمين، الموردين، خصوم المشروع).

و على إدارة المشروع أن تأخذ الوقت الكافي لتحديد وفهم وإدارة العلاقات مع جميع المهتمين بالمشروع.

3. خصوصية مشاريع تطوير البرمجيات:

مشاريع تطوير البرمجيات هي خالة خاصة من المشاريع عموماً، وتتبع نفس قواعد الادارة ومنهجيات العمل، إلا أنها تتميز بـ:

- صعوبة التحقق من انجازها (تم تحقيق جميع الوظائف المطلوبة)، وللتحقق يجب اختبار كامل الوظائف و جميع السيناريوهات المحتملة لها
- عدم وضوح المتطلبات في المراحل المبكرة من بناء المشروع (مرحلة جمع المتطلبات أو التحليل)، وهذا يقود إلى أخطاء صعبة التدارك في المراحل المتأخرة من المشروع، وهنا تتميز بعض طرق إدارة مشاريع تطوير البرمجيات عن غيرها، كما سنرى في الفصول التالية (منهجية سكرم – الفصل 2)
- صعوبة التحقق وقياس جودة المنتج، وقد يتطلب التحقق من بعض الميزات وقتاً طويلاً، وقد لا تظهر بعض الثغرات إلا بعد فترة تشغيل طويلة للمنتج البرمجي
- صعوبة مراقبة وقياس التقدم في انجاز المشروع، فبعض ميزات المنتج البرمجي تتطلب حلولاً إبداعية، قد تنجز في وقت قصير وقد تتطلب وقتاً أطول، ومن الصعب تحديد ذلك سلفاً
- سرعة التطور التكنولوجي، فقد يؤدي ظهور تقنيات وأدوات جديدة خطراً على المشروع، وقد يضطرنا ذلك لإعادة بناء المشروع بأدوات وتقنيات مختلفة

والإدارة الجيدة للمشاريع البرمجية هي مفتاح نجاح المشروع بكامله، وهي صعبة جداً وتتطلب إدارة على ثلاث مستويات: إدارة فريق العمل، إدارة المشكلة وإدارة إجراءات العمل:

- **إدارة المشكلة:** وتعني تعريفاً دقيقاً للمسألة المراد حلها وهي هنا غاية المشروع وأهدافه المرحلية.
- **إدارة الأشخاص:** وهم غالباً فرق عمل يجب التنسيق بينها، والاستفادة من مهارات وأعضائها.
- **إدارة إجراءات العمل:** والمقصود هنا اتباع منهجية إدارة تسمح بمراقبة وقياس تطور العمل.

4. عوامل نجاح وأسباب فشل المشاريع:

عوامل نجاح المشاريع بشكل عام

- التزام ودعم الإدارة العليا
- تحديد المهتمين بالمشروع
- معرفة وتحقيق توقعات المهتمين بالمشروع
- غاية معلنة وخطة جيدة للقيام بالمشروع
- ثقافة بناءة موجّهة نحو الهدف
- فريق تقني مختص
- فريق فعّال وملتزم
- تواصل جيّد
- الثقة

ويضاف إليها في حالة مشاريع تطوير البرمجيات:

- التركيز على المستخدم
- مدير مشروع ذو خبرة
- متطلبات واضحة
- نطاق مصغر
- بنية برمجية معيارية
- متطلبات ثابتة
- منهجية صورية (Formal Methodology)
- تقديرات صحيحة لحجم العمل وكلفته الزمنية

وتفشل المشاريع للأسباب التالية:

- فريق المشروع هو الوحيد المهتم بالنتيجة
- ضياع المسؤولية
- لا توجد بنية مشروع
- تفتقر الخطة إلى التفاصيل
- استراتيجية خاطئة لاتخاذ القرارات المتعلقة بالمشروع
- ميزانية و/أو موارد لا يمكن الاعتماد عليها
- نقص في التواصل
- الابتعاد عن الهدف الأساسي للمشروع
- عدم متابعة المشروع وفقاً للخطة الموضوعية

5. مهارات إدارة المشاريع:

يحتاج مدير المشروع إلى العديد من المهارات، فعليه أن يكون متكيفاً مع التغيير، وأن يفهم المنظمة التي يعمل فيها أو معها، وأن يكون قادراً على قيادة الفريق نحو تحقيق غاية المشروع.

يحتاج مدير المشروع إلى المهارات بنوعها المهارات القاسية (Hard Skills) والمهارات الناعمة (Soft Skills):

- **المهارات القاسية**
 - المعرفة المتعلقة بالمنتج والإجراءات والمنهجية المتبعة
 - معرفة كيفية استخدام أدوات وتقنيات إدارة المشاريع المختلفة

• المهارات الناعمة

يتمحور المشروع (وإدارة المشروع) حول الأشخاص وعمل الفريق، من يقوم بهذه المهمة؟ من يتولى هذه المخاطرة؟ من هو المهتم أو المتأثر بهذا الأمر؟. مما يُظهر أهمية مهارات التعامل مع الأشخاص (Interpersonal Skills)، مثل القدرة على التأثير والتفاوض والنقاش. وبشكل عام يمكن تصنيف المهارات الناعمة إلى:

- مهارات التواصل (Communication Skills)
- مهارات التنظيم (Organizational Skills)
- مهارات بناء فريق (Team Building Skills)
- مهارات القيادة (Leadership Skills)
- مهارات التكيف/التغلب مع/على المشكلات (Coping Skills)

6. العلاقة بين الجودة والكلفة والتسليم

لتحقيق الأهداف المرجوة من المشروع لا بد من التركيز على ثلاثة أمور أساسية، وهي الجودة والكلفة والتسليم والتي تربطها علاقات قوية تتطلب إدارتها بشكل متوازن.

• الجودة – الكلفة

إن توظيف مهندسين خبراء لبناء برمجيات ذات جودة عالية سيزيد من قيمة الرواتب، كما أن تطبيق الاختبارات المناسبة لتحسين الجودة يؤدي إلى إطالة فترة الاختبار، وهذا بالنتيجة سيزيد تكاليف الإنفاق على الموظفين حتى وإن كان الراتب نفسه لجميع المهندسين (توظيف مهندسين يتمتعون بنفس المستوى التقني).

• الجودة – التسليم

عندما تكون فترة التطوير أقصر تكون تكاليف الإنفاق على الموظفين أقل، إلا أن فترة التطوير القصيرة ستحول دون إجراء الاختبار على نحوٍ مناسب وبالتالي تتأثر الجودة، وسيطلب الأمر توظيف مهندسين خبراء لإنهاء العمل بأقصى سرعة بدون التأثير على الجودة، مما سيؤدي إلى زيادة الكلفة الكلية.

• التسليم – الكلفة

قد يؤدي تقصير فترة التطوير إلى انخفاض تكاليف الإنفاق، ولكن في هذه الحالة ستتفد المهام على نحوٍ غير مناسب، مما يؤثر على الجودة.

7. إطار عمل إدارة المشاريع

يشكّل إطار عمل إدارة المشاريع (Project Management Framework) البنية الأساسية لفهم إدارة المشاريع، ويتكون من:

• سياق إدارة المشاريع (Project Management Context)

يصف البيئة التي يشغّل فيها المشروع، ويشمل:

- أطوار المشروع ودورة حياته
- المهتمّين بالمشروع (Stakeholders)
- تأثيرات تنظيمية
- مهارات أساسية عامة في الإدارة
- التأثيرات الاجتماعية-الاقتصادية (Socioeconomic)

• إجراءات إدارة المشاريع (Project Management Processes)

تصف على نحوٍ عام، كيف تتفاعل الإجراءات المختلفة لإدارة المشاريع التي تُنجز عادةً من قبل أشخاص ذوي مهارات معينة. تُصنّف هذه الإجراءات إلى:

- إجراءات إدارة المشروع: تهتم بوصف وتنظيم عمل المشروع
- إجراءات موجّهة نحو المنتج: تهتم بتوصيف وبناء منتج المشروع

قام معهد إدارة المشاريع (Project Management Institute) بتطوير إطار عمل عام لأي مشروع، بحيث يجري كل شيء ضمن إجراء ما (Process)، ويتبع كل إجراء إلى واحد من خمس مجموعات إجراءات وإلى واحد من تسعة مجالات معرفة (Knowledge Area).

• مجموعات إجراءات إدارة المشاريع (Project Management Process Groups):

- إجراءات الإطلاق (Initiating Processes)
- إجراءات التخطيط (Planning Processes)
- إجراءات التنفيذ (Executing Processes)
- إجراءات التحكم (Controlling Processes)
- إجراءات الإنهاء (Closing Processes)

- مجالات المعرفة المتعلقة بإدارة المشاريع
- تصف مجالات المعرفة ما هي الكفاءات والمؤهلات التي على مدير المشروع تطويرها.
- أربعة مجالات معرفة تؤدي غايات محددة للمشروع (إدارة النطاق، الوقت، الكلفة، الجودة)
- أربعة مجالات معرفة تسهل تحقيق غايات المشروع (إدارة الموارد البشرية، التواصل، المخاطر، المشتريات)
- مجال معرفة واحد (إدارة تكامل المشروع) يؤثر ويتأثر بكل مجالات المعرفة الأخرى

8. مجالات المعرفة المتعلقة بإدارة المشاريع

يُعرّف معيار إدارة المشاريع (Project Management Body Of Knowledge, PMBOK) تسعة عناصر على أنها مجالات المعرفة المتعلقة بإدارة المشاريع:

- إدارة التكامل (Integration Management)
- إجرائية لا بدّ منها لإنجاز مختلف عناصر المشروع مع الحفاظ على التكامل بينها، وتتكون من: تخطيط المشروع، تحقيق المشروع، تكييف المشروع.
- إدارة الجودة (Quality Management)
- وهي الإجرائية التي تحقّق الاحتياجات التي تم إطلاق المشروع من أجلها، وتتكون من: تخطيط الجودة، ضمان الجودة، ضبط الجودة.
- إدارة الكلفة (Cost Management)
- وهي الإجرائية التي تمكّن من إنهاء المشروع دون تجاوز الكلفة المحدّدة، وتتكون من: تخطيط الموارد، تقدير الكلفة، وضع الميزانية.
- إدارة الوقت/التسليم (Time Management/Delivery)
- وهي الإجرائية التي تمكّن من إنهاء المشروع في الموعد المحدّد أو قبله، وتتكون من: تحديد الفعاليات، تحديد تسلسل التطوير، تقدير الوقت اللازم، تحضير جدولّة المشروع وإدارتها.
- إدارة نطاق المشروع (Scope Management)
- تدير هذه الإجرائية نطاق المشروع المراد تطويره ومجال مخرجاته ومهامه، وتتكون من إطلاق المشروع، تخطيط المشروع، تعريف المشروع، التحقّق من نطاق المشروع، إدارة تكييف الألفق.
- إدارة التواصل (Communication Management)
- وهي إجرائية أساسية لبناء وتجميع ونشر وحفظ معلومات المشروع في الزمن الحقيقي، وتتكون من: تخطيط التواصل، تزويد المعلومات، إعطاء تقارير عن الأداء الحقيقي، تنفيذ إجراءات الإنهاء.
- إدارة المشتريات (Procurement Management)
- الإجرائية المتعلقة بتأمين المنتجات والخدمات من خارج فريق المشروع، وتتضمّن: تخطيط المشتريات، تخطيط الاستعلام، اختيار الموردين الذين سيقدّم لهم الطلب، إدارة العقود، إنهاء العقود.

- إدارة الموارد البشرية (Human Resources Management) وهي الإجرائية المتعلقة ببناء التنظيم والمحافظة على بقاءه واستمراره، وهي تؤثر بفعالية أكثر على الموارد البشرية المشاركة في المشروع، وتتكون من: تخطيط التنظيم، تدبير الموظفين، دعم تطوير الفريق.
- إدارة المخاطر (Risk Management) إجرائية تحديد وتقدير المخاطر المتوقع حدوثها خلال تنفيذ المشروع، إضافةً إلى تحديد الإجراءات المضادة لهذه المخاطر.

9. أطوار إدارة مشاريع تطوير البرمجيات

- إطلاق المشروع والخطة الأولية يقوم مدير المشروع بوضع خطة أولية (قبل بداية المشروع) ويعرف فيها العناصر الأساسية في المشروع و تتضمن:
 - خطة عملية لتحقيق نجاح المشروع
 - تحديد سياسة أمثلية للمشروع
 - وضع جدول زمنية، خطة للجودة، خطة للكلفة
 - تأسيس تنظيم أمثلي للمشروع بناءً على مميزات المشروع والموارد البشرية وتوضيح الصلاحيات والأدوار
- إدارة المشروع (القيادة والضبط) تنفيذ دورات PDCA (خطة عمل تحقق تصرف Plan Do Confirm Action).
 - خطط (Plan)
 - الخطوة الأولى لتخطيط المشروع هي توضيح الأهداف و حالما تُحدّد الأهداف يجري وضع خطة لتحقيق المشروع بناءً على هذه الأهداف، توضح هذه الخطة السياسية التي ستتبع وتسلسل فعاليات العمل ضمن المشروع
 - اعمل (Do)
 - إطلاق المشروع والبدء بالعمل الفعلي بحيث تنجز المهام حسب الخطة الموضوعية وليس حسب الأهواء
 - تحقق (Check)
 - المقارنة بين الخطة الموضوعية والنتائج التي تم الوصول إليها
 - تصرف (Action)
 - إيجاد حلول للمشاكل التي تم تحديدها في مرحلة التحقق بحيث يجري تحديد أسباب هذه المشاكل والتخلص منها لمنع وقوعها ثانيةً

● إنهاء المشروع (التقييم وإنشاء التقارير)

- ينتهي المشروع عندما تكتمل أطواره، وقبل إعلان انتهائه يجب التحقق من:
 - إجراء تحليل وتقييم يحدّد فيما إذا كان المنتج سيقدم إلى المستخدم في نهاية المشروع
 - إنشاء تقارير تتضمّن تفاصيل هذا التقييم وتقديمها إلى المدراء التنفيذيين (Chief Executives)، وتخزينها للاستفادة منها كمرجع هام في المشاريع القادمة
 - يتضمن هذا المرجع الخطط، بيئة التطوير، الموظفين (عددهم ومهاراتهم)، أدوات التطوير، منتجات مثل حزم البرمجيات، مهارات المهندسين التي تم الاستعانة بها من الخارج، طريقة العمل ومعلومات التكاليف

ملاحظات:

- يجب التأكد من إتمام المشروع وذلك اعتماداً على نتائج الاختبار (عدد حالات الاختبار، عدد الأخطاء، نتائج تصحيح الأخطاء)
- رفع تقرير بالنتائج إلى رئيس المشروع في قسم التطوير وكذلك إلى المستخدمين ومن ثم يجري التحضير لتشغيل المشروع
- يجب تسليم المنتجات النهائية (مثل الأجهزة، البرمجيات و/أو الوثائق) إلى المستخدمين ولكن يختلف وقت التسليم حسب المشروع
- في حال حدوث أية مشاكل بعد بدء التشغيل (أو عملية اختبار معينة) يجب وضع تقرير بحالات هذه المشاكل وبالإجراءات المضادة لها
- المعلومات الخاصة بالمشروع مفيدة جداً من أجل تحقيق المشروع في المستقبل
- من الضروري، بعد إتمام المشروع، مراجعة هذه المعلومات (سواء كانت القيم الأولية صحيحة أم لا، مثل تقدير الكلفة، الجدول الزمني، عدد عناصر الاختبار، عدد الأخطاء وغيرها)
- مراجعة طريقة تنظيم المشروع ومهارات أعضائه وطريقة تطويره
- يجب أن تؤثر النتائج في المشاريع اللاحقة

10. مفاهيم إدارة المشروع

فيما يلي بعض الملاحظات والقواعد، التي تساعد في ضمان إنهاء المشروع بشكل سليم وتحقيق الغاية منه، وهي اعتبارات لا بد لمدير المشروع بالدرجة الأولى أن يأخذ بها:

• قواعد التشغيل

- تحديد قواعد عملياتية هو أمر ضروري لإنجاز المشروع إنجازاً سليماً
- يجب أن توضع قواعد لتحقيق المشروع، كطرق تشارك المعلومات المتعلقة بالمشروع وإجراءات حلّ المشاكل التي قد تحصل وغيرها
- من المهم التأكد من فهم الجميع لهذه القواعد

• توضيح صلاحيات الموظفين المسؤولين

- عندما تحصل مشكلة معينة يتابع رئيس المجموعة العمليات الحالية من خلال اتخاذ القرارات المناسبة ضمن حدود صلاحياته ومن ثم يرفع تقريراً بالنتائج إلى مديره
- في حال كانت المشكلة خارج صلاحياته، عليه أن يطلب قرار المدير بخصوصها وقد يكون ذلك من خلال اجتماع معه

• كيفية الإنجاز

- يحصل مدراء/أعضاء المشروع من خلال الاجتماعات على معلومات هامة لإنجاز المشروع أو/و لإعطائها للمستخدمين
- يجب القيام بأنواع مختلفة من الاجتماعات كالاتماعات ضمن مجموعة لمدراء المجموعة و/أو المستخدمين، وذلك حسب حجم المشروع
- على المدراء تحديد نوع ومحتوى ومشاركين وبرنامج كل اجتماع

• توضيح تفاصيل التقارير

- من المهم تحديد صيغة للتقارير التي تُكتب حول تقدّم المشروع والمشاكل، بالإضافة إلى تحديد تفاصيل هذه التقارير

11. دورة حياة المشروع ودورة حياة المنتج

دورة حياة المنتج ودورة حياة المشروع، هما شيئين منفصلين ومستقلين تماماً، ولكن قد يحصل تقاطع بينهما بالزمن فقط

• دورة حياة المشروع (Project Life Cycle):

▪ الإِطلاق (Initiating)

▪ التخطيط (Planning)

▪ التنفيذ (Execution)

▪ الإِنتهاء (Closure)

• دورة حياة المنتج

للمنتجات دورة حياة أيضاً، تعتبر دورة حياة تطوير الأنظمة (Systems Development Life Cycle) إطار عمل لتوصيف الأطوار المتضمنة في تطوير أنظمة المعلومات (Information Systems)، وتتبع دورة حياة المنتج للمنهجية المستخدمة في تطويره، وسنغطي دورة حياة المنتج بالتفصيل في الفصول اللاحقة.

12. تدريبات:

1. دورة حياة المنتج تتألف من أربعة مراحل (الاطلاق، التخطيط، التنفيذ، الانتهاء) ؟ (اختر 1 من الإجابات)

- صح
- خطأ

2. قبل إعلان إنهاء المنتج يجب ؟ (احذف 1 من الإجابات)

- التأكد من نتائج اختبارات التشغيل
- انشاء تقارير التقييم ورفعها للمدراء التنفيذيين
- مراجعة تقارير الانتهاء وتدقيق معلوماتها (تقدير الكلفة، الجدول الزمني، عدد عناصر الاختبار، عدد الأخطاء وغيرها)
- انشاء تقارير تتضمن الخطط، بيئة التطوير، الموظفين (عددهم ومهاراتهم)، أدوات التطوير، منتجات مثل حزم البرمجيات، التكاليف وغيرها وتقديمها للمدراء التنفيذيين

3. من عوامل نجاح المشاريع ؟ (احذف 1 من الإجابات)

- مدير مشروع ذو خبرة
- متطلبات واضحة
- تقديرات صحيحة لحجم العمل وكلفته الزمنية
- التركيز على أرباح المشروع

4. من خصائص المشروع ؟ (احذف 1 من الإجابات)

- له تاريخ بدء وتاريخ نهاية
- له معايير جودة يجب أن يحققها
- له أدوات ومجموعة تقنيات يجب الالتزام بها
- له ميزانية ويجب انجازه ضمن حدود هذه الميزانية

5. من أهم الصعوبات التي تميز مشاريع تطوير البرمجيات عن غيرها من المشاريع، صعوبة التحقق وقياس

جودة المنتج ؟ (اختر 1 من الإجابات)

- صح
- خطأ

إجابة الصحيحة

1. (خطأ)

(المراحل المذكورة هي دورة حياة المشروع، أما المنتج فمراحل دورة حياته تعتمد على منهجية الإدارة المستخدمة)

2. (مراجعة تقارير الانتهاء وتدقيق معلوماتها (تقدير الكلفة، الجدول الزمني، عدد عناصر الاختبار، عدد الأخطاء وغيرها).)
(تقييم تقارير الانتهاء يتم بعد إعلان إنهاء المنتج)

3. (التركيز على أرباح المشروع)

(التركيز على مستخدمي المنتج و متطلباتهم يساعد على نجاحه أكثر من التركيز على الأرباح المتوقعة)

4. (له أدوات ومجموعة تقنيات يجب الالتزام بها.)

(قد تتغير الأدوات والتقنيات المستخدمة في المشروع أثناء تطوره، وليست بالضرورة ثابتة طيلة حياته)

5. (صح)

(وقد يتطلب التحقق من بعض الميزات وقتاً طويلاً، وقد لا تظهر بعض الثغرات إلا بعد فترة تشغيل طويلة للمنتج البرمجي)



الفصل الثاني: دورة حياة المنتج البرمجي

كلمات المفتاحية:

سكرم، آجايل، مرحلة، منهجية إدارة، دورة حياة المنتج البرمجي، منهجية حرف V، منهجية الشلال، دورة حياة المنتج التراكمية، المنهجية الحلزونية، منهجية التطبيق السريع.

Scrum, Agile, Water Fall model, V-shaped model, Spiral model, SDLC, structured evolutionary prototyping model, RAD, Incremental SDLC model, Spiral SDLC model.

ملخص:

في هذا الفصل عرض للمنهجيات الأشهر في إدارة نشاطات مشاريع بناء البرمجيات، مع عرض لميزات وسيئات كل منهجية، والبيئة المثالية لتطبيق كل من هذه المنهجيات.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- تعريف دورة حياة المنتج البرمجي
- منهجية الشلال Waterfall model
- منهجية V (V-shaped SDLC Model)
- منهجية النمذجة المتعاقبة (Structured evolutionary prototyping model)
- منهجية التطبيق السريع (Rapid application model)
- دورة حياة المنتج التراكمية (Incremental SDLC)
- منهجية Spiral
- منهجية آجايل

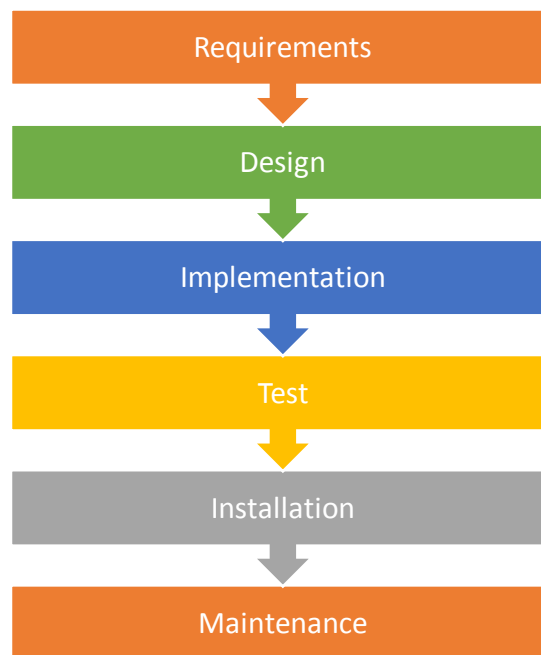
1. دورة حياة المنتج البرمجي:

دورة حياة المنتج البرمجي، هي وصف لمجموعة النشاطات في كل مرحلة من مراحل تطوير المنتج، وتختلف مراحل التطوير ومحتواها من النشاطات حسب المنهجية المعتمدة لإدارة المشروع. فيما يلي سنستعرض أهم المنهجيات المستخدمة في إدارة مشاريع تطوير البرمجيات، وسنشرح ميزات وسيئات كل منها، مع مقارنة فيما بينها وشرح للظروف الملائمة لاستخدام كل منها.

2. منهجية الشلال (Water Fall model):

تعريف المنهجية:

تعد المنهجية التقليدية في تطوير المنتج المعلوماتي (Waterfall)، على مجموعة مراحل محددة (جمع وتحليل المتطلبات، التصميم، البرمجة، الاختبار، التنصيب، الصيانة) والانتقال من مرحلة إلى أخرى يتم دون إمكانية الرجوع إلى الوراء.



(الشكل 1)

• جمع وتحليل المتطلبات:

جميع المتطلبات المحتملة والمطلوبة صراحة من النظام، يتم جمعها في وثيقة تدعى وثيقة توصيف المتطلبات (Requirements specification doc).

- **التصميم:**
في هذه المرحلة تتم دراسة المتطلبات التي تم جمعها في المرحلة السابقة، ويتم بالاعتماد عليها تصميم النظام، بما يتضمنه ذلك من بناء قاعدة المعطيات، توصيف بنية النظام ومجزئاته، العتاد المادي اللازم للتشغيل.
- **التنفيذ:**
تتم في هذه المرحلة برمجة متجزئات النظام.
- **الاختبار:**
اختبار ما تم انجازه في مرحلة التنفيذ اختبارات جزئية واختبارات كاملة (Unit testing & integration test).
- **التصويب:**
وضع النظام في بيئة الاستخدام الحقيقية.
- **الصيانة:**
تدارك ملاحظات الزبون والاطفاء التي قد تظهر نتيجة التشغيل الفعلي، ويمكن أن تتضمن هذه المرحلة أيضاً تحسينات على طرق العرض والإدخال، ويمكن أن تكون مقدمة لإنتاج اصدار جديد من النظام نفسه.

مميزات منهجية الشلال:

- سهولة الفهم والاستخدام
- تقدم نمطية في العمل تخدم الكادر الغير مؤهل بشكل كاف
- نقاط العلام فيها واضحة
- المتطلبات ثابتة بالنسبة لفريق العمل، بدءاً من مرحلة التصميم
- الادارة ومراقبة تطور العمل فيها مهمة سهلة
- مثالية في حال كانت النوعية وجودة العمل أهم من زمن التنفيذ والكلفة في المشروع

مساوئ منهجية الشلال:

- جميع المتطلبات يجب أن تكون واضحة مع انتهاء المرحلة الأولى في المشروع
- مخرجات كل مرحلة ثابتة والتعديل عليها صعب، خاصة في المراحل المتقدمة من المشروع
- قد تعطي انطباعاً خاطئاً عن تقدم العمل
- لا تتسجم مع طبيعة البرمجيات المعقدة، والتي تحتاج لحل مشاكل تقنية وتتطلب حلولاً ابداعية
- النتيجة قد تكون مفاجئة في نهاية المشروع

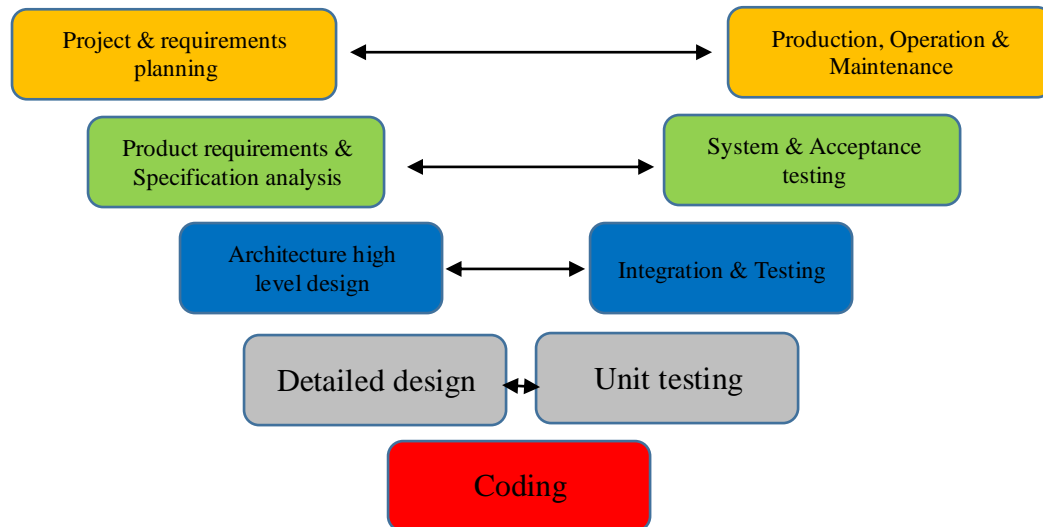
- تفاعل الزبون قليل، والفرصة غير متاحة للزبون لرؤية المنتج إلا في مراحل متأخرة، وعندها قد يكون فات الأوان على إرضائه

متى تستخدم منهجية الشلال:

- المتطلبات واضحة وثابتة
- تعريف المنتج ثابت وغير خاضع للتغيير مع الزمن
- تقنيات العمل معروفة
- المنتج المراد بناؤه هو نسخة معدلة عن منتج سابق
- المنتج المراد بناؤه هو نقل لمنتج سابق إلى بيئة عمل جديدة

3. منهجية V (V-shaped model):

هي تعديل على منهجية الشلال، حيث تضاف لكل من مراحل جمع المتطلبات والتصميم والتنفيذ، مرحلة تحقق وتأكيد على صحة النتائج، ويتم التحقق من خلال اختبارات تتم على التوازي مع كل مرحلة منها.



(الشكل 2)

هذه المنهجية مبنية على مجموعة مراحل يمكن أن تتم على التوازي:

Project and requirement planning حجز المصادر المطلوبة	Production, Operation and maintenance اجراء التحسينات والتعديلات على المنتج
Product requirement and specification analysis توصيف مفصل وكامل للنظام المطلوب	System and acceptance testing اختبار كامل النظام في بيئة العمل الحقيقية
Architecture or high-level design تعريف كيف تحقق وظائف النظام المتطلبات والتصميم	Integration testing اختبار عمل متجزئات النظام مع بعضها
Detailed design وضع خوارزميات العمل لكل مجتزئ	Unit testing التحقق من أن كل مجتزئ يؤدي الوظائف المطلوبة منه
Coding تحويل الخوارزميات إلى نصوص برمجية	

مميزات منهجية V:

- سهولة الفهم والاستخدام
- التحقق يبدأ في مرحلة مبكرة من المشروع
- كل مخرج أو مكون جزئي يخضع للاختبار
- يمكن لإدارة المشروع تتبع تقدمه من خلال نقاط علام واضحة

مساوئ منهجية V:

- ليس من السهل التعامل مع التغيير على المتطلبات
- لا تتضمن تحليل للمخاطر

متى تستخدم منهجية V:

- مناسبة للمشاريع التي تتطلب وثوقية عالية (سجلات المرضى في مستشفى مثلاً)
- جميع المتطلبات معروفة وثابتة
- بنية الحل والتقنيات المطلوب استخدامها معروفة

4. منهجية النمذجة المتعاقبة (Structured evolutionary prototyping)

:(model

- يقوم فريق العمل بوضع نموذج أولي في مرحلة جمع وتحليل المتطلبات
- يتم تقييم النموذج من قبل الزبون (المستثمر النهائي)
- يمكن للمستخدمين وضع تعليقاتهم وملاحظاتهم على النموذج الأولي
- يقوم فريق العمل بتعديل النموذج الأولي وتحسينه، آخذاً بعين الاعتبار ملاحظات الزبون
- عندما يكون النموذج ملائماً للزبون ومحققاً لمتطلباته، يتم اعتماده كتوصيف للمنتج النهائي

مرحل العمل يكن تلخيصها بما يلي:

- توضع خطة العمل في بداية المشروع
- يتم وضع النموذج الأولي (نموذج ورقي) بناءً على متطلبات الزبون
- يتضمن النموذج الأولي كافة وظائف النظام (للعرض)، دون برمجتها، وهذا النموذج هو بديل عن تحليل المتطلبات ومرجع لتوثيقها
- يقوم فريق العمل ببناء النموذج الأولي متضمناً، قاعدة البيانات، واجهات الاستخدام و وظائف التجوال بين الواجهات
- يقوم فريق التصميم بعرض النموذج (عرض حي) على الزبون، لأخذ تعديلاته واقتراحاته
- يقوم فريق التصميم بتكرار الخطوتين الأخيرتين لحين الوصول إلى موافقة الزبون، وعندها يعتبر التصميم نهائياً، ويعبر عن جميع المتطلبات وطريقة إخراجها

مميزات النمذجة المتعاقبة:

- يستطيع الزبون أن "يرى" المتطلبات كما فهمها المطورون
- التواصل جيد بين فريق التطوير والزبون
- المنتج النهائي أكثر مطابقة لتوقعات الزبون، مما تقدمه منهجية الشلال و منهجية V
- اشارات تقدم العمل واضحة
- لا توجد تعديلات مفاجئة
- التصميم المتكرر، يفسح المجال لظهور متطلبات جديدة وتحسينات تغني النظام

مساوي النمذجة المتعاقبة:

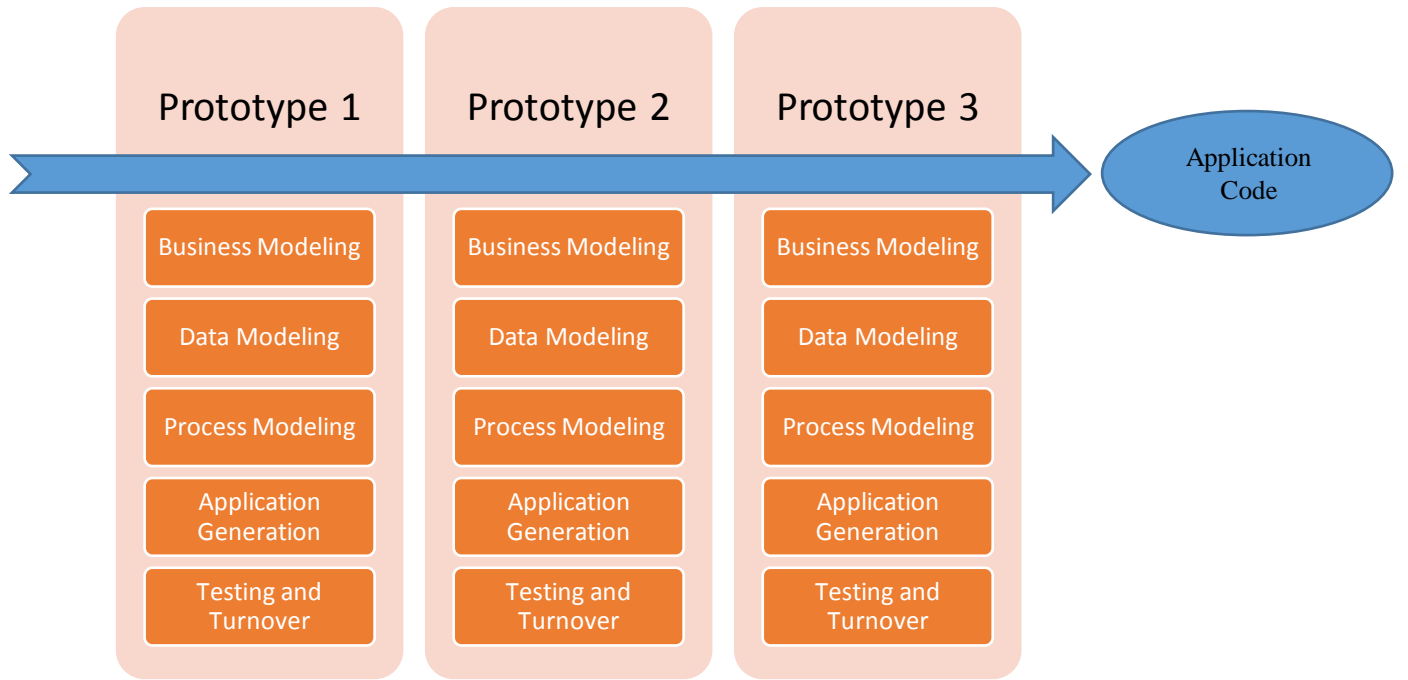
- أقرب إلى طريقة البناء والتصحيح Build and Fix في البرمجة منها للطرق الهيكلية Structured programming
- سمعة سيئة عند التطبيق العملي، بسبب التسرع في التصميم والتعديل عليه
- قد يضطر فريق التطوير إلى تعديل النموذج النهائي لأسباب تقنية، وهذا قد يزعج الزبون
- تكرار مرحلة التصميم والتحسين عليه قد يستغرق وقتاً طويلاً

متى تستخدم النمذجة المتعاقبة:

- المتطلبات غير واضحة، ويجب الاتفاق على ترجمتها مع الزبون
- يمكن مزاجنة هذه المنهجية مع منهجية الشلال، باعتماد النموذج الأولي كتحويل للمتطلبات
- عندما يكون التصميم والخراج مهم جداً، وشرط أساسي لقبول النظام
- عندما يكون المطلوب إخراج لفكرة إبداعية (نظم الدعاية والاعلان مثلاً)

5. منهجية التطبيق السريع (Rapid application model):

تعتمد هذه المنهجية على قليل من التخطيط، لصالح سرعة في انتاج النماذج، والنموذج في هذه الحالة هو منتج فعلي قابل للاستخدام لأحد متجزئات النظام. يتم بناء النماذج (متجزئات النظام) على التوازي، وتكامل لاحقاً والغاية هي سرعة انجاز النظام. وبما أن التخطيط قليل الأثر في هذه المنهجية، فالاستجابة للتعديلات والتغيرات مقبولة وسريعة. أهم ما في هذه المنهجية هو أن تكون النماذج المطورة مختبرة وقابلة لإعادة الاستخدام، ومن أهم مزاياها الاعتماد على الأدوات المؤتممة (Code generators, Screen generators, etc).



مميزات التطبيق السريع:

- دورة حياة قصيرة وعدد قليل من المطورين يعني كلفة أقل
- المخاطر أقل واكتشافها أسرع، خاصة في ما يتعلق بتجاوز الكلفة
- تفاعل جيد مع الزبون
- استخدام مفاهيم النمذجة لتحصيل المتطلبات وبنى المعطيات واجراءات العمل

مساوئ التطبيق السريع:

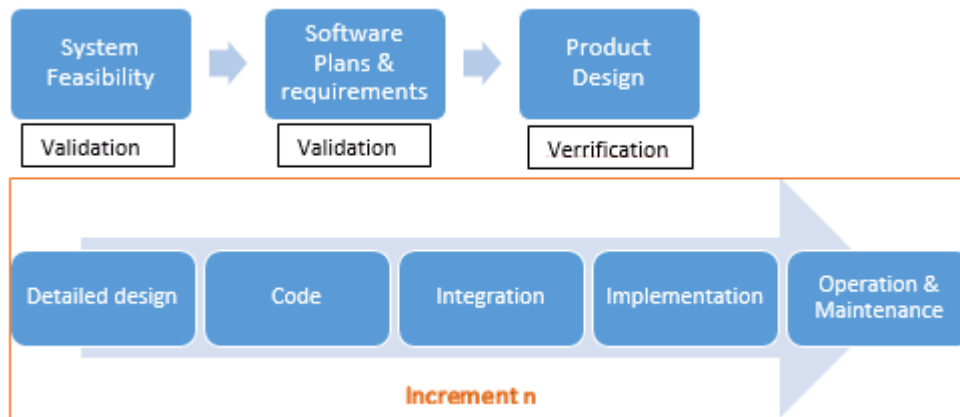
- تدخل الزبون وضرورة ابداء رأيه بعد تطوير كل اصدار، يفرض ضغطاً اضافياً على ادارة المشروع
- من الممكن عدم الوصول إلى نهاية المشروع، خاصة مع تدخل طرف خارجي (الزبون)
- صعوبة التطبيق مع النظم الموروثة (legacy systems)
- بنية المشروع يجب أن تكون قابلة للتجزئة وظيفياً، والأجزاء يجب أن تكون واضحة المعالم والحدود
- تتطلب مستو عالي من التفاهم بين الزبون وفريق التطوير

متى نستخدم التطبيق السريع:

- المتطلبات واضحة
- تدخل الزبون شرط أساسي ومطلوب
- يمكن تجزئة المشروع إلى إصدارات
- الأداء العالي ليس شرطاً
- التقنيات المستخدمة سهلة

6. دورة حياة المنتج التراكمية (Incremental SDLC):

- نبدأ بتنفيذ جزء من النظام، ثم نضيف الوظائف المتبقية بشكل تراكمي
- وظائف النظام ترتب حسب الأهمية، ونبدأ بتنفيذ الجزء الأكثر أهمية من وجهة نظر الزبون
- عند تنفيذ الوظائف، ترتب في مجموعات، وكل مجموعة تتم اضافتها تمثل إصداراً جديداً من النظام
- تتوالى الاصدارات لحين الوصول إلى اصدار يشمل جميع الوظائف المطلوبة من قبل الزبون، وهو يمثل المنتج النهائي



مميزات المنهجية التراكمية:

- الوظائف الأهم يتم بناؤها أولاً
- كل اصدار هو نسخة مصغرة من المنتج، ويمكن استثمارها
- يمكن للزبون التفاعل وأبداء الرأي مع كل اصدار
- كلفة التطوير متدرجة وتبدأ بحجم صغير
- يمكن وضع أجزاء من النظام قيد التشغيل الفعلي وبسرعة
- يحصل الزبون على ما يهمله أولاً
- أثر تغير المتطلبات والتعديل على المشروع صغير

مساوئ المنهجية التراكمية:

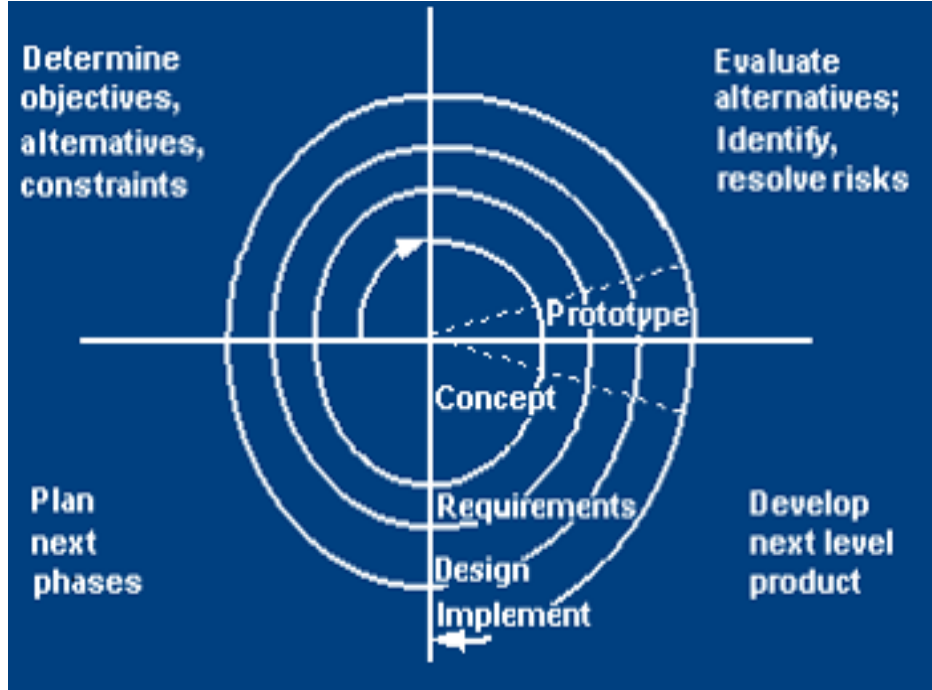
- تتطلب تخطيط وتصميم جيد
- تتطلب تعريفاً واضحاً للمتطلبات
- تعريف واضح للمتجزئات وواجهات المستخدم وواجهات الربط مع المتجزئات الأخرى
- الكلفة النهائية قد تكون كبيرة

متى نستخدم المنهجية التراكمية:

- المطلوب سرعة في التنفيذ، على الأقل لبعض أجزاء النظام
- المتطلبات واضحة مع احتمال الزيادة عليها مع الزمن
- بعض أجزاء النظام مطلوب وضعها قيد التشغيل قبل غيرها
- في المشاريع ذات الجدول الزمني الطويل للتنفيذ
- في المشاريع التي تعتمد على تقنيات جديدة ومتغيرة

7. النموذج الحلزوني (Spiral SDLC model):

هذه المنهجية هي مزيج من منهجية الشلال وتقنيات RAD وإضافة تحليل المخاطر لهما. وتعتمد على مجموعة حلقات تكرر فيها مراحل الشلال.



- تحديد الهدف (Objectives): الوظائف، الأداء، واجهات الربط البرمجية والعتاد المادي، محددات النجاح
- البدائل (Alternatives): طور، أعد استخدام، اشترى، تعاقد
- الشروط (Constraints): الجدول الزمني، الكلفة، الواجهات التخاطبية
- تقييم ودراسة البدائل (Evaluate alternatives)
- تحديد المخاطر (Identify risks)
- معالجة المخاطر (Resolve risks)
- تطوير المستوى الثاني من المنتج (Develop next level product)
- التخطيط للمرحلة التالية (Plan next phase)

مميزات النموذج الحلزوني:

- التكيف مع التغيرات في المتطلبات
- استخدام مكثف للنمذجة
- توصيف المتطلبات يتم بدقة كبيرة
- يطلع الزبون على النتائج مبكراً
- الأجزاء الأكثر خطورة يتم تنفيذها أولاً، وتقييم النتائج يتم مبكراً

مساوئ النموذج الحلزوني:

- إدارة معقدة للمشروع
- نهاية المشروع لا يمكن رؤيتها باكراً
- غير مناسبة للمشاريع الصغيرة ومحدودة الخطورة
- الاجراءات فيها معقدة
- التفاف الحلزون يكن أن يستمر إلى ما لانهاية
- المراحل الوسيطة الكثيرة تتطلب توثيقاً كثيفاً

متى نستخدم النموذج الحلزوني:

- ميزانية المشروع ثابتة لا يمكن التعديل عليها، وتقييم المخاطر ذو أهمية كبية خاصة فيما يتعلق بتجاوز الميزانية المحددة
- في المشاريع ذات عوامل الخطورة العالية
- امكانية تغير الأولويات مع تغير المتطلبات
- الزبون غير متأكد من متطلباته
- الاصدارات الأولى من المنتجات، والتي تبني عادة لأخذ تغذية راجعة من الزبائن قبل الإصدارات الأساسية من المشروع
- تغيرات متوقعة أثناء فترة حياة المشروع

8. منهجية آجيل (Agile):

تعتمد هذه المنهجية على مجموعة متعاقبة من المراحل، وكل مرحلة منها تتضمن دورة حياة منتج كاملة (تحليل - تصميم - برمجة - اختبار - تنصيب)، هذه المراحل مضبوطة زمنياً ولها مجموعة من قواعد العمل والورقيات والاجتماعات وطرق تعقب ومراقبة للإنجاز. سنقوم بشرح هذه المنهجية شرحاً مفصلاً، وسنفرد لها الفصول الأربع القادمة، لما لها من أهمية.

9. تدريبات:

1. أي من الحالات التالية مناسبة لتطبيق منهجية الشلال ؟ (اختر إجابة واحدة)

- المنتج المراد تطويره هو مسألة غير محلولة لتاريخه
- تطوير نسخة من برنامج مستخدم حالياً، تدارك مشاكل سابقة فيه
- المنتج يتطلب مستوى عالي من التنسيق بين المصمم والمبرمج
- فريق تطوير مميز بخبرته

2. في حال كان المطلوب هو مجموعة تطبيقات جزئية يجب أن توضع قيد الاستخدام بالتدرج، فإن

المنهجية المناسبة لإدارة المشروع ؟ (اختر 1 من الإجابات)

- منهجية الشلال
- منهجية V
- منهجية دورة حياة المنتج التراكمية
- منهجية RAD

3. منهجية الشلال تناسب المشاريع طويلة الأمد، أو تلك التي تتطلب تطويراً مستمراً ؟ (اختر 1 من

الإجابات)

- صح
- خطأ

4. منهجية RAD تناسب المشاريع التي تتطلب أداءً عالياً ؟ (اختر 1 من الإجابات)

- صح
- خطأ

5. النمذجة المتعاقبة هي حل مثالي لتوضيح المتطلبات المبهمة ؟ (اختر 1 من الإجابات)

- صح
- خطأ

6. في حال كان المطلوب هو منتج يبدأ بإصدار يتضمن مجموعة ميزات، يوضع قيد الاستخدام ومن ثم يتم التطوير عليه بإصدارات متتالية من خلال ملاحظات الزبائن ومتطلباتهم الجديدة، في هذه الحالة المنهجية المناسبة لإدارة المشروع هي ؟ (اختر 2 من الإجابات)

- منهجية آجايل
- منهجية V
- منهجية دورة حياة المنتج التراكمية
- منهجية الشلال

الإجابة الصحيحة:

1. (تطوير نسخة من برنامج مستخدم حالياً، تدارك مشاكل سابقة فيه.)
(في هذه الحالة المتطلبات واضحة وغير خاضعة للتغير، وهي البيئة الأمثل لاستخدام منهجية الشلال)
2. (منهجية دورة حياة المنتج التراكمية)
(راجع الفقرة a-6)
3. (خطأ)
(المنهجية الأنسب هي منهجية دورة حياة المنتج التراكمية أو آجايل)
4. (خطأ)
(منهجية RAD تناسب المشاريع التي لا تتطلب أداءً عالٍ - راجع الفقرة c-5)
5. (صح)
(تساعد النمذجة المتعاقبة في توضيح متطلبات الزبون من خلال تعليقاته على كل نموذج، زهي خيار صحيح لإدارة المشروع عندما يكون توصيف المتطلبات غير واضح - آجايل هي خيار آخر)
6. (منهجية آجايل - منهجية دورة حياة المنتج التراكمية)

10. المراجع:

1. Scrum reference card by Michael James
2. http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
3. http://www.tutorialspoint.com/sdlc/sdlc_v_model.htm
4. http://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm
5. http://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm
6. http://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm
7. https://en.wikipedia.org/wiki/Spiral_model



الفصل الثالث: منهجية التطوير سكرم (1)

كلمات المفتاحية:

سكرم، آجايل، مرحلة، تغذية راجعة، مالك المنتج، فريق التطوير، منسق سكرم، منهجية إدارة، اطار عمل، دور وظيفي، إصدار

Scrum, Agile, Sprint, Scrum master, Product owner, Development team, Product backlog

ملخص:

في هذا الفصل ملخص سريع لمنهجية التطوير سكرم، مع شرح العلاقة بين سكرم وفلسفة آجايل، ومقارنة مع الطرق التقليدية في إدارة مشاريع تطوير البرمجيات، وعرض مختصر لعناصر منهجية سكرم.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- تعريف منهجية سكرم
- شرح العلاقة بين سكرم وفلسفة آجايل
- مقارنة مع الطرق التقليدية في إدارة مشاريع تطوير البرمجيات
- عناصر سكرم
 - الأدوار الوظيفية في سكرم
 - اجتماعات سكرم الدورية
 - الورقيات المستخدمة في سكرم

1. مقدمة:

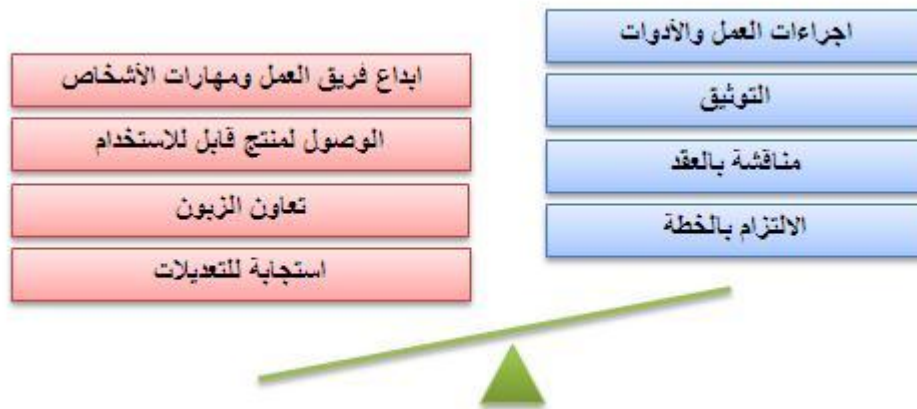
سكرم Scrum هي منهجية عمل معدة لتطوير المنتجات المعقدة (مشاريع تطوير البرمجيات هي مثال نموذجي عن مشاريع سكرم).

فرق العمل في سكرم متعددة المهام، منظمة ذاتياً يتألف كل منها من سبعة أشخاص وسطياً، وهي مبنية على مجموعة من الأدوار، الاجتماعات، القواعد والورقيات و سنأتي على تفصيل كل منها لاحقاً. فريق العمل في سكرم منظم ذاتياً وهو المسؤول عن وضع مهامه وتنسيقها ضمن مجموعة مراحل متعاقبة يطلق عليها تسمية سبرنت (Sprints)، ويكون مجمل المشروع مؤلفاً من سلسلة من هذه المراحل، وكل مرحلة مضبوطة زمنياً، تبدأ بانتهاء المرحلة السابقة وتنتهي بانتهاء مدتها، وعادة ما تكون مدة المرحلة من اسبوعين إلى ثلاثين يوم عمل.

نتيجة كل مرحلة في حال نجاحها هي منتج مختبر وقابل للاستثمار، وهو شرط الانتقال إلى المرحلة التالية، وإلا فإن نتيجة المرحلة تكون الفشل وتعاد من بدايتها.

تعتمد سكرم على مبادئ فلسفة أجايل Agile، فهي تنم:

- ابداع فريق العمل والمهارات الشخصية لأفراد الفريق أكثر من أدوات العمل والإجراءات الموصفة بدقة
- منتج قابل للاستخدام ومختبر أكثر من التوثيق المفصل، وبما أن فريق التطوير في سكرم يعمل في مكان واحد ويتعاون أفرادهم على إنجاز مهام كل مرحلة، فالتنسيق بينهم يبني على النقاشات اليومية بدلاً من الورقيات المتبادلة
- تعتمد سكرم على تعاون الزبون أثناء تطوير المنتج من خلال ممثله (مالك المنتج)، بدلاً من الالتزام ببنود العقد، والتعديل عليها في حال ظهور متطلبات جديدة أو سوء فهم للمتطلبات.
- أهم ما يميز سكرم هو الاستجابة السريعة للتعديلات بدلاً من الالتزام الأعمى بالخطة المعدة في بداية المشروع



(الشكل 1)

منهجية سكرم مناسبة جداً لتطوير المشاريع الداخلية، ولها بعض المساوئ في حالة المشاريع المحكومة بتاريخ تسليم، وسنأتي على تفصيل مساوئ وحسنات هذه المنهجية في الفصول اللاحقة.

2. سكرم بديل عن المنهجيات التقليدية في إدارة مشاريع تطوير البرمجيات:

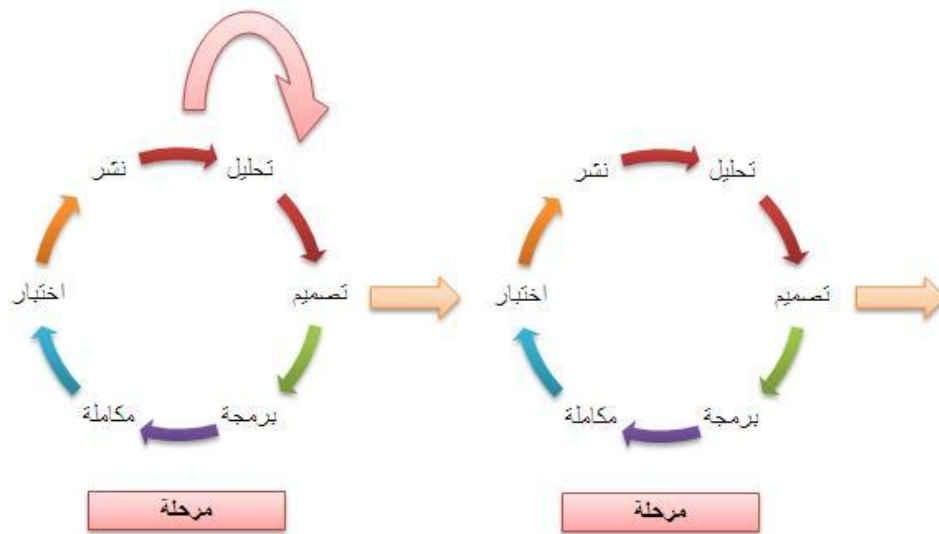
تعد المنهجية التقليدية في تطوير المنتج المعلوماتي (Waterfall)، على مجموعة مراحل محدودة (تحليل المتطلبات، التصميم، البرمجة، المكاملة، الاختبار، النشر) والانتقال من مرحلة إلى أخرى يتم دون إمكانية الرجوع إلى الوراء:



(الشكل 2)

ولنجاح المشروع، يجب انجاز كل مرحلة دون أخطاء (هامش الخطأ صفر)، وذلك يتطلب فهم كامل للمتطلبات من لحظة البدء بالمشروع، وهو أمر مخالف للطبيعة، إذ نادراً ما نكون على علم بكافة التفاصيل ونادراً ما يكون فهمنا عميق للمتطلبات في مراحل المشروع المبكرة (مرحلة التحليل)، وهذا يقود إلى أخطاء في مرحلة التحليل يبني عليها التصميم وتكتشف الأخطاء في المراحل المتأخرة، وهذا يستدعي تعديلات متكررة و إعادة الاختبار مراراً وتكراراً.

سكرم تتجاوز هذه المعضلة بخلط جميع المراحل في سلسلة متعاقبة من مراحل يطلق عليها تسمية سبرنت (Sprints)، وكل مرحلة في سكرم محدودة زمنياً وتتضمن تحليلاً لجزء من متطلبات المشروع وتصميم وتنفيذ واختبار هذا الجزء وصولاً إلى منتج جزئي مختبر وقابل للاستخدام.



(الشكل 3)

3. عناصر سكرم:

الأدوار في سكرم Roles:

الأدوار المتاحة في سكرم هي مالك المنتج، المنسق وفريق التطوير، وفيما يلي تفصيل لمهام ومسؤوليات كل منها:

• مالك المنتج

مالك المنتج Product owner هو شخص واحد، أهم مسؤولياته هي زيادة العائد على الأرباح، ويمكن تلخيص باقي مهامه في:

- وضع التصور النهائي للمنتج
- وضع الأولويات للميزات (Features) المطلوب تنفيذها، ووضع خطة نشر إصدارات المنتج (Release plan)، حيث يقوم عادة مالك المنتج بوضع تاريخ نشر الإصدار الأول، ويقوم لاحقاً (خلال الاجتماعات الدورية في سكرم)، بإعادة ترتيب الأولويات، ويقرر ما هي الميزات التي سيتم تنفيذها في الإصدار الأول للمنتج، ومن ثم ما سيتم تنفيذه في الإصدارات اللاحقة
- هو الحكم في أي جدال يخص ميزات المنتج ضمن فريق العمل
- هو المسؤول عن إعلان فشل أو نجاح مرحلة، حيث يقوم فريق العمل بنهاية كل مرحلة وضمن اجتماع التقييم، بعرض ما تم تنفيذه واختباره من مهام المرحلة أمام مالك المنتج وأصحاب المصلحة (Stakeholders)، ومالك المنتج هو من يقرر أن الميزات المقررة للمرحلة تمت وبالتالي نجحت المرحلة، أو أن الميزات المقررة لم تتم وبالتالي يعلن فشل المرحلة وإعادتها
- هو من يقرر الاستمرار في عملية التطوير أو إيقافها
- هو من يأخذ رغبات أصحاب المصلحة بعين الاعتبار، ويعكسها كميزات للمنتج أثناء تطويره
- يمكن أن يشارك كعضو في فريق التطوير

• فريق التطوير

فريق التطوير Development Team هو فريق متعدد المهام، يتألف من خمس إلى تسع أعضاء من ذوي الخبرة في تحليل النظم، البرمجة، الاختبار، التصميم وغيرها، مهمته تطوير المنتج وهو:

- منظم ذاتياً، ولا تسند الأدوار لأعضائه من خارج الفريق
- يناقش التزامه مع مالك المنتج كل مرحلة على حدى
- يملك أعضاؤه خبرة العمل كفريق
- تزداد فرص نجاح الفريق في تأدية مهامه، عند وضع جميع الأعضاء في حيز مكاني واحد (مكتب أو مخبر)، خاصة في المراحل الأولى من تطوير المنتج
- تزداد فرص نجاح الفريق في تأدية مهامه، إذا كان جميع أعضائه متفرغين تماماً للعمل في المنتج

• منسق سكرم

يمكن تلخيص دور منسق سكرم Scrum Master في الحرص على الالتزام بقواعد منهجية العمل سكرم، ومن مهامه:

- يساعد في خلق بيئة العمل المناسبة لفريق العمل، وتذليل العقبات في حال وجودها
- يقوم بمتابعة تطور العمل واستخلاص مؤشرات الإنجاز وتعديل التوقعات أولاً بأول
- يحمي فريق العمل من أي تدخل خارجي
- يفرض المواعيد النهائية للاجتماعات ويعلن نهاية كل مرحلة (بصرف النظر عن نتائجها)
- يضمن وجود ورقيات سكرم في متناول يد الجميع
- لا يملك أي سلطة إدارية على فريق العمل (أي شخص له سلطة إدارية على أعضاء فريق العمل، لا يصلح أن يكون منسق سكرم)

اجتماعات سكرم :Scrum meetings

تعتمد منهجية سكرم على الكثير من الاجتماعات، التي تضمن مستوى أعلى من التعاون بين أعضاء فريق التطوير، وانسجاماً بين متطلبات أصحاب المصلحة وما يتم تنفيذه تبعاً من قبل الفريق. وتتيح الاجتماعات المتوالية امكانية التعديل على المتطلبات واستجابة عالية للتعديلات. اجتماعات سكرم هي خمسة أنواع، كما هو موضح في الشكل رقم 4.



(الشكل 4)

• اجتماع تخطيط المرحلة

يقوم كل من منسق سكرم وفريق التطوير ومالك المنتج في بداية كل مرحلة باجتماع مهمته الأساسية هي انتقاء مجموعة من ميزات المنتج، لتكوّن قائمة مهام المرحلة (Sprint backlog items)، هذه القائمة هي ما يلتزم فريق التطوير بإنجازه ضمن الإطار الزمني للمرحلة، وما يعنيه الإنجاز هنا، هو تحويل مجموعة الميزات التي تم انتقاؤها إلى منتج مختبر وقابل للاستثمار.

عادة يكون احتمال الفشل في تنفيذ مهام المرحلة أكبر في المراحل الأولى من العمل، وفي المراحل اللاحقة يزداد التعاون بين أعضاء فريق التطوير، وتزداد قدرتهم على تقدير زمن الإنجاز لمهام المرحلة، ولذلك ينصح بأن يلتزم فريق التطوير بمهام أقل في المراحل الأولى.

• اجتماع سكرم اليومي

هو اجتماع يومي مدته لا تتجاوز خمسة عشر دقيقة، ليست الغاية منه حل المشاكل، وإنما هي الإجابة على ثلاثة أسئلة لكل واحد من أعضاء فريق التطوير:

- ماذا انجزت خلال اليوم السابق؟
- ماذا سأنجز اليوم؟
- ما هي العقبات التي تعيق عملي؟

وبعد عرض كل من أعضاء فريق التطوير اجابته على الاسئلة السابقة، يتم تعديل الإنجاز اليومي في قائمة مهام المرحلة، والتي تتضمن ستة قوائم فرعية هي: قائمة ميزات المنتج، قائمة ميزات المرحلة، مهام المرحلة، المهام التي لم يتم البدء بها، المهام قيد التنفيذ والمهام المنتهية. يقوم كل عضو في فريق التطوير بنقل المهام حسب ما أنجز بين القوائم الثلاث الأخيرة، ولنجاح المرحلة يجب أن تكون جميع المهام قد نقلت من قائمة مهام المرحلة إلى قائمة المهام المنتهية، وذلك في الاجتماع الأخير من الاجتماعات اليومية للمرحلة.

مميزات المنتج	مميزات المرحلة	مهام المرحلة	لم تبدأ	قيد التنفيذ	منتهية
<div style="background-color: #f4a460; padding: 5px; margin-bottom: 5px;">الولوج للنظام</div> <div style="background-color: #d9d9d9; padding: 5px; margin-bottom: 5px;">عرض قائمة بأسماء الطلاب</div> <div style="background-color: #f4c400; padding: 5px; margin-bottom: 5px;">.</div> <div style="background-color: #6699cc; padding: 5px; margin-bottom: 5px;">.</div> <div style="background-color: #66cc66; padding: 5px;">.</div>	<div style="background-color: #f4a460; padding: 5px; margin-bottom: 5px;">الولوج للنظام</div> <div style="background-color: #d9d9d9; padding: 5px; margin-bottom: 5px;">عرض قائمة بأسماء الطلاب</div> <div style="background-color: #f4c400; padding: 5px;">.</div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">التحقق من أن المستخدم مسجل في قاعدة البيانات</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">التحقق من كلمة المرور</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">عرض الاسم الكامل للمستخدم في الصفحة الرئيسية</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">.</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">.</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">.</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">.</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">.</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">.</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">.</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">.</div>			

(الشكل 5)

يوضح الشكل رقم 5 مثلاً عن قائمة ميزات المرحلة، وفيه نرى أن ثلاثة ميزات من قائمة ميزات النظام تم انقائها لتكوّن قائمة ميزات المرحلة (تم ذلك في اجتماع تخطيط المرحلة)، ومن ثم تمت ترجمة هذه الوظائف إلى مهام نقلت تدريجياً من قائمة المهام إلى قائمة المهام المنتهية أثناء تطور العمل. يجب الانتباه هنا إلى أن قائمة ميزات النظام يمكن أن تتغير ويمكن أن تتغير الأولويات بينها حسب قرار مالك المنتج (يتم التعديل على قائمة ميزات المنتج خلال اجتماعات تنقيح ميزات المنتج التي تلي كل مرحلة)، ويمكن أن تتغير قائمة المهام خلال الاجتماعات اليومية إلا أن وظائف المرحلة تبقى ثابتة طيلة فترتها.

• اجتماع تقييم المرحلة

يعقد هذا الاجتماع عند انتهاء زمن المرحلة، وفيه يقدم فريق التطوير عرضاً لما قام بتنفيذه، أمام مالك المنتج وأصحاب المصلحة، ويفضل أن يكون ذلك على شكل عرض حي (Demo)، وليس على شكل تقرير، وفي حال كانت جميع العناصر المذكورة في قائمة ميزات المرحلة قد انتهت، يقوم مالك المنتج بإعلان نجاح المرحلة وإمكانية الانتقال إلى مرحلة تالية، أما في حال كانت بعض الميزات غير منتهية فيقوم مالك المنتج بإعلان فشل المرحلة والعودة بها إلى بدايتها (أي الانطلاق بها من اجتماع تخطيط مرحلة جديد).

يمكن أن يحضر هذا الاجتماع كل من هو مهتم بالمنتج، بما في ذلك المستثمرين النهائيين، وبنتيجة المناقشات في هذا الاجتماع يمكن أن تظهر متطلبات جديدة، أو تصحيح لفهم سابق لبعض المتطلبات، ويمكن لمالك المنتج في هذه الحالة تعديل قائمة ميزات المنتج وترتيب الأولويات فيها.

• اجتماع التغذية الراجعة للمرحلة

هذا الاجتماع مخصص لأعضاء فريق التطوير فقط، وفيه يعرض أعضاء الفريق ما تعلموه خلال المراحل المنجزة لتاريخه، ويتداولون في ما يعيق عملهم وما يمكن تحسينه في شروط وطرق العمل لتحسين الإنتاجية في المراحل التالية من العمل. يمكن أن يحضر الاجتماع منسق سكرم، في حال أراد فريق التطوير مناقشة العوائق ومقترحات لتحسين بيئة العمل معه.

• اجتماع تنقيح ميزات المنتج

قائمة ميزات المنتج هي قائمة بالوظائف المطلوب من المنتج أن يحققها عند إنجازه، يتم وضعها من قبل مالك المنتج لحظة الانطلاق بالمشروع، تكون صغيرة في البداية ويزداد حجمها بعد كل مرحلة، بنتيجة تغير وتطور المتطلبات وتطور فهم الجميع (فريق التطوير، مالك المنتج ومنسق سكرم) لما هو مطلوب من النظام (المنتج).

يقوم فريق التطوير والمنسق ومالك المنتج، بمناقشة التعديلات والإضافات على هذه القائمة في اجتماع دوري، يفضل أن يلي كل مرحلة في بداية العمل، ومن ثم يمكن أن يلي عدة مراحل مجتمعة، والقرار النهائي لتعديل قائمة ميزات المنتج هو لمالك المنتج حصراً.

الورقيات Scrum artifacts:

• قائمة ميزات المنتج

هي قائمة مرتبة حسب الأهمية للوظائف المطلوب من النظام تحقيقها، ويجب أن تكون مرئية لجميع أصحاب المصلحة، ويمكن لأي من أصحاب المصلحة - بمن فيهم أعضاء فرق التطوير - إضافة إلى هذه القائمة بعد موافقة مالك المنتج، ويمكن إعادة ترتيب عناصر القائمة حيث تمثل العناصر في قمة القائمة العناصر الأكثر أهمية، ويتم تصحيح هذه القائمة دورياً خلال اجتماعات تعديل قائمة ميزات المنتج التي تلي كل مرحلة تطوير.

يمثل كل عنصر من القائمة إحدى الوظائف المطلوبة من النظام، وتوصف بكلمات تعبر عما هو مطلوب وليس عن كيفية تحقيقه، وأحياناً يطلق على هذه العناصر تسمية "سيناريو عمل"، ويمكن التعبير عنها كشرط قبول للنظام.

حجم العمل المطلوب لتنفيذ هذه العناصر يتم تقديره من قبل فريق التطوير حصراً.

• قائمة وظائف المرحلة

هي مجموعة من عناصر قائمة ميزات النظام، ينتقيها فريق التطوير ويلتزم بتنفيذها خلال المرحلة (تتم عملية الانتقاء في اجتماع تخطيط المرحلة)، ثم يقوم فريق التطوير بترجمة عناصر هذه القائمة إلى مهام Tasks، توضع في عمود المهام التي لم يتم البدء بها (كما هو موضح في الشكل رقم 5)، وتنتقل تباعاً، في الاجتماعات اليومية، إلى عمود المهام قيد التنفيذ ومن ثم إلى عمود المهام المنتهية.

قد يجد فريق التطوير وخلال تنفيذه لمهام المرحلة، مهام جديدة ضرورية لتحقيق هدف المرحلة، عندها يمكن لفريق التطوير إضافة هذه المهام إلى قائمة ميزات المرحلة خلال الاجتماع اليومي لفريق التطوير.

عادة تجزء كل وظيفة من قائمة ميزات المرحلة، إلى مهام صغيرة، لا تتعدى مدة تنفيذ الواحدة منها، يوم عمل (ثمان ساعات).

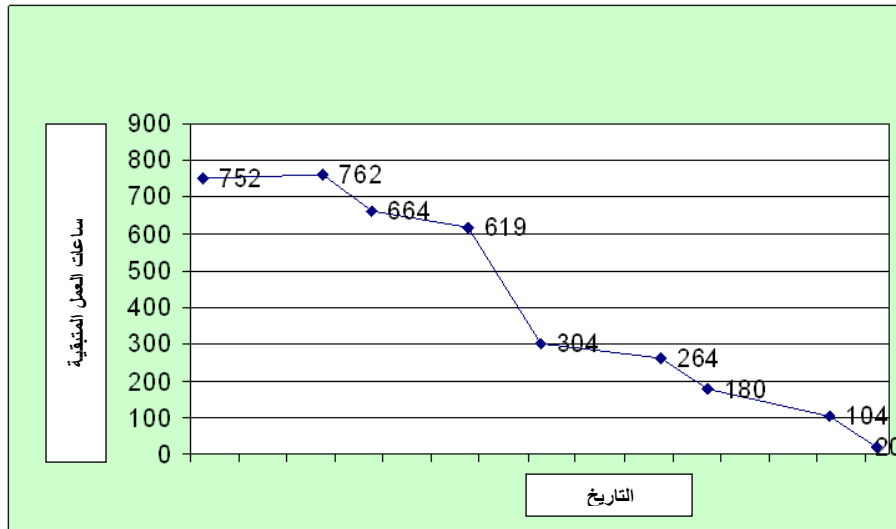
اسناد المهام اليومية لأعضاء فريق التطوير، يتم بتطوع أعضاء الفريق لانتقاء مهمة أو أكثر في الاجتماع اليومي، كل حسب خبرته، وحسب ارتباط المهمة بمهام منجزة سابقاً من قبل نفس الشخص.

● **مخطط تعقب المرحلة**

مخطط تعقب المرحلة Sprint Burn down Chart هو مخطط بياني، يعرض يومياً ما تبقى من الجهد، لإنجاز المرحلة، مقدراً بساعات عمل ويعاد رسمه يومياً حسب المتغيرات اليومية في الإنجاز.

عند ترجمة ميزات المرحلة إلى مهام (في اجتماع تخطيط المرحلة)، وعند إضافة مهام جديدة للقائمة، يقدر حجم المهمة (صغيرة، متوسطة أو كبيرة)، ويترجم هذا الحجم لساعات عمل في مخطط تعقب المرحلة.

أحياناً يزداد تقدير المتبقي من ساعات العمل بدلاً من أن يتناقص في مخطط التعقب (أي أن المنحني البياني يتصاعد بدلاً من أن يهبط)، وذلك بسبب سوء تقدير حجم المهام وسوء تقدير أعضاء فريق التطوير لإنتاجيتهم، إلا أنه لا يلبث أن يبدأ بالتناقص لينتهي إل الصفر مع نهاية زمن المرحلة في حال نجاحها.



(شكل 6)

● **مخطط تعقب الإصدار/النظام**

في مخطط تعقب المرحلة يتم تتبع الإنجاز يوماً بيوم، أما في مخطط تعقب الإصدار/النظام في مخطط Release/Product Burn down Chart فيتم تتبع الإنجاز مرحلة تلو الأخرى، ويقدر المتبقي من الجهد فيه بالأيام بدلاً من ساعات العمل. ومنه يمكن توقع تاريخ إنهاء النظام أو الإصدار، علماً أن هذا التاريخ ثابت في مخطط الإصدارات، إلا أن التأخير أو الإنجاز المبكر وارد الحدوث، ومن المفيد التنبؤ بحدوثه مسبقاً.

4. مقارنة بين سكرم والمنهجيات التقليدية لتطوير البرمجيات:

منهجية الشلال	
السيئات	الحسنات
<ul style="list-style-type: none"> • منهجية الشلال خطية، وخرج كل مرحلة هو دخل للمرحلة التالية، وفي حال الأخطاء في إحدى المراحل، سيزداد الوضع سوءاً كلما تقدمنا في المراحل، والعودة لمرحلة سابقة مكلف جداً. • قد تظهر مشاكل تقنية في مراحل متقدمة (خاصة في مرحلة البرمجة)، لا يمكن التنبؤ بها في المراحل الأولى، وهذا يفرض الانتقال إلى بدائل تتطلب إعادة مراحل سابقة. • الاستجابة للتعديلات أثناء المراحل المتقدمة، صعبة جداً، وهي تتطلب اختبارات متكررة ومضجرة لفريق العمل. • قياس ما تم انجازه في لحظة زمنية معينة، و ما نسبة المتبقي لإنجازه، صعب وغير دقيق. 	<ul style="list-style-type: none"> • الاشكالات التي قد تظهر أثناء مرحلة التطوير، يمكن اكتشافها في مرحلة تصميم النظام، أي قبل البدء بالبرمجة، وعندها يمكن اختيار بدائل توفر على فريق العمل جهداً كبيراً. • تعطي منهجية الشلال أهمية كبير للتوثيق، خاصة في مرحلة جمع المتطلبات ومرحلة التصميم، وهذا يعطي ثقة للشركات، كون احتمال تغير فريق العمل واردة، وهنا يلعب التوثيق دوراً مهماً في إتاحة امكانية متابعة العمل من نقطة توقفه، ويسهل عمليات الصيانة لاحقاً. • منهجية الشلال خطية، بمعنى أن المرحلة تبدأ بانتهاء المرحلة السابقة، ودخل المرحلة هو خرج سابقتها، والنتيجة أن منهجية الشلال سهلة الفهم والتطبيق. • تتيح منهجية الشلال امكانية تخصص فريق العمل، فمن يعمل في مرحلة التصميم ليس بالضرورة أن يكون في مرحلة التطوير، وليس بالضرورة أن يكون لديه مهارات في لغات البرمجة.

منهجية سكرم	
السيئات	الحسنات
<ul style="list-style-type: none"> • أصعب للفهم والتطبيق من المنهجيات التقليدية • بسبب التركيز على مفهوم المنتج القابل للاستخدام كنتيجة لكل مرحلة، قد يهمل توثيق النظام، وهذا يقود إلى صعوبات في التعديل لاحقاً • فريق التطوير في سكرم، مسؤول عن كثير من القرارات بشكل جماعي، هذا قد يسبب ضياعاً في تحديد المسؤوليات 	<ul style="list-style-type: none"> • تسليم منتج قابل للاستخدام، أسرع بكثير من المنهجيات التقليدية، ويمكن وضع المنتج أو الاصدار الأول منه قيد الاستخدام، ومن ثم تطوير ميزات أخرى في اصدارات لاحقة • تعاون كبير بين فريق التطوير وأصحاب المصلحة من النظام، ذلك يضمن فهماً أعمق لمتطلبات أصحاب المصلحة، ويقال من احتمالات سوء الفهم (في المنهجيات التقليدية، ينتظر أصحاب المصلحة إلى نهاية المرحلة الأخيرة من دورة حياة المنتج ليروا النتيجة، وقد تكون مختلفة عن تصوراتهم) • استجابة سريعة للتعديلات، فأى تعديل يدخل كميزة في قائمة ميزات المنتج في أي لحظة من دورة حياة المنتج، ويقرر أهميته مالك المنتج بالاتفاق مع أصحاب المصلحة • تتيح سكرم امكانية التحسين المتواصلة للمنتج، من خلال الاصدارات المتتالية • شفافية عالية بين فريق التطوير ومالك المنتج ومنسق سكرم وأصحاب المصلحة من المنتج

5. تدريبات:

1. الأدوار الوظيفية في سكرم هي ؟ (احذف 1 من الإجابات)

- مالك المنتج
- مبرمج
- منسق سكرم
- فريق تطوير

2. منهجية سكرم تقلص كثير من الأعباء الإدارية، كون فريق التطوير منظم ذاتياً ؟ (اختر 1 من

الإجابات)

- صح
- خطأ

3. هو المسؤول عن زيادة العائد على الأرباح (ROI) ؟ (اختر 1 من الإجابات)

- مالك المنتج
- مدير المشروع
- منسق سكرم
- فريق تطوير

4. أولى مهام مالك المنتج هي وضع قائمة مرتبة حسب الأولوية للوظائف المطلوبة من النظام، هذه

القائمة تدعى ؟ (اختر 1 من الإجابات)

- قائمة مهام المرحلة
- الدراسة التحليلية للنظام
- قائمة ميزات المنتج
- وثيقة جمع المتطلبات

5. يمكن أن يكون منسق سكرم ومالك المنتج هما الشخص نفسه ؟ (اختر 1 من الإجابات)

- صح
- خطأ

الإجابة الصحيحة:

1. (مبرمج)

(أعضاء فريق التطوير في سكرم متعددو المهام، يمكن أن يكون العضو مبرمج، محلل نظم أو مصمم واجهات تخاطبية، وغير ذلك)

2. (صح)

(فريق التطوير في سكرم منظم ذاتياً بمعنى أنه لا سلطة إدارية لأحد الأعضاء على الآخرين، والقرارات تؤخذ بالتشارك فيما بينهم، والمسؤولية تقع على كامل فريق التطوير، هذه النقطة تعتبر أحد المآخذ على منهجية سكرم لأن المسؤولية فيها غير محددة بدقة على مستوى الشخص)

3. (مالك المنتج)

(راجع مهام مالك المنتج الفقرة 3-a)

4. (قائمة ميزات المنتج)

(راجع الورقيات في سكرم الفقرة 3-c)

5. (خطأ)

(مهام المنسق ومالك المنتج مختلفة عن بعضها، وقد تكون متناقضة، فمثلاً المنسق هو من يحل الخلاف بين فريق التطوير ومالك المنتج في حال حدوثه)

6. المراجع:

1. Scrum reference card by Michael James
2. An Introduction to Agile scrum methodology by Bhaskar V. Shankar
3. The scrum guide by Ken Schwaber and Jeff Sutherland



الفصل الرابع: منهجية التطوير سكرم (2)

كلمات المفتاحية:

سكرم، آجايل، مرحلة، تغذية راجعة، مالك المنتج، فريق التطوير، منسق سكرم، منهجية إدارة، اطار عمل، دور وظيفي، إصدار، اجتماع سكرم اليومي، اجتماع مراجعة المرحلة، اجتماع تنقيح ميزات المنتج، اجتماع التخطيط للمرحلة، اجتماع التغذية الراجعة للمرحلة.

Scrum, Agile, Sprint, Scrum master, Product owner, Development team, Product backlog, sprint backlog, scrum daily meeting, sprint planning meeting, backlog refinement meeting, sprint retrospective meeting, sprint review meeting

ملخص:

في هذا الفصل نشرح بالتفصيل مع الأمثلة عناصر منهجية سكرم، مع مجموعة تدريبات وتعليقات عليها تعمق فهمنا للمنهجية وكيفية تطبيقها.

أهداف تعليمية:

يهدف هذا الفصل إلى شرح:

- عناصر منهجية سكرم
- اجتماعات سكرم
- الورقيات في سكرم
- تطبيق منهجية سكرم مع عدة فرق تطوير
- منهجيات أخرى معتمدة في آجايل

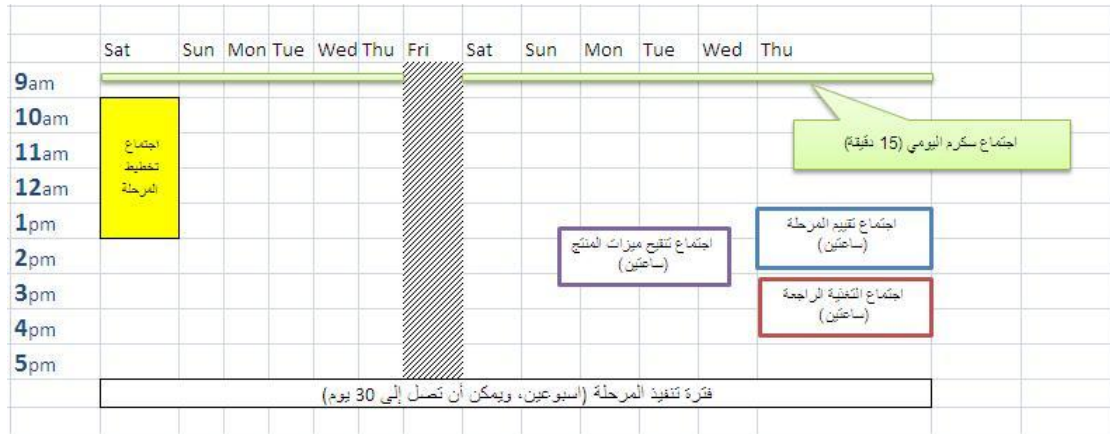
1. مقدمة:

رأينا في الفصل السابق أن سكرم Scrum هي منهجية معتمدة على أجائل Agile لتطوير المنتجات تراكمياً و أكثر مجالات استخدامها هي تطوير المنتجات البرمجية. و رأينا أن سكرم تعتمد على فرق عمل متعددة المهام، منظمة ذاتياً، وهي مبنية على مجموعة من الأدوار، الاجتماعات، القواعد والورقيات، وهذه العناصر هي ما سنشرحه في هذا الفصل مع أمثلة وتدريبات تعمق فهمنا للمنهجية و كيفية تطبيقها.

2. اجتماعات سكرم:

اجتماعات سكرم هي خمسة أنواع كما نرى في الشكل 1، لكل منها غاية ومجموعة من الحاضرين، سنعرضها فيما يلي:

أولاً: اجتماع تنقيح ميزات المنتج:



(الشكل 1)

قائمة ميزات المنتج، هي قائمة مرتبة من الميزات التي يفترض أن يحققها النظام عند اكتماله، وهي نقطة الانطلاق في بناء المنتج، ويعمل على وضعها مالك المنتج بالتعاون مع أصحاب المصلحة من المنتج. كل عنصر من قائمة ميزات المنتج له مجموعة خصائص، تدعى اختصاراً بـ INVEST وهي تشير إلى:

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

Independent	كتلة مترابطة منطقياً، ويمكن عزلها عن باقي متطلبات النظام.
Negotiable	المطورون وأصحاب المصلحة يتناقشون حول حدود وتفاصيل هذه الكتلة، بمعزل عن باقي أجزاء النظام.
Valuable	ذات أهمية بالنسبة لأصحاب المنفعة من النظام.
Estimable	يمكن قياس المجهود اللازم لإنجازها.
Small	يجب ألا تكون أكبر حجماً من ربع المرحلة، والمثالي أن تكون بحجم عمل يوم أو يومين.
Testable	لها مخرجات محددة، ويمكن التحقق من إنجازها بالكامل.

أحد الأعضاء الأساسيين في اجتماع تنقيح ميزات المنتج، هو مالك النظام الذي يمثل وجهة نظر أصحاب المصلحة، ومن المفيد أن يكون أصحاب المصلحة حاضرين وحتى المستخدمين النهائيين، ومن الجدير ذكره هنا أن أحد أهم مبادئ فلسفة آجايل التي تعتمد عليها سكرم، هي أن يعمل أصحاب المصلحة مع المطورين بشكل مباشر ويومي.

فريق التطوير منظم ذاتياً: يعني ذلك أن كل فرد من الفريق يعمل لدى الآخرين ومسؤول أمامهم، وليس لأحدهم سلطة على الآخرين دون غيره.

لتقدير حجم العمل اللازم لإنجاز أحد عناصر قائمة الميزات، يقوم كل واحد من أعضاء فريق التطوير بوضع تقديره الخاص (صغير، وسط، كبير، كبير جداً)، وبعد عرض كل عضو في الفريق لتصوره عن هذا العنصر والنقاش مع الآخرين، قد تظهر متطلبات جديدة.

مثال:

أحد أفراد الفريق قام بتقدير حجم العمل اللازم للعنصر الأول "عرض علامات الطلاب" بـ "كبير جداً"، وعند نقاشه تساءل عن امكانية عرض علامة امتحان سابق (من فصل سابق)، وهنا وبالمناقشة مع مالك المنتج أوضح مالك المنتج أن عرض العلامات السابقة هو متطلب رئيسي ولا يمكن الاكتفاء بالفصل الحالي، ويمكن وضعه في الأولوية بعد "تعديل العلامات من قبل المدرس"، وترجمة ذلك هي اضافة عنصر جديد إلى القائمة "عرض علامات فصول سابقة".



يعيد الفريق تقديره لحجم العمل بعد النقاش، ويتم اعتماد وسطي التقديرات، أو أكبرها باتفاق جميع الأعضاء.

يمكن تكرار اجتماع تنقيح ميزات المنتج كلما دعت الحاجة، وعادة يمتد لساعتين وسطياً.

تمارين:

1. أحد أهم أهداف اجتماع تنقيح ميزات المنتج ؟ (اختر 1 من الإجابات)

- تقدير دقيق لحجم العمل
- فهم المتطلبات بشكل عميق، وجمعها في عناصر أكبر حجماً
- فهم المتطلبات بشكل عميق، وجمعها في عناصر أصغر حجماً
- توثيق المتطلبات بشكل دقيق وقابل للفهم من قبل أعضاء فريق التطوير

(فهم المتطلبات بشكل عميق، وجمعها في عناصر أصغر حجماً)

(عادة يزداد حجم قائمة ميزات المنتج مع تقدم العمل، وذلك نتيجة تجزئة عناصره وظهور متطلبات

جديدة تترجم لعناصر جديدة في القائمة)

2. عناصر قائمة ميزات المنتج التي لها نفس الأهمية، تجمع في فئات ؟ (اختر 1 من الإجابات)

- صح
- خطأ

(خطأ)

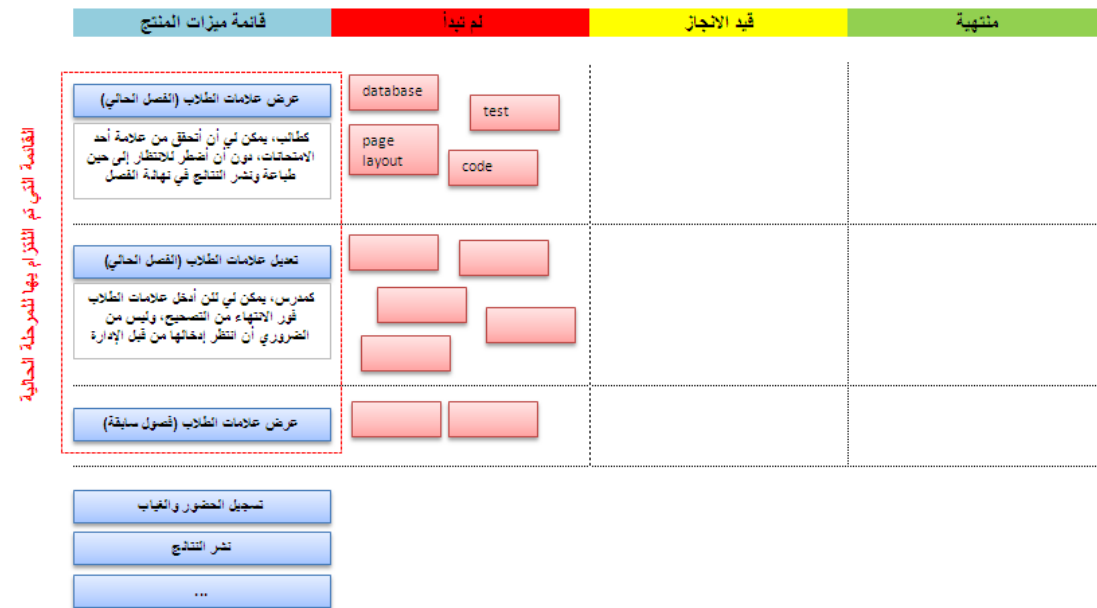
(عناصر قائمة ميزات المنتج، هي حتماً مرتبة حسب أولوية تنازلية، العناصر الأهم في أعلى القائمة والعناصر الأقل أهمية في أسفلها)

ثانياً: اجتماع التخطيط للمرحلة:

يقوم كل من منسق سكرم وفريق التطوير ومالك المنتج في بداية كل مرحلة باجتماع مهمته الأساسية هي انتقاء مجموعة من ميزات المنتج، لتكوّن قائمة ميزات المرحلة (Sprint backlog items)، هذه القائمة هي ما يلتزم فريق التطوير بإنجازه ضمن الإطار الزمني للمرحلة، وما يعنيه الإنجاز هنا، هو تحويل مجموعة الميزات التي تم انتقاؤها إلى منتج مختبر وقابل للاستثمار.

مهمة المنسق في هذا الاجتماع، هي الاشراف على اتباع قواعد سكرم، ومهمة الفريق مع مالك المنتج هي انتقاء الميزات من قائمة ميزات المنتج، وتقدير الجهد اللازم لإنجازها، وترجمتها لمهام (حجم المهمة يقابل عمل شخص لمدة يوم أو يومين على الأكثر)، والاتفاق على القائمة النهائية للمرحلة والتي يلتزم الفريق بإتمامها خلال مدة المرحلة.

عادة يمتد الاجتماع لأربع ساعات لمرحلة مدتها اسبوعين، أي بمعدل ساعتين لكل اسبوع من مدة المرحلة.



يكون احتمال الفشل في تنفيذ مهام المرحلة أكبر في المراحل الأولى من العمل، وفي المراحل اللاحقة يزداد التعاون بين أعضاء فريق التطوير، وتزداد قدرتهم على تقدير زمن الإنجاز لميزات المرحلة، ولذلك ينصح بأن يلتزم فريق التطوير بميزات أقل في المراحل الأولى. اجتماع تخطيط المرحلة يتألف من جزئيين، في الجزء الأول يقوم الفريق بانتقاء الميزات من قائمة ميزات المنتج والنقاش لتقدير حجمها، وفي الجزء الثاني تتم ترجمة هذه الميزات إلى مهام وإعادة تقدير حجمها.

تمارين:

1. ما الفرق بين قائمة ميزات المنتج وقائمة ميزات المرحلة ؟ (اختر 1 من الإجابات)

- لا يوجد فرق
 - قائمة ميزات المنتج تتضمن ميزات، قائمة ميزات المرحلة تتضمن أخطاء لتصحيحها
 - قائمة ميزات المنتج تتضمن كل ما سيتم انجازه، بينما تتضمن قائمة ميزات المرحلة ما يلتزم فريق التطوير بإنجازه خلال المرحلة الحالية
- (قائمة ميزات المنتج تتضمن كل ما سيتم انجازه، بينما تتضمن قائمة ميزات المرحلة ما يلتزم فريق التطوير بإنجازه خلال المرحلة الحالية)

2. هل يتوجب على فريق التطوير معرفة كل المهام اللازم انجازها، لتحقيق ميزات المرحلة ؟ (اختر 1 من الإجابات)

- نعم
- لا

(لا)

(كثير من المهام يتم اكتشافها أثناء تنفيذ المرحلة)

3. هل من المقبول تأجيل اختبارات مرحلة إلى مرحلة لاحقة ؟ (اختر 1 من الإجابات)

- نعم
- لا

(لا)

(نتيجة كل مرحلة هي منتج جزئي قابل للاستخدام، أي أنه مختبر حتماً)

ثالثاً: اجتماع سكرم اليومي:

اجتماع يومي، يتم كل صباح لمدة 15 دقيقة، الهدف منه وضع جدول أعمال اليوم، ويحضره فريق العمل ومنسق سكرم، وقد يحضره مالك المنتج بناءً على طلب من فريق التطوير للاستيضاح عن بعض المتطلبات أو لإضافة ميزات جديدة إلى قائمة ميزات المنتج.

في هذا الاجتماع يجب كل من أعضاء فريق التطوير على ثلاثة أسئلة، هي:

- ماذا فعلت بالأمس
- ماذا سأفعل اليوم
- ما المعوقات التي واجهتها

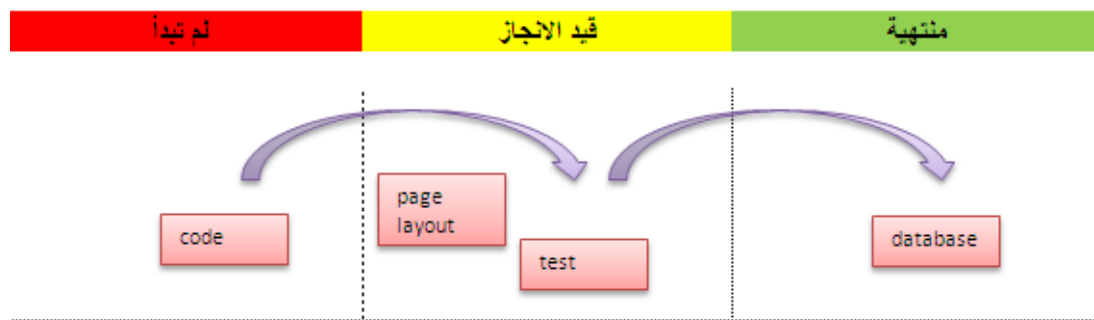
مثال:

"يوم أمس أنهيت مقارنة بين قاعدة بيانات النظام القديم، وقاعدة بيانات النظام الحالي، وقمت بوضع مخطط نقل البيانات الخاصة ببيانات الطلاب، وعلامات الامتحانات من الفصول السابقة إلى واجهات العرض في نظامنا الحالي"

"اليوم سأقوم بكتابة الاجرائيات اللازمة لاستيراد بيانات الطلاب وعلامات الامتحانات، وسأقوم باختبارها على مجموعة طلاب موجودين فعلياً في النظام القديم"

"الصعوبة التي قد تعوق اختبراتي، هي أن واجهات العرض غير منتهية لتاريخه، وليست لدي فكرة عن أسلوب العرض (علامات طالب في جميع الامتحانات - علامات مادة لجميع طلاب صف معين - ...)، ويجب أن نتفق اليوم على من سيقوم بتصميم واجهات العرض، وعلى طريقة العرض الأنسب" من المفيد أن يقوم منسق سكرم بوضع قائمة بكافة الصعوبات التي تكلم عنها أعضاء فريق التطوير، بشكل ظاهر للجميع (على لوح مثلاً)، يتيح ذلك امكانية مشاركة الأعضاء في حل مشاكل الآخرين. يشجع الاجتماع اليومي على التواصل بين أعضاء فريق التطوير، ويعزز العمل الجماعي.

يقوم كل فرد من فريق التطوير بانتقاء المهمة التي سيعمل عليها اليوم، من قائمة مهام المرحلة، ويضعها تحت عنوان "قيد الانجاز"، وعند انقائها ينقلها تحت العنوان "منتهية"، وفي نهاية المرحلة يجب أن تكون جميع المهمات قد انتقلت من "لم تبدأ" إلى "منتهية"، حتى تكون نتيجة المرحلة هي النجاح.



تمارين:

1. الحجم المثالي لمهمة في قائمة مهام المرحلة ؟ (اختر 1 من الإجابات)

- عمل يوم لشخص واحد أو أقل
 - عمل شخصين أو ثلاث لمدة يومين أو ثلاث على الأكثر
- (عمل يوم لشخص واحد أو أقل)
- (الحجم الصغير للمهام يسهل اكتشاف المشاكل باكراً، ويسهل تتبع الانجاز وقياسه)

2. من المسؤول عن انتقاء أدوات تنفيذ المشروع ؟ (اختر 1 من الإجابات)

- منسق سكرم
 - إدارة الشركة
 - فريق التطوير بالتنسيق بين أعضائه
 - مالك المنتج
- (فريق التطوير بالتنسيق بين أعضائه)

رابعاً: اجتماع تقييم المرحلة:

يتم في اجتماع تقييم المرحلة، عرض حي لما تم انجازه (يجب أن نتذكر هنا أن نتيجة كل مرحلة هي منتج مختبر وقابل للاستخدام، وبما أن اجتماع تقييم المرحلة يأتي في نهايتها، فالمفروض أن يكون بإمكان فريق التطوير عرض ما تم انجازه كمنتج وليس عرض تقديمي Power point presentation).

وفي هذا الاجتماع يقرر مالك المنتج أي من الميزات ثم انجازها (Done)، وهل تم تحقيق أهداف المرحلة أم لا.

العناصر غير المحققة، تعاد إلى قائمة ميزات المنتج لتدخل في مراحل لاحقة.

يحضر هذا الاجتماع كل من فريق التطوير، منسق سكرم، مالك المنتج وأصحاب المصلحة.

أصحاب المصلحة قد يزيدون بعض الميزات إلى قائمة ميزات المنتج، بعد نقاشهم مع فريق العمل والعرض الحي الذي تم بحضورهم.

ويمكن تلخيص جدول الأعمال النموذجي لاجتماع تقييم المرحلة، في أربع نقاط هي:

- عرض حي لما تم انجازه
- مالك المنتج يقرر العناصر المنتهية، وقرر نتيجة المرحلة
- يمكن قياس تطور العمل (نسبة ما تم انجازه، وهل نحن متأخرين)
- رد فعل أصحاب المصلحة من المنتج

تمارين:

1. ما هو أول ما تتوقع رؤيته في اجتماع تقييم المرحلة ؟ (اختر 1 من الإجابات)

- عرض Power point لم أنجز
- عرض حي لما تم انجازه
- تقارير ومخططات بيانية توضح سير العمل وتقدمه
- تقرير مختصر عما حدث في المرحلة

(عرض حي لما تم انجازه)

2. حسب فلسفة آجايل، ما هي الطريقة الأفضل لقياس تقدم العمل ؟ (اختر 1 من الإجابات)

- كم من المهام أنجز
- كم من المهام تم البدء به

(كم من المهام أنجز)

مثال:

عن النقاش الممكن أن يحدث في اجتماع تقييم المرحلة:

• "فريق التطوير: كم نرى يمكن للطلاب أن يدخل النظام، ويستعرض علاماته للمواد التي تم نشرها"

• "أصحاب المصلحة: كنا نتوقع استخدام ألوان ذات دلالة، كأن تظهر نتيجة الرسوب بلون أحمر مثلاً"

• "منسق سكرم: يمكن نقاش اضافة استخدام الألوان كميزة جديدة تضاف إلى قائمة ميزات المنتج مع مالك المنتج"

• "مالك المنتج: يمكن إضافة ميزة استخدام ألوان للدلالة على النتيجة، ولكن ليست ذات أهمية كبيرة لذلك سنضعها في أسفل قائمة ميزات المنتج، ويمكن أن تدخل في المراحل اللاحقة "

في حال كان أحد عناصر قائمة ميزات المرحلة غير مختبر، فهو غير منتهي، ولا معنى للقول أنه منتهي بنسبة 80% أو ما شابه، لأن من أهم مبادئ فلسفة آجايل هي وضوح النتائج، وأي عنصر غير مختبر لا يمكن التنبؤ بحجم التعديلات التي قد تنتج عن اختباره.

3. ماذا يتوجب على مالك المنتج فعله تجاه العناصر غير المنتهية من قائمة ميزات المرحلة ؟ (اختر 1 من الإجابات)

- العنصر غير المنتهي يتم تجزئته، ويعاد الجزء غير المنتهي كعنصر جديد إلى قائمة ميزات المنتج.
- العنصر غير المنتهي يعاد كاملاً إلى قائمة ميزات المنتج.
- العنصر غير المنتهي، يكلف بإتمامه أحد من خارج فريق التطوير.

(العنصر غير المنتهي يعاد كاملاً إلى قائمة ميزات المنتج)

أحد بنود جدول أعمال اجتماع تقييم المرحلة، هو قياس تطور العمل، وهي مهمة صعبة ولا يمكن أن تتم بدقة، ولكن يمكن تقدير حجم الانجاز بنقاط تحسب من عدد وتثقل العناصر المنتهية، فمثلاً إذا كان المنجز هو ثلاثة عناصر، منها اثنان بحجم صغير وواحد متوسط الحجم، فمجموع النقاط يمكن أن يكون $8=4+2+2$.

هذه النقاط والتي تمثل قياساً لنتيجة المرحلة، يمكن اعتمادها للمقارنة بين انجاز المرحلة وانجاز المراحل الأخرى، وبحساب نقاط عدة مراحل متتالية يمكن التنبؤ بما يمكن انجازه قبل تاريخ الاصدار التالي للمنتج.

4. ما هي الغاية من حساب النقاط في نهاية كل مرحلة ؟ (اختر 1 من الإجابات)

- التنبؤ بما يمكن انجازه قبل تاريخ الاصدار التالي للمنتج.
- تنبيه فريق التطوير في حال تراجع أدائه.
- تحفيز فريق التطوير، وتشجيعه على زيادة نقاط المرحلة التالية.
- تعتمد نتيجة المرحلة (نجاح أو فشل)، على مجموع نقاطها.

(التنبؤ بما يمكن انجازه قبل تاريخ الاصدار التالي للمنتج)

خامساً: اجتماع التغذية الراجعة للمرحلة:

في هذا الاجتماع، يتم التعامل مع المرحلة كمصدر لدروس نتعلمها، ونستفيد منها في المراحل اللاحقة. يبدأ الاجتماع عادة بطلب من منسق سكرم لأعضاء الفريق أن يكتب كل منهم ملاحظاته (إيجابية – سلبية – حيادية) على مجموعة من البطاقات، وعندما ينتهي الجميع من الكتابة تتم مناقشة كل ملاحظة دون تجاهل أي منها.

وينتهي الاجتماع بمجموعة قرارات تدون وتبقى للتداول ضمن نطاق الفريق فقط، كون فريق العمل منظم ذاتياً في سكرم.

هذه القرارات قد تكون بسيطة وقد تتطلب مساهمة من خارج الفريق، كأن يطلب الفريق إضافة شخص لمجموعة العمل بخبرة معينة تنقص الفريق.

من الأمثلة على القرارات التي يمكن أن تنتج عن اجتماع التغذية الراجعة:

- نقل الاجتماع اليومي من الساعة 09:00 إلى الساعة 10:00
- الاتفاق على التزام حجم عمل أقل في المراحل التالية
- تجزئ عناصر قائمة ميزات المرحلة أكثر
- اتفاق على طريقة تسمية للمتحويلات في البرمجة وتصميم قاعدة البيانات

اجتماع التغذية الراجعة للمرحلة يتم في نهايتها، ويحضره فريق التطوير مع منسق سكرم فقط، دون أي تدخل من الخارج.

تمارين:

1. أي من الفريقين التاليين تقدّره سكرم أكثر؟ (اختر 1 من الإجابات)

- فريق ذو مهارات استثنائية، وسريع في الانجاز
- فريق قابل للتعلم

(فريق قابل للتعلم)

(الفريق القابل للتعلم يكون فعالاً عند الحاجة)

3. ورقيات سكرم:

قائمة ميزات المنتج

- قائمة مرتبة حسب الأهمية للوظائف المطلوب من النظام تحقيقها
- مرئية لجميع أصحاب المصلحة
- يمكن لأي من أصحاب المصلحة - بمن فيهم أعضاء فرق التطوير - الإضافة إلى هذه القائمة بعد موافقة مالك المنتج
- يمكن إعادة ترتيب عناصر القائمة حيث تمثل العناصر في قمة القائمة العناصر الأكثر أهمية
- يتم تصحيح هذه القائمة دورياً خلال اجتماعات تعديل قائمة ميزات المنتج التي تلي كل مرحلة تطوير
- يمثل كل عنصر من القائمة إحدى الوظائف المطلوبة من النظام
- يمكن التعبير عن كل ميزة مطلوبة، كشرط قبول للنظام

حجم العمل المطلوب لتنفيذ هذه الميزات يتم تقديره من قبل فريق التطوير حصراً، ويمكن أن يضيف فريق العمل تنقيلاً لكل ميزة (عدد من النقاط) تدعى عادة (Story points)، وتقيد هذه النقاط في تقدير الحجم لكل ميزة مقارنة بغيرها من الميزات، ويمكن في الوثيقة التالية (قائمة وظائف المرحلة)، تعديل هذه النقاط لتعكس دقة أكبر ويمكن ترجمتها إلى رجل / شهر (Man/day).

مثال:

كزيون:
<ul style="list-style-type: none"> • يمكن لي أن أدخل للنظام من موقع وب أو من الهاتف المحمول • يمكن لي انشاء حساب جديد وادخال بياناتي الشخصية • في حال نسيت كلمة المرور يمكن أن أطلب محوها وارسال كلمة مرور جديدة عبر البريد الالكتروني •

كصاحب متجر:
<ul style="list-style-type: none"> • يمكن لي أن أدخل للنظام من موقع وب أو من الهاتف المحمول. • يمكن لي انشاء حساب جديد وادخال بياناتي الشخصية. • يمكن أن ادخل مجموعة صور يمكن للزبون استعراضها (Gallery) • يمكن أن أدخل العلامات التجارية التي أتعامل بها مع أيقوناتها

عند صياغة الميزات، يتم عادة ترتيبها حسب اللاعبين الأساسيين (Main actors)، وتعرض بالصيغة التالية (بافتراض أن اللاعب هو الزبون):

- كزيون يجب أن ...
- كزيون أستطيع ...
- كزيون ال حق لي ...

قائمة وظائف المرحلة

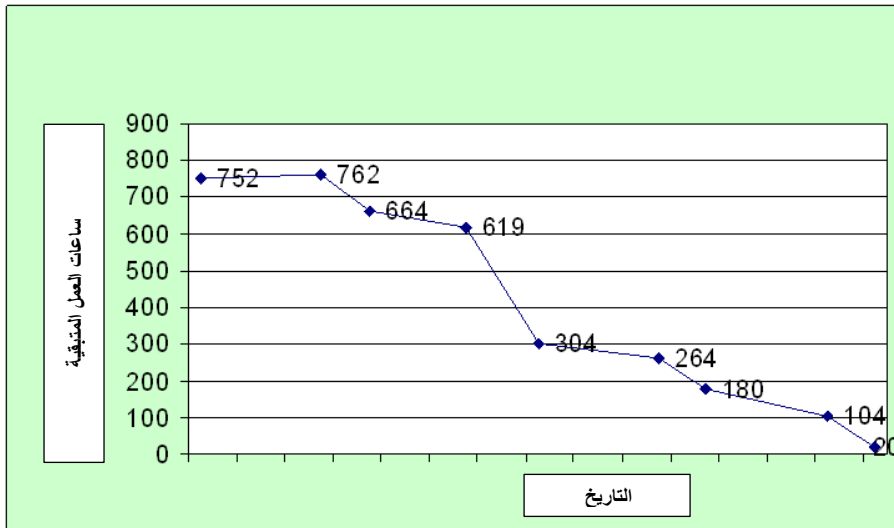
- هي مجموعة من عناصر قائمة ميزات النظام، ينتقيها فريق التطوير ويلتزم بتنفيذها خلال المرحلة
- تتم عملية الانتقاء في اجتماع تخطيط المرحلة
- يقوم فريق التطوير بترجمة عناصر هذه القائمة إلى مهام Tasks
- توضع المهام في عمود المهام التي لم يتم البدء بها (كما هو موضح في الشكل رقم 5)، وتنتقل تبعاً، في الاجتماعات اليومية، إلى عمود المهام قيد التنفيذ ومن ثم إلى عمود المهام المنتهية
- قد يجد فريق التطوير وخلال تنفيذه لمهام المرحلة، مهام جديدة ضرورية لتحقيق هدف المرحلة، عندها يمكن لفريق التطوير إضافة هذه المهام إلى قائمة ميزات المرحلة خلال الاجتماع اليومي لفريق التطوير
- تجزء كل وظيفة من قائمة ميزات المرحلة، إلى مهام صغيرة، لا تتعدى مدة تنفيذ الواحدة منها، يوم عمل (ثمان ساعات)
- اسناد المهام اليومية لأعضاء فريق التطوير، يتم بتطوع أعضاء الفريق لانتقاء مهمة أو أكثر في الاجتماع اليومي، كل حسب خبرته، وحسب ارتباط المهمة بمهام منجزة سابقاً من قبل نفس الشخص
- تضاف لكل مهمة علامة (Story point)، يمكن ترجمتها إلى حجم الجهد المطلوب لتنفيذ هذه المهمة (تقريباً)

مثال:

الزبون:	
يمكن لي أن أدخل للنظام من موقع وب أو من الهاتف المحمول.	
<ul style="list-style-type: none"> • ادخال اسم الحساب وكلمة المرور، والتحقق من وجود الحساب ومطابقة كلمة المرور • في حال الخطأ في الادخال عرض رسالة تفيد بشرح الخطأ للمستخدم (اسم الحساب غير موجود - كلمة المرور خطأ - الاتصال بالمخدم غير متوافر) • في حال صحة الادخالات، اظهار رسالة ترحيب تحوي اسم المستخدم، وعرض صفحة البداية (Home page) • 	

مخطط تعقب المرحلة

- مخطط تعقب المرحلة Sprint Burn down Chart هو مخطط بياني، يعرض يومياً ما تبقى من الجهد، لإنجاز المرحلة، مقدراً بساعات عمل ويعاد رسمه يومياً حسب المتغيرات اليومية في الإنجاز
- عند ترجمة ميزات المرحلة إلى مهام (في اجتماع تخطيط المرحلة)، وعند إضافة مهام جديدة للقائمة، يقدر حجم المهمة (صغيرة، متوسطة أو كبيرة)، ويترجم هذا الحجم لساعات عمل في مخطط تعقب المرحلة (Story points)
- أحياناً يزداد تقدير المتبقي من ساعات العمل بدلاً من أن يتناقص في مخطط التعقب (أي أن المنحني البياني يتصاعد بدلاً من أن يهبط)، وذلك بسبب سوء تقدير حجم المهام وسوء تقدير أعضاء فريق التطوير لإنتاجيتهم، إلا أنه لا يلبث أن يبدأ بالتناقص لينتهي إل الصفر مع نهاية زمن المرحلة في حال نجاحها



(شكل 6)

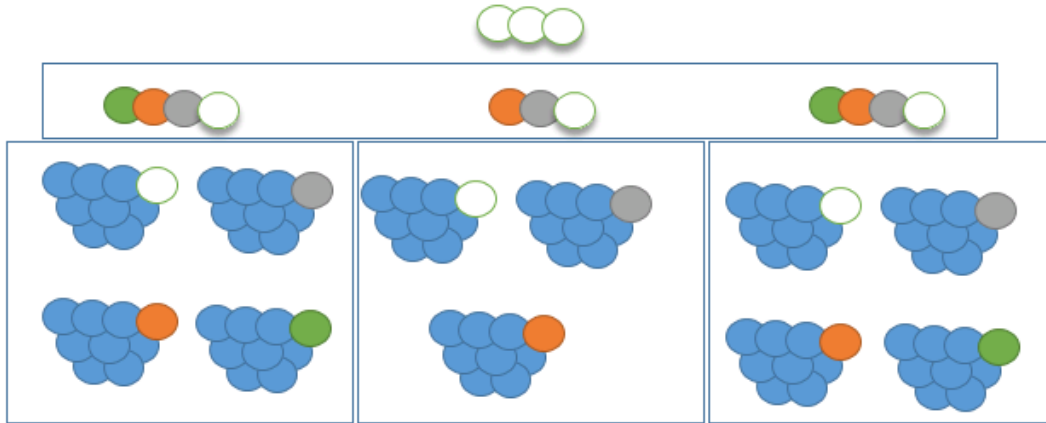
مخطط تعقب الإصدار/النظام

في مخطط تعقب المرحلة يتم تتبع الإنجاز يوماً بيوم، أما في مخطط تعقب الإصدار/النظام Release/Product Burn down Chart فيتم تتبع الإنجاز مرحلة تلو الأخرى، ويقدر المتبقي من الجهد فيه بالأيام بدلاً من ساعات العمل. ومنه يمكن توقع إنهاء النظام أو الإصدار، علماً أن هذا التاريخ ثابت في مخطط الإصدارات، إلا أن التأخير أو الإنجاز المبكر وارد الحدوث، ومن المفيد التنبؤ بحدوثه مسبقاً.

4. سكرم مع عدة فرق تطوير:

فريق التطوير في سكرم يكون عادة من خمس الى تسعة أشخاص، وفي حال لمشاريع الكبيرة التي تتطلب كادراً أكبر حجماً، فإن سكرم لا تزيد عدد أعضاء فريق سكرم بل تزيد من عدد الفرق. وجود فريق من الفرق (Scrum of Scrums) يتطلب جهداً إضافياً للإدارة، وعادة يتم التنسيق بين فرق العمل من خلال اجتماع اضافي هو "اجتماع تنسيق فرق العمل". يقوم كل فريق بانتخاب أحد الأعضاء كمنسق، وتكون احدى مهامه حضور اجتماعات التنسيق مع المنتخبين من الفرق الأخرى.

اجتماع تنسيق فرق العمل مشابه للاجتماع اليومي من حيث القضايا التي يتم طرحها فيه، إلا أنه أقل تواتراً، ويكفي أجرأه مرتين أو ثلاث أسبوعياً. في المثال التالي لدينا أحد عشر فريق عمل، في كل منها تسعة أشخاص، انتخب منهم ثلاث فرق تنسيق، اثنان منها فيها أربعة منسقين والثالث فيه ثلاث منسقين، من هذه الفرق الثلاث تم انتخاب ثلاثة أشخاص للتنسيق على مستوى المشروع بأكمله.



(شكل 7)

5. منهجيات أخرى معتمدة في آجايل:

سكرم هي واحدة من عدة منهجيات معتمدة في آجايل، من المنهجيات الأخرى:

- Adaptive Software Development (ASD)
- Feature Driven Development (FDD)
- Crystal Clear
- Dynamic Software Development Method (DSDM)
- Rapid Application Development (RAD)
- Extreme Programming (XP)
- Rational Unify Process (RUP)

6. المراجع:

1. <http://se.cs.depaul.edu/ise/agile.htm>
2. <http://www.extremeprogramming.org/>
3. <http://www.nebulon.com/articles/index.html>



الفصل الخامس: تدريبات

كلمات المفتاحية:

سكرم، آجايل، مرحلة، تغذية راجعة، مالك المنتج، فريق التطوير، منسق سكرم، منهجية إدارة، إطار عمل، دور وظيفي، إصدار.

ملخص:

في هذا الفصل تدريب على مفاهيم سكرم، من خلال مجموعة اختبارات وشرح بسيط للخيارات الصحيحة.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- مراجعة لمفاهيم منهجية سكرم
- تصحيح المفاهيم الخاطئة عن منهجية سكرم وآليات تطبيقها
- تذكير بعناصر منهجية سكرم (الأدوار – الاجتماعات – الورقيات)

1. مالك المنتج هو المسؤول عن؟ (احذف 1 من الإجابات)

- زيادة العائد على الاستثمار ROI
- الحكم النهائي في أي نقاش يدور حول المتطلبات، وترتيب الأولويات بينها
- يركز على ما هو المطلوب بدلاً من كيفية تحقيقه
- هو المسؤول عن اختيار فريق العمل

2. ما هي العبارة التي تصف سكرم بأفضل شكل؟ (اختر 1 من الإجابات)

- مجموعة اجراءات معرفّة وموصفة بدقة، تنسجم مع طرق الإدارة التقليدية
- اطار عمل يناسب المشاريع غير محددة المواصفات بشكل مسبق
- توثيق لأفضل الطرق والملاحظات في تطوير مشاريع تطوير البرمجيات
- منهجية تصف كيفية بناء التطبيقات المعلوماتية

3. متى تنتهي المرحلة في سكرم؟ (اختر 1 من الإجابات)

- عند تنفيذ جميع المهمات التي تم انتقاؤها كهدف لهذه المرحلة في اجتماع تخطيط المرحلة
- عند انتهاء المدة المقررة لها
- عند انتهاء الاختبارات لما تم انجازه فيها
- يعتمد ذلك على قرار مالك المنتج

4. ما تواتر تكرار اجتماع تنقيح ميزات المنتج؟ (اختر 1 من الإجابات)

- كل مرحلة
- كل اصدار
- في بداية المشروع
- في نهاية المشروع

5. طلب مدير شركة التطوير من أحد أفراد فريق تطوير سكرم، إنجاز مهمة خارج نطاق القائمة المحددة

للمرحلة الحالية، ما التصرف السليم لهذا الفرد للانسجام مع منهجية سكرم؟ (اختر 1 من الإجابات)

- إعلام مالك المنتج، ليناقدش المهمة المطلوبة مع مدير الشركة ويقرر بشأن إدراجها في قائمة مهمات المرحلة
- إضافة المهمة المطلوبة إلى قائمة مهام المرحلة التالية
- إضافة المهمة المطلوبة إلى قائمة مهام المرحلة الحالية
- استبدال إحدى مهام المرحلة الحالية بالمهمة المطلوبة من قبل المدير، على أن تكون مساوية لها تقريباً في حجم العمل اللازم لإنجازها

6. ماذا تتوقع من فريق التطوير انجازه في المرحلة الأولى؟ (اختر 2 من الإجابات)

- وضع خطة عمل لكامل المشروع
- انتقاء مجموعة من ميزات المنتج، وترجمتها إلى مهام وانجازها وتحويلها إلى منتج قابل للتسويق
- تجزيء ميزات المنتج وترجمتها إلى مهمات
- اختبار منتج المرحلة

7. من الذي يقرر ماذا سينجز في المرحلة، ويلتزم بذلك؟ (اختر 1 من الإجابات)

- منسق سكرم
- فريق العمل مجتمعاً
- العضو الأكثر خبرة من أعضاء فريق التطوير
- أحد أعضاء فريق التطوير، يتم انتخابه من قبل منسق سكرم

8. كيف يتم ترتيب قائمة ميزات المنتج؟ (اختر 1 من الإجابات)

- الأهم في الأعلى والأقل أهمية في الأسفل
- عشوائياً
- في فئات فئة أولى، فئة ثانية وهكذا
- العناصر الأكبر حجماً (التي تتطلب جهداً أكبر للإنجاز) في الأعلى والأصغر في الأسفل

9. من يقرر في حجم الجهد المطلوب لإنجاز عنصر من قائمة الميزات؟ (اختر 1 من الإجابات)

- مالك المنتج بمساعدة فريق التطوير
- أكثر أعضاء فريق التطوير خبرة
- منسق سكرم بمساعدة فريق التطوير
- فريق التطوير مجتمعاً بعد مناقشة وإيضاح المتطلبات

10. أي من النقاط التالية هي من مسؤوليات فريق التطوير؟ (احذف 1 من الإجابات)

- التخطيط لإنجاز المرحلة
- حل الخلافات الداخلية في فريق التطوير
- اختيار مالك المنتج
- زيادة الإنتاجية والتعلم المستمر
- التنسيق اليومي أثناء تنفيذ المرحلة، بهدف الوصول إلى غايتها

11. هل تتضمن قائمة ميزات المنتج مهام؟ (اختر 1 من الإجابات)

- نعم
- لا

12. ما هي المهارات المفيدة ليتمتع بها فريق التطوير؟ (اختر جميع الإجابات الصحيحة)

- التحليل وجمع المتطلبات
- البرمجة
- تصميم الواجهات التخاطبية
- هندسة البرمجيات
- قواعد المعطيات
- الاختبار
- التوثيق

13. أثناء تنفيذ المرحلة، متى يتم وضع المهام الجديدة؟ (اختر 1 من الإجابات)

- عند موافقة المنسق عليها
- عند تعريف هذه المهام بشكل واضح، ولا تتناقض مع التزام الفريق بتنفيذ مهام المرحلة في الوقت المحدد لها
- لا يتم اضافة مهام جديدة للمرحلة أثناء تنفيذها
- عندما يتم تعريف هذه المهام من قبل مالك المنتج

14. ماهي المدة الوسطية التي ينصح بها للاجتماع اليومي؟ (اختر 1 من الإجابات)

- 15 دقيقة
- 60 دقيقة
- 5 دقائق
- بالقدر المطلوب لإنهاء النقاشات بين أعضاء فريق التطوير

15. متى يتم انشاء قائمة مهام المرحلة؟ (اختر 1 من الإجابات)

- في اجتماع تخطيط المرحلة
- عند الحاجة
- في بداية المشروع
- في اجتماع تنقيح قائمة ميزات المنتج

16. ما هو الحجم المنصوح به لفريق تطوير سكرم؟ (اختر 1 من الإجابات)

- 2 ± 9
- 2 ± 7
- 2 ± 10
- غير محدود

17. اي من النشاطات التالية لا تساعد مالك المنتج على التعاون مع فريق التطوير؟ (اختر 1 من الإجابات)

- الاستماع إلى مناقشاتهم
- العمل كحكم بينهم
- تشجيع فريق التطوير على ذكر احتياجاتهم
- تعليم مالك المنتج على التقنيات التي يستخدمها فريق التطوير

18. هل سكرم لها قواعد؟ (اختر 1 من الإجابات)

- سكرم لها خطوط عريضة وليس لها أية قواعد
- فقط بعض القواعد البسيطة

19. عادة يقوم أعضاء فريق التطوير بالاختبار؟ (اختر 1 من الإجابات)

- مرة كل 30 يوم
- مرة يومياً
- كلما تم تعديل أي مكون برمجي (ممكن أكثر من مرة في اليوم)
- كلما تم طلب ذلك من فريق الجودة

20. أي من الأشياء التالية لا يشجع على التعاون بين فريق تطوير سكرم وفريق التطوير الأخرى؟
(اختر 1 من الإجابات)

- تعريفهم على بعضهم البعض
- الاستماع إلى نقاشاتهم
- تنظيم لقاءات بينهم في أماكن العمل المختلفة
- العمل كوسيط بين فرق العمل

21. ما هو السبب الأهم لعرض العمل المنجز في سكرم في اجتماع تقييم المرحلة؟ (اختر 1 من الإجابات)

- استخلاص ميزات أخرى للمنتج
- فهم متطلبات المستخدم بشكل أفضل

22. هل هناك مدير مشروع في سكرم؟ (اختر 1 من الإجابات)

- نعم
- لا

23. ماذا يجب أن يحدث في اجتماع تقييم المرحلة؟ (اختر 3 من الإجابات)

- عرض (PowerPoint) لما تم انجازه
- أخذ رأي المستفيدين من المنتج، مما قد يزيد عناصر قائمة ميزات المنتج
- عرض حي لما تم انجازه في المرحلة
- تكريم الأعضاء المتميزين في فريق التطوير
- تقرير عن الاشكالات التي حدثت أثناء فترة المرحلة
- تصريح من قبل مالك المنتج عن أي من عناصر المرحلة تم انجازه، وأيها لم يتم

- 24.** ماذا يحدث لقائمة ميزات المنتج عند الانتقال من مرحلة لأخرى؟ (اختر 1 من الإجابات)
- يزداد حجمها لأن الانتقال من مرحلة لأخرى يكشف عن مزيد من الميزات يجب إضافتها إلى القائمة
 - يصغر حجمها لأن إنتاجية فريق العمل تزداد من مرحلة لأخرى
 - يصغر حجمها لأن بعض الميزات يتم الاستغناء عنها
- 25.** أي من النقاط التالية هي من مسؤوليات الإدارة تجاه فريق تطوير سكرم؟ (اختر 1 من الإجابات)
- تنظيم النشاطات وتوزيعها بين أعضاء فريق التطوير
 - عرض النشاط اليومي لفريق التطوير، وتحفيز الأعضاء المتميزين
 - استعراض انجاز الفريق للتأكد من أنه على المسار الصحيح
 - حل المشاكل والصعوبات التي يتم طرحها من قبل الفريق للإدارة عن طريق المنسق
 - وضع معايير للجودة
- 26.** متى يتم إعادة تنظيم عناصر قائمة ميزات المنتج؟ (اختر 1 من الإجابات)
- بشكل مستمر من قبل مالك المنتج
 - قبل البدء بالمرحلة
 - لا يتم التعديل مطلقاً، لما يفرضه ذلك من كلفة في التعديل
 - قائمة ميزات المنتج يتم قفلها في بداية كل مرحلة
- 27.** متى يتم عقد اجتماع التغذية الراجعة للمرحلة في سكرم؟ (اختر 1 من الإجابات)
- مع نهاية الاصدار
 - مع نهاية المشروع
 - في نهاية كل مرحلة وقبل اجتماع تقييم المرحلة
 - عندما يقرر ذلك مالك المنتج
 - بعد اجتماع تقييم المرحلة

28. في اجتماع التغذية الراجعة للمرحلة، متى يتم أخذ القرارات؟ (اختر 1 من الإجابات)

- في بدء الاجتماع، لتفادي التصادم في وجهات النظر
- أثناء الاجتماع، وعند اقتراح أحد الحاضرين
- في نهاية الاجتماع بعد أن يعبر جميع الحضور عن آرائهم

29. أغلب المؤسسات التي تدعي أنها تطبق سكرم؟ (اختر 1 من الإجابات)

- تطبق سكرم كما هو واجب
- تطبق نسخة معدلة وخاصة بها من سكرم

30. فشل المرحلة في سكرم يعني؟ (اختر 1 من الإجابات)

- إعادتها من البداية
- تمديد مدة تنفيذها
- نقل المهامات غير المنجزة إلى المرحلة اللاحقة
- انشاء مرحلة جديدة تتضمن المهامات الناقصة

الإجابة الصحيحة:

1. (هو المسؤول عن اختيار فريق العمل)
(فريق العمل في سكرم منظم ذاتياً وليس فيه هرمية إدارية، وعادة يتم انتقاؤه من أشخاص منسجمين في العمل مع بعضهم، ويتم الاختيار من قبل إدارة الشركة ومالك المنتج ليس له أي سلطة إدارية عليهم)
2. (إطار عمل يناسب المشاريع غير محددة المواصفات بشكل مسبق)
(في المشاريع غير واضحة المعالم والمواصفات منذ لحظة البدء، كمشاريع المعلوماتية، يؤدي الفهم الخاطئ للمتطلبات إلى تراكم الأخطاء في حال استخدام طرق الإدارة التقليدية، ويصبح التعديل والاختبار مكلفاً جداً في المراحل النهائية من المشروع، وهنا تقدم سكرم بديلاً مناسباً، كون الاستجابة للتعديلات فيها سريعة والتطوير يتم لأجزاء من المنتج وبالتنسيق مع ممثل المستفيد أولاً بأول)
3. (عند انتهاء المدة المقررة لها)
(المرحلة في سكرم محكومة بمدة زمنية، تنتهي بإنتهائها وإعلان فشل المرحلة أو نجاحها من قبل مالك المنتج، وتفشل المرحلة في حال وجود بعض المهام غير المنجزة أو غير المختبرة من قائمة مهام المرحلة، وفي حال فشل المرحلة تتم إعادتها من بدايتها)
4. (كل مرحلة)
5. (إعلام مالك المنتج، ليناقدش المهمة المطلوبة مع مدير الشركة ويقرر بشأن إدراجها في قائمة مهام المرحلة)
6. (وضع خطة عمل لكامل المشروع - اختبار منتج المرحلة)
7. (فريق العمل مجتمعاً)
8. (الأهم في الأعلى والأقل أهمية في الأسفل)
9. (فريق التطوير مجتمعاً بعد مناقشة وإيضاح المتطلبات)
10. (اختيار مالك المنتج)

11. (لا)

(قائمة ميزات المنتج تتضمن ميزات تترجم لاحقاً إلى مهمات في قائمة مهام المرحلة)

12. (جميع الاجابات صحيحة)

13. (عند تعريف هذه المهمات بشكل واضح، ولا تتناقض مع التزام الفريق بتنفيذ مهام المرحلة في الوقت المحدد لها)

14. (15 دقيقة)

15. (في اجتماع تخطيط المرحلة)

16. (2 ± 7)

(مجموعة أصغر قد ينقصها بعض المهارات، ومجموعة أكبر تجعل مهمة التنسيق بين أعضاء فريق التطوير صعبة)

17. (العمل كحكم بينهم)

18. (فقط بعض القواعد البسيطة)

(مالك المنتج لا يمكن أن يكون أحد أعضاء فريق التطوير مثلاً)

19. (كلما تم تعديل أي مكون برمجي (ممكن أكثر من مرة في اليوم))

20. (العمل كوسيط بين فرق العمل)

21. (فهم متطلبات المستخدم بشكل أفضل)

22. (لا)

23. (أخذ رأي المستفيدين من المنتج، مما قد يزيد عناصر قائمة ميزات المنتج - عرض حي لما تم انجازه في المرحلة - تصريح من قبل مالك المنتج عن أي من عناصر المرحلة تم انجازه، وأيها لم يتم)

24. (يزداد حجمها لأن الانتقال من مرحلة لأخرى يكشف عن مزيد من الميزات يجب إضافتها إلى القائمة)

25. (حل المشاكل والصعوبات التي يتم طرحها من قبل الفريق للإدارة عن طريق المنسق)

26. (بشكل مستمر من قبل مالك المنتج)

27. (بعد اجتماع تقييم المرحلة)

28. (في نهاية الاجتماع بعد أن يعبر جميع الحضور عن آرائهم)

29. (تطبق نسخة معدلة وخاصة بها من سكرم)

30. (إعادتها من البداية)



الفصل السادس: الجدولة الزمنية للمشاريع

كلمات المفتاحية:

الجدول الزمني للمشروع، إدارة الوقت، المخطط الشبكي، المخطط القضيب، تقدير احتمالي، ركود الفعالية، المسار الحرج، تقدم المشروع، مقارنة السلسلة الحرجة.

ملخص:

يناقش هذا الفصل مفاهيم أساسية في إدارة المشاريع

أهداف تعليمية:

يهدف هذا الفصل إلى التعريف بـ:

- شرح دورة PDCA
- الجدولة الزمنية للمشروع
- شرح كيفية بناء المخطط الشبكي للمشروع
- شرح كيفية تقدير وقت فعاليات المشروع
- شرح مفهوم ركود الفعالية (Activity Slack)
- شرح كيفية تحديد المسار الحرج للمشروع
- شرح النقاط الأساسية في إدارة تقدم المشروع
- الجدولة الزمنية في آجايل

1. دورة PDCA في إدارة المشاريع:

عادة تدار مشاريع البرمجيات من خلال أربع خطوات تدعى اختصاراً PDCA (Plan-Do-Check-Act).

• خطط (Plan)

- وضع الجدول الزمني للمشروع، وعادةً ما يجري هذا الأمر على مرحلتين، الأولى: وضع جدول زمني أولي، والثانية: وضع خطة تفصيلية اعتماداً على هذا الجدول
- عند وضع الجدول الزمني، من المستحسن وضع معالم (Milestones) لأحداث هامة في المشروع ومراقبة تقدّم هذه الأحداث، و هذا يخفف من المخاطر
- يجب أن يجري خلال مرحلة تخطيط المشروع، إسناد الموارد البشرية اللازمة لكل مهمة وذلك في الخطة التفصيلية

• اعمل (Do)

- سيقوم الأشخاص المسؤولون بعملهم اعتماداً على الجدول الزمني
- يجب أن يكون هناك نظاماً لتوصيل المعلومات المتعلقة بأي مشكلة إلى المدراء بأقصى سرعة

• تحقق (Check)

- يجب على المدراء التحقق من تقدّم تقدم العمل ومقارنته مع الجدول الزمني أولاً بأول، لتدارك التأخير في حال حدوثه، وهنا يجب الانتباه إلى جودة المهام
- أحد الأمور الأساسية في إدارة تقدم تنفيذ الخطة، هو توضيح المستويات المرغوبة من الجودة
- يجب التحقق من جودة المهمة ومن تقدم تنفيذها بنفس الوقت
- يجب الانتباه كذلك إلى الفعاليات الموجودة على المسار الحرج للمشروع (Critical Path)، وهو المسار الذي لا يكون فيه وقتاً ضائعاً (وقت غير مستخدم بشكل تام) (Slack Time). وأي تأخير في أي مهمة من هذا المسار سيؤدي آلياً إلى تأخير التاريخ النهائي للمشروع
- يجب تنفيذ هذه المهام بحذر، بحيث لا يحصل أي تأخير. على سبيل المثال، يجب أخذ المخاطر المحتملة بعين الاعتبار قبل البدء بكل مهمة

• تصرف (Act)

- غالباً ما تحصل بعض الأوضاع غير المتوقعة، والتي تعوق تقدم تنفيذ المهام المجدولة. في مثل هذه الحالات، يجب اتخاذ الإجراءات المضادة بأقصى سرعة ممكنة، مثل تعديل الجدول الزمني وإضافة موارد بشرية
- عندما يجرى تنفيذ خطط مستحيلة (Impossible Plans) أو عندما تحصل العديد من المشاكل الطارئة، قد تصبح المشكلة خارج إمكانات أعضاء المشروع، في مثل هذه الحالات، وإذا كان ذلك ممكناً، يجب الاعتماد على مساعدة خارجية (Outside Assistance)

2. الجدولة الزمنية للمشروع

• المخطط الرئيسي

نبدأ بوضع خطة مفاهيمية (Conceptual Plan) للمشروع، نحدد فيها النقاط الأساسية مثل الإجراءات وتاريخ إنجاز كل منها، بحيث يكون المخطط الكلي للمشروع واضحاً. في مشاريع تطوير البرمجيات، يجب تضمين الخطوات الأساسية الخاصة بإنشاء البرمجيات وتسليمها، في خطة التنفيذ وكذلك توضيح تاريخ إنجاز كل منها.

• بنية تقسيم العمل

الهدف الأساسي من هذه البنية هو معاينة العمل المطلوب وإزالة الأخطاء الموجودة بهدف تقليل المخاطر. هذا يسهل عملية ضبط المشروع من خلال توضيح المسؤوليات وبالتالي تسهيل توجيه التعليمات. تفيد هذه البنية كذلك في حساب الكلفة باستخدام طريقة الجمع أو المراكمة (Accumulation Method).

• المخططات التفصيلية

يجب وضع تخطيط تفصيلي للمهام (عناصر العمل) المحددة في بنية تقسيم العمل. من الممكن تقسيم هذه المهام للحصول على خطط أكثر تفصيلاً. يستحسن وضع الجدولة الزمنية للمهام حسب اليوم أو الأسبوع وليس حسب الشهر.

• مصفوفة مسؤوليات المهام (Task Responsibility Matrix)

تُعرف كذلك بمصفوفة تعيين المسؤوليات (Responsibilities Assignment Matrix)، وتبين من هو الشخص المسؤول عن كل مهمة من المهام.

3. إدارة الجدولة الزمنية للمشروع

تتطلب إدارة الجدول الزمني للمشروع أن يكون هناك إدراك تام لخطة عمل المشروع وللكيفية التي يجب أن تتقدم فيها خطوات المشروع. هناك العديد من الأساليب التي تساعد على توضيح الجدول الزمني وتسهيل متابعته، مثل الجداول التي تمكن من عرض حالة المشروع في لحظة معينة، وكذلك الأدوات البرمجية الخاصة بإدارة المشاريع، مثل برنامج (MS Project).
عموماً يجري بناء الجدول الزمني للمشروع باستخدام أنواع مختلفة من الجداول أو المخططات، منها المخططات الشبكية (Network Chart) والمخططات القضيبيية (Bar Chart).

Bar Chart	Network Chart	
<ul style="list-style-type: none"> البساطة والوضوح إظهار فترة العمل إظهار تقدّم العمل سهولة التصحيح 	<ul style="list-style-type: none"> إظهار العلاقات بين المهام إظهار تأثير التأخير على الجدولة الزمنية 	الإيجابيات
<ul style="list-style-type: none"> عدم إظهار الاعتمادية (العلاقات) بين الفعاليات عدم إظهار تأثير التأخير على الجدولة الزمنية 	<ul style="list-style-type: none"> عدم إظهار تقدّم المشروع صعوبة ترتيب الفعاليات معقّد عندما يكون عدد المهام كبيراً 	السلبيات

وفي حال اعتماد منهجية آجايل لإدارة المشروع، فإن المخططات التي تساعد على تتبع التنفيذ ومقارنته مع الخطة وحساب التأخير، هي مخطط تعقب المرحلة (Sprint burn down chart)، ومخطط تعقب الاصدار (release burn down chart).

4. المخطط الشبكي للمشروع

المخطط الشبكي المعروف باسم بيرت (PERT) هو نموذج شبكي يتميّز بإمكانية وجود تقديرات احتمالية لفترات الفعاليات (المهام). يستخدم هذا المخطط العقد لتمثيل المهام والأسهم لتمثيل علاقات الاعتمادية بين الفعاليات.

تعريف فعاليات المشروع

الفعاليات هي المهام اللازمة لإتمام المشروع، من المفيد أن نقوم بوضع قائمة بكل المهام على شكل جدول، بحيث من الممكن أن نضيف لاحقاً معلومات أخرى إلى هذا الجدول، مثل معلومات تسلسل وتقدير الفعاليات.

تحديد التسلسل المناسب للفعاليات

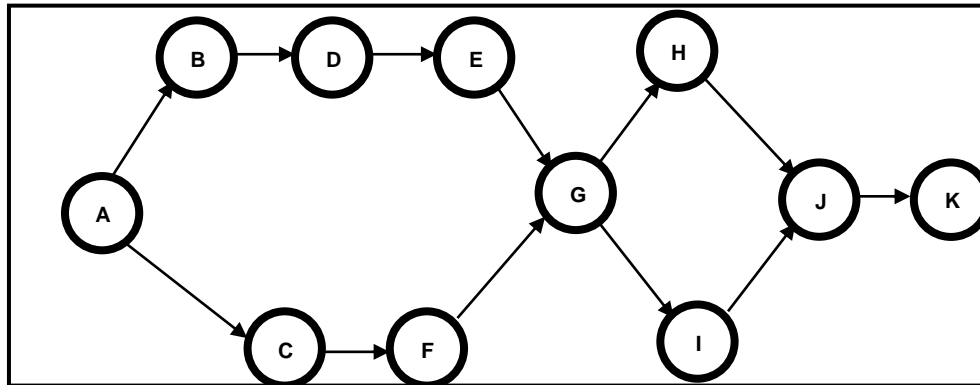
قد يكون التسلسل المناسب لبعض المهام واضحاً، ولكن قد تتطلب بعض الحالات المعقدة تحليلاً دقيقاً لتحديد التسلسل المناسب. يمكن بعد تحديد التسلسل المناسب أن نضيف بعض المعلومات المتعلقة بهذا الأمر إلى جدول.

ويبين الجدول التالي 11 فعالية تم تحديدها لمشروع ما، والتسلسل المناسب لهذه الفعاليات:

الفعالية	توصيف	الفعالية (الفعاليات) التي تسبقها مباشرةً
A	تطوير مواصفات المنتج	لا يوجد
B	تطوير إجرائية التصنيع	A
C	شراء المواد	A
D	شراء التجهيزات والأدوات	B
E	استلام وإعداد التجهيزات والأدوات	D
F	استلام المواد	C
G	الإنتاج الأولي	E&F
H	تقويم تصميم المنتج	G
I	تقويم أداء الإجرائية	G
J	كتابة التوثيق	H&I
K	الانتقال إلى التصنيع	J

بناء المخطط الشبكي

يمكن الاعتماد على معلومات التسلسل الموجودة في الجدول السابق لبناء شبكة الفعاليات. يمكن الاعتماد على برمجيات تقوم بتحويل آلي للجدول السابق إلى المخطط الشبكي المناسب.

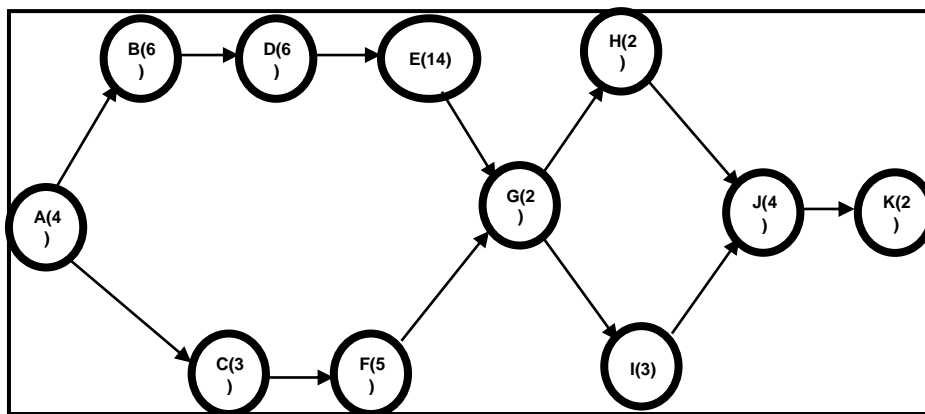


تقديرات محدّدة لوقت الفعاليات

يجري تقدير الوقت اللازم لكل فعالية، يمكن إضافة هذه المعلومات إلى جدول الفعاليات السابق، ليصبح بالشكل التالي:

الفترة (أسبوع)	الفعالية (الفعاليات) التي تسبقها مباشرة	توصيف	الفعالية
4	لا يوجد	تطوير مواصفات المنتج	A
6	A	تطوير إجرائية التصنيع	B
3	A	شراء المواد	C
6	B	شراء التجهيزات والأدوات	D
14	D	استلام وإعداد التجهيزات والأدوات	E
5	C	استلام المواد	F
2	E&F	الإنتاج الأولي	G
2	G	تقويم تصميم المنتج	H
3	G	تقويم أداء الإجرائية	I
4	H&I	كتابة التوثيق	J
2	J	الانتقال إلى التصنيع	K

نحصل في هذه الحالة على المخطط الشبكي التالي:



ويكون لدينا المسارات التالية، وهي مسارات مترابطة (Connected Paths):

1. A, B, D, E, G, H, J, K
2. A, B, D, E, G, I, J, K
3. A, C, F, G, H, J, K
4. A, C, F, G, I, J, K

المسار الحرج

يبين الجدول التالي الفترات اللازمة لإتمام المسارات السابقة:

رقم المسار	مدة المسار
1	40
2	41
3	22
4	23

إن أطول مسار (المسار رقم 2) هو الذي يحدد أو يقيّد فترة المشروع (لا يمكن أن ينتهي المشروع خلال فترة أقل من الفترة اللازمة لإتمام المسار الأطول). يُسمى هذه المسار "الأطول" بالمسار الحرج (Critical Path).

تعريف الفعاليات

يوجد مجموعة من القيم التي تحدّد كل فعالية من الفعاليات:

- **البداية الأبعد (Earliest Start ES)**
هو أقل تاريخ يمكن أن تبدأ فيه الفعالية، يعتمد على تواريخ الانتهاء الأبعد للفعاليات التي تسبقها، وعلى قيود أخرى أو أي تأخير يتعلق بترتيب (أولوية) الفعاليات.
- **الانتهاء الأبعد (Earliest Finish EF)**
هو أقل تاريخ يمكن أن تنتهي فيه الفعالية، يعتمد على تواريخ الانتهاء الأبعد للفعاليات السابقة واللاحقة، وعلى قيود أخرى أو أي تأخير يتعلق بترتيب (أولوية) الفعاليات.
- **المدة المتوقعة للفعالية (Expected Activity Duration)**
بفرض (T) هي المدة المتوقعة للفعالية، فإن:
$$EF = ES + T$$
- **البداية الأكثر تأخيراً (Latest Start LS)**
وهو أكبر تاريخ يمكن أن تبدأ فيه الفعالية بدون أن تؤخر موعد انتهاء المشروع.

- **الانتهاء الأكثر تأخيراً (Latest Finish LS)**

وهو أكبر تاريخ يمكن أن تنتهي فيه الفعالية بدون أن تؤخر موعد انتهاء المشروع، يعتمد على تاريخ البدء المتأخر للفعالية وعلى تواريخ البدء والانتهاء المتأخرة للفعاليات السابقة واللاحقة، وعلى قيود أخرى:

$$LS = LF - T$$

ركود الفعالية

- **ركود الفعالية**

هو مقدار الوقت التي يمكن أن تنقضي خلالها الفعالية قبل أن تؤثر على مهمة أخرى أو على الموعد النهائي للمشروع.

- **الركود الحر (Free Slack)**

هو مقدار الوقت التي يمكن أن تنقضي خلالها الفعالية قبل أن تؤخر مهمة أخرى.

- **الركود الكلي (Total Slack)**

هو مقدار الوقت التي يمكن أن تنقضي خلالها الفعالية قبل أن تؤخر المشروع.

- **المهمة الحرجة (Critical Task)**

هي المهمة التي يجب أن تنهي في الوقت المجدول لها لكي ينتهي المشروع في الوقت المطلوب. إذا تأخرت مهمة حرجة، فقد يتأخر تاريخ إتمام المشروع كذلك. إن سلسلة من المهام الحرجة تشكّل المسار الحرج للمشروع (Critical Path). جميع المهام الموجودة على المسار الحرج، يجب أن يكون ركودها مساوياً للصفر:

$$Slack = LS - ES = LF - EF$$

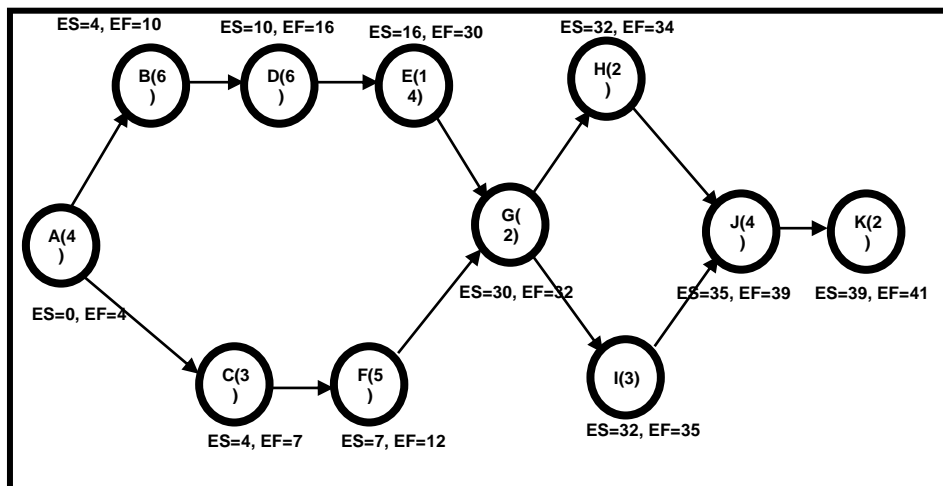
حساب ركود الفعالية – مثال

ليكن لدينا جدول الفعاليات التالي:

الفعالية	الفعالية (الفعاليات) التي تسبقها مباشرة	الفترة المحددة (أسبوع)
A	لا يوجد	4
B	A	6
C	A	3
D	B	6
E	D	14
F	C	5
G	E&F	2
H	G	2
I	G	3
J	H&I	4
K	J	2

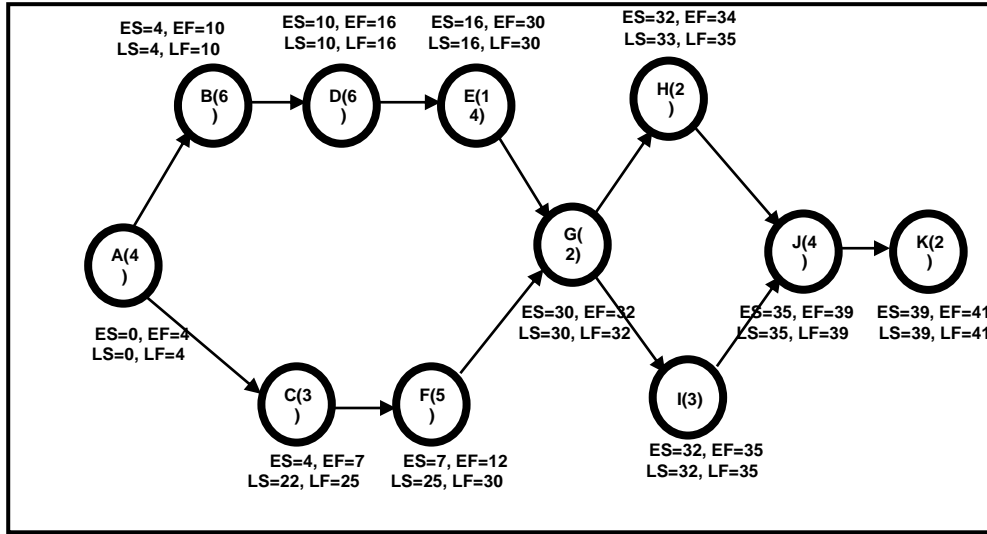
• تحديد أوقات البدء/الانتهاء الأبعد

يجري في المخطط الشبكي التالي تحديد وقت البدء/الانتهاء الأبعد لكل فعالية:



• تحديد أوقات البدء/الانتهاء الأكثر تأخيراً

يجري في المخطط الشبكي التالي تحديد وقت البدء/الانتهاء الأكثر تأخيراً لكل فعالية:



• حساب ركود الفعاليات

يبين الجدول التالي مقدار فترة الركود (Slack) لكل فعالية في المخطط الشبكي:

الركود	الانتهاء المبكر	الانتهاء المتأخر	الفعالية
0	4	4	A
0	10	10	B
18	7	25	C
0	16	16	D
0	30	30	E
18	12	30	F
0	32	32	G
1	34	35	H
0	35	35	I
0	39	39	J
0	41	41	K

تقديرات احتمالية للوقت

• تقديرات احتمالية لفترات الفعاليات

يبين الجدول التالي التقديرات الاحتمالية للفعاليات اعتماداً على التقدير ثلاثي النقط (Three-Point Estimate)، حيث لدينا التقدير التفاولي (Optimistic Estimate) والتقدير الأكثر احتمالاً (Most Likely Estimate) والتقدير التشاؤمي (Pessimistic Estimate):

التقدير التشاؤمي	التقدير الأكثر احتمالاً	التقدير التفاولي	الفعالية
6	4	2	A
10	7	3	B
5	3	2	C
9	7	4	D
20	16	12	E
8	5	2	F
2	2	2	G
4	3	2	H
5	3	2	I
6	4	2	J
2	2	2	K

• حساب الوقت المتوقع للفعالية (Expected Time)

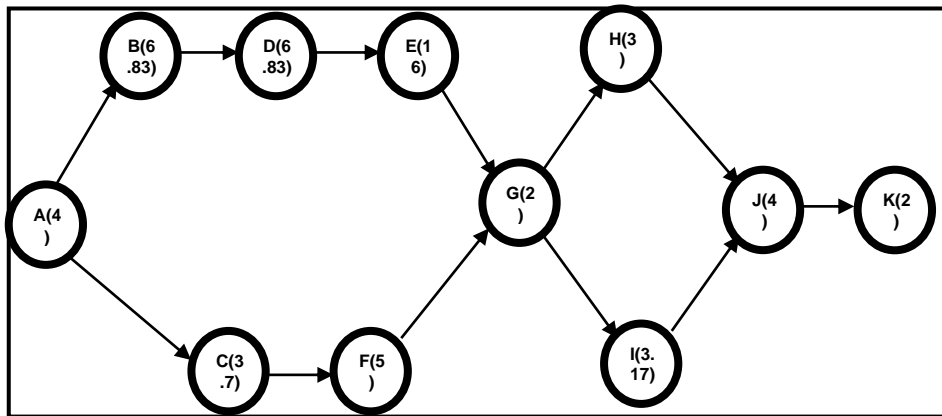
يجري حساب المدة المتوقعة للفعالية باستخدام المعادلة التالية:

$$\text{الوقت المتوقع} = \frac{\text{التقدير التفاولي} + 4(\text{التقدير الأكثر احتمالاً}) + \text{التقدير التшаؤمي}}{6}$$

يبين الجدول التالي الوقت المتوقع للفعاليات:

الفعالية	التقدير التفاولي	التقدير الأكثر احتمالاً	التقدير التшаؤمي	الوقت المتوقع
A	2	4	6	4
B	3	7	10	6.83
C	2	3	5	3.17
D	4	7	9	6.83
E	12	16	20	16
F	2	5	8	5
G	2	2	2	2
H	2	3	4	3
I	2	3	5	3.17
J	2	4	6	4
K	2	2	2	2

• المخطط الشبكي باستخدام الوقت المتوقع للفعاليات



وتكون المسارات المتصلة بالشكل التالي:

1. A, B, D, E, G, H, J, K
2. A, B, D, E, G, I, J, K
3. A, C, F, G, H, J, K
4. A, C, F, G, I, J, K

ويبين الجدول التالي الفترة المتوقعة (Expected Duration) لكل مسار:

رقم المسار	مدة المسار
1	44.66
2	44.83
3	23.17
4	23.34

ويكون المسار رقم 2 هو المسار الحرج المتوقع (Expected Critical Path)، وتكون الفترة المتوقعة للمشروع هي 44.83 أسبوع.

مقاربة السلسلة الحرجة

تركز مقاربة السلسلة الحرجة (Critical chain Approach) على تاريخ انتهاء المشروع بدلاً من التركيز على الفعاليات المفردة، وعلى الحقائق التالية:

- هناك شك في تقدير وقت المشروع، لذلك نحتاج إلى إضافة وقت أمان (Safety Time)
- تقوم بعض المنظمات بإضافة وقت إضافي لتصبح في وضع آمن
- إن إضافة وقت إضافي لكل لفعالية أو ما يُسمى بصوان الفعالية (Activity Buffer)، قد يكون مضيعة للوقت في الفعاليات ذات الأولوية المنخفضة
- المقاربة الأفضل في هذه الحالة هي إضافة صوان الأمان في نهاية المشروع أو ما يعرف بصوان المشروع (Project Buffer)

مثال

المسار الحرج الأصلي:

Activity A	Activity B	Activity C	Activity D	Activity E
------------	------------	------------	------------	------------

والمسار الحرج مع صوان المشروع

Activity A	Activity B	Activity C	Activity D	Activity E	Project Buffer
------------	------------	------------	------------	------------	----------------

إدارة تقدم المشروع

هناك مجموعة من النقاط الأساسية التي يجب مراعاتها عند إدارة تقدم المشروع:

- **تحديد معيار إجرائية العمل وجودته (Work Procedure and Quality Criteria)**
يجب أن يفهم جميع أعضاء المشروع ما هو معيار التقدم في المشروع قبل البدء بتنفيذ المهام. لذلك لا بد من توصيف تفاصيل وتواريخ تقرير تقدم المشروع، بالإضافة إلى الاعتبارات المتعلقة بالجودة والكلفة بحيث تكون واضحة لجميع الأعضاء.
- **التحقق من تقدم المشروع على نحوٍ منظم**
يُعتبر التحقق من تقدم المشروع على نحوٍ منظم أمراً ليس سهلاً، لذلك من الضروري أن يتعاون أعضاء المشروع. كذلك من الضروري أن يجري تحديد صيغ معينة للتقارير، بما يضمن إنشاء التقارير بشكل فعال، وكذلك من الضروري إقامة اجتماعات منتظمة.
- **التحقق من صحة الخطة الزمنية للمشروع**
غالباً ما تتأخر فعاليات المشروع نتيجة لظروف غير متوقعة. كذلك قد تؤدي الخطة المتفائلة إلى تأخير خطير في المشروع، لذلك على مدير المشروع أن يتحقق من الخطة ومن قابليتها للتنفيذ، وكذلك أن يقوم بتعديل هذه الخطة -عند الضرورة- خلال مراقبة تقدم المشروع.
- **التحقق من مهام المسار الحرج**
من الضروري التحقق من تنفيذ مهام المسار الحرج كما يجب، وذلك لاتخاذ الإجراءات المضادة بأسرع وقت ممكن. يجب الانتباه إلى المسار الحرج للمشروع، حتى ولو كان المشروع يتقدم بشكل سليم، لأن أي تأخير في مهمة من المهام الحرجة سيؤدي إلى تأخير المشروع بكامله.

تأخر مهام المسار الحرج

بعض الإجراءات المضادة (Countermeasure) لتأخر مهام المسار الحرج:

- التركيز على مهام المسار الحرج واستثناء المهام الغير الأساسية من ذلك
- إضافة موارد أثناء متابعة المهام المتأخرة
 - عمل إضافي (Overtime Work).
 - أشخاص إضافيين.
 - إجراء عقود جزئية (Subcontractors).
 - تسهيلات إضافية.
 - التحفيز.
- إعادة تعريف (Redefine) أو تضيق النطاق
- تمديد الموعد النهائي للمشروع

5. الجدولة الزمنية للمشروع في آجايل

ينطلق التخطيط وجدولة المشروع في آجايل، من قائمة ميزات المنتج، وهي قائمة توضع بالاتفاق مع الزبون للميزات التي يجب أن يحققها المنتج عند الانتهاء من تنفيذه، هذه الميزات يتم لاحقاً تجزئتها إلى مهام وتوزع على مجموعة مراحل (sprints)، وتوضع لكل منها مجموعة نقاط (Story points)، هذه النقاط تمثل تنقيلاً للجهد اللازم بذله لتحقيق الميزة أو المهمة، وهي تتضمن تنقيلاً للحجم ومستوى الصعوبة ويمكن ترجمتها إلى وقت (رجل ١ شهر).

المراحل في آجايل ضبوطة زمنياً، ويتم انقضاء الميزات التي سيتم تنفيذها في المرحلة بناءً على التقدير الزمني لكل ميزة وبما يتوافق مع الزمن المحدد مسبقاً للمرحلة.

قبل بدء تنفيذ المرحلة (في اجتماع التخطيط للمرحلة)، يقوم فريق التطوير باختيار الميزات التي يلتزم تحقيقها في المرحلة، وفشل المرحلة احتمال وارد وهو أمر يجب على إدارة المشروع أن تأخذه بالحسبان عند تقييم زمن التنفيذ للمشروع وعند وضع خطة المخاطر.

الزمن الكلي لتنفيذ المشروع هو مجموع الأزمنة اللازمة لتنفيذ جميع المراحل والتي تغطي جميع ميزات المنتج.

يتم تتبع الأنجاز ومدى الالتزام بالجدول الزمني من خلال مخطط تعقب المرحلة، ومخطط تتبع الإصدار، ويمكن تقدير نسب الانجاز من مجموع نقاط الميزات والمهام المنفذة نسبة إلى مجموع نقاط الميزات للمرحلة أو للمنتج كاملاً.

6. تدريبات

1. اتحقق من الاتجاز هو عمل دوري لمدير المشروع، ويتضمن بشكل أساسي مقارنة المنجز مع

المخطط، وفي حال انزياح المنجز عن المخطط يرجع مدير المشروع إلى خطة المخاطر؟ (اختر 1

من الإجابات)

- صح
- خطأ

2. في حال اعتماد منهجية آجايل لإدارة المشروع، مصفوفة المهام والمسؤوليات (Task

responsibility matrix)، تظهر لمدير المشروع عن الخطأ أو التأخير في حال حدوثه؟ (اختر 1

من الإجابات)

- صح
- خطأ

3. المسار الحرج في المخطط الشبكي للمشروع؟ (اختر 1 من الإجابات)

- هو المسار الأطول من حيث عدد العقد
- هو المسار الأقصر من حيث عدد العقد
- هو المسار الأطول زمنياً
- هو المسار الأقصر زمنياً

4. ركود المهمة الحرجة يجب أن يكون مساوياً للصفر؟ (اختر 1 من الإجابات)

- صح
- خطأ

5. يتألف مشروع إعداد برمجية من المراحل التالية:

رمز المرحلة	البيان	الأسبوعية	المدة التقديرية (يوم)
A	وضع إطار للمنتج	-	2
B	تحليل المتطلبات	A	10
C	التصميم	B	5
D	التنفيذ	C	30
E	شراء العتاد المادي	A	2
F	التصويب	E,D	1
G	الاختبار	F	5
H	التوثيق	F	10
I	التسليم	G,H	1

طول المسار الحرج هو؟ (اختر 1 من الإجابات)

- 11 يوم
- 59 يوم
- 16 يوم
- 54 يوم

6. ركود المهمة E في المثال السابق مقبول وليس بالضرورة أن يؤدي إلى تأخر المشروع؟ (اختر 1

من الإجابات)

- صح
- خطأ

الإجابة الصحيحة

1. (صح)
(قد تتضمن الحلول المدرجة في خطة المخاطر، زيادة حجم الكادر أو اللجوء إلى بدائل كالعقود الجزئية (Outsourcing))
2. (خطأ)
(لا وجود لمصفوفة المهام والمسؤوليات في آجايل، ومنهجية آجايل تفترض أن فريق التطوير في المرحلة مسؤول بأكمله عن نجاح المرحلة أو فشلها)
3. (هو المسار الأطول زمنياً)
(في المخطط الشبكي لكل عقدة مدة زمنية، وطول المسار هو مجموع أزمان العقد التي يمر فيها، والمسار الحرج هو أطول المسارات زمنياً)
4. (صح)
(التأخر في مهمة حرجة يعني تأخر المشروع)
5. (59 يوم)
(المسار الحرج هو A-B-C-D-F-H-I ومجموع مدة مراحلها هو 59 يوم)
6. (صح)
(المهمة E لا تنتمي للمسار الحرج)



الفصل السابع: إدارة الكلفة

كلمات المفتاحية:

إدارة الكلفة، تقدير الكلفة، تقدير حجم البرنامج، نموذج CoCoMo، قياس البرمجيات، نقطة الوظيفة، تكديس الجهد، نمذجة الجهد، امتداد المشروع، حجم المشروع.

ملخص:

يناقش هذا الفصل أهم القضايا المتعلقة بإدارة كلفة المشروع.

أهداف تعليمية:

يهدف هذا الفصل إلى التعريف بـ:

- شرح دورة PDCA في إدارة الكلفة
- توضيح كيفية تقسيم الكلفة الكلية للمشاريع البرمجية
- شرح الطرق المستخدمة لقياس البرمجيات
- شرح الطرق الرئيسية لتقدير الكلفة في المشاريع البرمجية
- توضيح تغير الكلفة حسب مرحلة المشروع وطريقة التقدير المستخدمة
- شرح القواعد الأساسية في نموذج CoCoMo
- شرح كيفية نمذجة الجهد في نموذج CoCoMo

1. تكاليف المشاريع:

من أهم عناصر المخاطرة في مشاريع تطوير البرمجيات، هو تقدير الكلفة، وتكمن صعوبة تقدير الكلفة في اعتمادها على تقديرات أخرى تخضع لهامش ارتياب كبير (مثل زمن التنفيذ)، وخطورة هذا العنصر هي أن تجاوز الكلفة قد يؤدي إلى إيقاف المشروع وفشله .

وعادة تدار التكلفة في مشاريع البرمجيات من خلال أربع خطوات تدعى اختصاراً PDCA (Plan-Do-Check-Act).

• خطط (Plan)

تبدأ إدارة الكلفة في مرحلة التخطيط للمشروع، وحسب المنهجية المعتمدة في إدارة المشروع (Waterfall or Agile)، يتم تقدير كلف مراحل التنفيذ (التحليل- التصميم- التنفيذ- الاختبار- الصيانة) أو دورات العمل (Sprints). والنتيجة أن الكلفة النهائية للمشروع هي مجموع كلف مجمل المراحل.

• اعمل (Do)

خلال تطور سير العمل، يتوجب على مدير المشروع مقارنة الكلفة المخططة مع حجم الانجاز، ومن خلال المقارنة يمكن له معرفة وتقدير الانزياح ربحاً أو خسارة لتاريخه.

• تحقق (Check)

من مهام مدير المشروع مراقبة وتتبع الكلفة أولاً بأول، وذلك من خلال تقارير دورية تعرض حالة المشروع (Burn down charts).

• تصرف (Act)

نتيجة المقارنة السابقة قد تستدعي إجراءات تصحيح، وهنا يمكن لمدير المشروع الاستعانة بخطة إدارة المخاطر وضمان عدم تجاوز المشروع للكلفة المقررة في الميزانية.

2. تصنيف الكلف في المشاريع البرمجية:

يمكن تمييز الكلف في مشاريع تطوير البرمجيات:

- حسب الزمن إلى كلفة أولية (Initial cost) وكلف جارية (Running cost)
- حسب طبيعة الكلف بين كلف ملموسة (Tangible costs) وكلف غير ملموسة (Intangible costs)
- حسب ارتباطها بالمنتج بين كلفة مباشرة (Direct cost) وكلف غير مباشرة (Indirect cost)

- الكلفة الأولية:
تشمل كلفة شراء العتاد المادي (مخدمات، حواسيب شخصية ...)، التراخيص وأدوات الإدارة (قواعد البيانات، مكتبات جاهزة ...)، ونفقات تطوير البرمجيات.
- الكلف الجارية:
تشمل النفقات اللازمة لصيانة وإدارة النظام المطور (نفقات استضافة سنوية، اشتراك شهري أو سنوي ...)، وتشمل صيانة التجهيزات وكلفة التشغيل والإشراف على الاستثمار .
- الكلفة الملموسة:
تشمل الكلف التي يمكن قياسها بالنقود.
- الكلف غير الملموسة:
يصعب تقديرها بالنقود.
- الكلفة المباشرة:
يمكن ربطها بشكل مباشر بالمنتج أو المشروع (تكاليف يد عاملة، أدوات برمجية ...).
- الكلف غير المباشرة:
ترتبط بشكل غير مباشر بالمنتج (اهتلاك تجهيزات، قرطاسية ...).

3. القياسات الممكنة في مشاريع تطوير البرمجيات

من الضروري وضع أسس معينة تمكن من أخذ قياسات البرمجيات، بحيث يمكن إجراء مقارنات وتقدير كلفة بناءً على هذه القياسات.

القياسات الحجمية للبرمجيات

تُستنتج المقاييس الحجمية (Size-Oriented Metrics) اعتماداً على حجم البرمجية الناتجة. يبين الجدول التالي المقاييس الحجمية في مشاريع تطوير البرمجيات في شركة:

العنصر البشري	العيوب	الأخطاء	الصفحات/ الوثيقة	\$(000)	الجهد	LOC	المشروع
3	29	134	365	168	24	12100	ألفا
5	86	321	1224	440	62	27200	بيتا
6	64	256	1050	314	34	20200	غاما
...

في المشروع ألفا، جرى تطوير 12100 سطراً من الترميز بجهد 24 شخص/شهر، وبكلفة \$168000. يجب الانتباه إلى أن الجهد والكلفة السجلين في الجدول السابق يمثلان جميع مراحل تطوير البرمجية (التحليل والتصميم والترميز والاختبار) وليس فقط الترميز. تُشير المعلومات الأخرى عن المشروع ألفا إلى أنه جرى تطوير 365 صفحة من الوثائق، وتسجيل 134 خطأً قبل إصدار البرمجية، ومصادفة 29 عيباً بعد تسليم البرمجية للزبون خلال العام الأول من التشغيل. وقد عمل ثلاثة أشخاص في تطوير برمجيات المشروع ألفا.

يمكن أن نستنتج، اعتماداً على الجدول السابق، مجموعة من المقاييس الحجمية للمشروع:

- عدد الأخطاء في كل ألف سطر من الترميز
- عدد العيوب في كل ألف سطر من الترميز
- كلفة كل سطر من الترميز
- عدد صفحات الوثائق لكل ألف سطر من الترميز

ويمكن إضافة إلى ذلك حساب مقاييس أخرى مثيرة للاهتمام:

- عدد الأخطاء لكل شخص/شهر
- عدد سطور الترميز لكل شخص/شهر
- كلفة كل صفحة وثائق

ولا تعدّ المقاييس الحجمية عموماً أفضل طريقة لقياس البرمجيات. إذ يدور معظم الجدل حول استخدام عدد أسطر الترميز (LOC) قياساً أساسياً. يدّعي مناصرو هذا القياس أن عدد أسطر الترميز هو نتاج إنساني في جميع مشاريع تطوير البرمجيات ويمكن إحصاؤه بسهولة، وأن العديد من نماذج التقدير البرمجية الحالية تستخدم (LOC) أو (KLOC) دخلاً رئيسياً. من جهةٍ أخرى، يدّعي المعارضون أن قياسات (LOC) تابعة للغة البرمجة المستخدمة في تطوير البرمجية، وأنها تعاقب البرامج المصممة تصميمياً جيداً لكنها أقصر.

المقاييس الوظيفية للبرمجيات

تعتمد المقاييس الوظيفية (Function-Oriented Metrics) على قياس الوظيفة التي تقوم بها البرمجية الناتجة عن المشروع. ولما كان من غير الممكن قياس "الوظيفية" قياساً مباشراً، فلا بدّ من استنتاجها بأسلوب غير مباشر باستخدام قياسات مباشرة أخرى.

ومن هذه القياسات المباشرة "نقطة الوظيفة" (Function Point).
 يجري استنتاج نقاط الوظيفة باستخدام علاقة تجريبية تستند إلى قياسات (مباشرة) قابلة للعد لنطاق
 المعلومات البرمجية، وإلى تقييم تعقيد البرمجيات.
 تُحسب نقاط الوظيفة بإكمال الجدول المبين:

	عامل التثقيل Weighting Factor				العدد count	موسطات أخذ القياسات
	بسيط Simple	متوسط Average	معقد Complex			
=	3	4	6	×		عدد مدخلات المستخدمين
=	4	5	7	×		عدد مخرجات المستخدمين
=	3	4	6	×		عدد استفسارات المستخدمين
=	7	10	15	×		عدد الملفات
=	5	7	10	×		عدد الواجهات الخارجية
	المجموع					

لنشرح مداخل الجدول السابق:

- عدد مدخلات المستخدمين
يُحصى كل دخل (Input) للمستخدم يوفر معطيات تطبيقية التوجُّه مميزة للبرمجية، يجب التمييز بين المدخلات والاستفسارات (Inquiries) التي تحصى على حدة؛
 - عدد مخرجات المستخدمين
يُحصى كل خرج (Output) للمستخدم يوفر معطيات تطبيقية التوجُّه مميزة للمستخدم. ضمن هذه السياق، يشير الخرج إلى التقارير، الشاشات، رسائل الأخطاء وغير ذلك؛
 - عدد استفسارات المستخدمين
يُحصى كل استفسار مميّز، ويعرّف الاستفسار بأنه دخل آني يؤدي إلى توليد استجابة فورية من البرمجية على شكل خرج آني.
 - عدد الملفات
يُحصى كل كلف رئيسي منطقي (أي تجميع منطقي لمجموعات من المعطيات التي قد تكون جزءاً واحداً من قاعدة معطيات واسعة أو ملف مستقل)؛
 - عدد الواجهات الخارجية
تُحصى جميع الواجهات التي يمكن لآلة قراءتها (كملفات المعطيات على الأقراص) والتي تستخدم لنقل المعلومات من نظام لآخر.
- تطور المنظمات التي تستخدم نقاط الوظيفة، معايير لكي تحدّد فيما إذا كان مدخل ما من مداخل الجدول السابق معقداً أو بسيطاً أو متوسطاً. ومع ذلك، فإن تحديد درجة التعقيد هو شيء شخصي إلى حدّ ما.

حساب نقاط الوظيفة

لحساب نقاط الوظيفة نستخدم العلاقة التالية:

$$FP = \text{count-total} \times [0.65 + 0.01 \times \text{SUM}(Fi)]$$

حيث التعداد الكلي هو مجموع جميع المداخل التي حصلنا عليها من الجدول السابق.

إن قيم Fi هي "قيم تعديل درجة التعقيد" ونستند إلى الإجابات على الأسئلة التالية:

- هل يتطلب النظام إجراء نسخ احتياطي دوري واستعادة موثوقين؟
- هل هناك حاجة إلى اتصالات المعطيات؟
- هل تتطلب الوظائف معالجة موزعة؟
- هل تشكّل فعالية الأداء (Efficiency of Performance) أمراً أساسياً؟
- هل ستعمل البرمجية ضمن بيئة تشغيل موجودة وتستخدم بكثافة؟
- هل تتطلب البرمجية إدخالاً آنياً للمعطيات؟

- هل يتطلب إدخال المعطيات المباشر أن يجري بناء مناقلات الدخل (Input Transaction) عبر شاشات أو عمليات متعددة؟
- هل يُحدَّث الملف الرئيسي تحديثاً مباشراً (On-Line)؟
- هل المدخلات أو المخرجات أو الملفات أو الاستفسارات معقّدة؟
- هل المعالجة الداخلية معقّدة؟
- هل تم تصميم الترميز بحيث يمكن إعادة استخدامه؟
- هل يتضمن التصميم عمليتي التحويل (Conversion) والتجهيز (Installation)؟
- هل تم تصميم البرمجية من أجل تجهيز متعدد (Multiple Installation) في مؤسسات مختلفة؟
- هل تم تصميم البرمجية بحيث تُيسّر التغيير وتوفّر سهولة الاستخدام من قبل المستخدم؟

تُستخدم نقاط الوظيفة فور حسابها على وجهٍ مشابه لعدد أسطر الترميز (LOC)، لتنظيم قياسات أخرى للبرمجية، مثل:

- عدد الأخطاء في كل نقطة وظيفة.
- عدد العيوب في كل نقطة وظيفة.
- كلفة كل نقطة وظيفة.
- عدد صفحات الوثائق لكل نقطة وظيفة.
- عدد نقاط الوظيفة لكل شخص/شهر.

4. الطرق الأساسية لتقدير الكلفة في مشاريع تطوير البرمجيات

تشكّل نفقات الأشخاص المشاركين في مشاريع تطوير البرمجيات النسبة الأكبر من نفقات هذه المشاريع. لذلك يتطلب تقدير الكلفة أن يجري تقدير الجهد الذي يقوم به الشخص بشكلٍ أساسي. وهناك عدة طرق تستخدم لهذه الغاية:

- تقدير حجم البرنامج (Program Size Estimation) وهي طريقة لتقدير الجهد اعتماداً على التميز المصدري للبرنامج، تعرف باسم نموذج CoCoMo (Constructive COst MOdel). يجري تقدير الجهد اعتماداً على المواصفات الداخلية (Internal Program Specification) فقط، وبالتالي تكون دقة التقدير (Accuracy of Estimation) منخفضة في المراحل الأولى من تطوير البرمجية، وهي غير مناسبة في هذه المراحل.

- تقدير نقطة الوظيفة (Function Point Estimation) يجري في هذه الطريقة تقدير الجهد اعتماداً على تعقيد البرمجية (Software Complexity)، باستخدام الوحدة "نقطة الوظيفة" (Function Point)، بالإضافة إلى استخدام مواصفات خارجية للبرنامج (External Program Specification).

- طريقة التشابه (Similarity Method) يجري في هذه الطريقة تقدير الجهد والكلفة اعتماداً على النتائج السابقة لمشاريع مشابهة للمشروع الحالي. قد يكون من الصعب -أحياناً- تحديد فيما إذا كانت مشاريع قديمة مشابهة للمشروع الحالي وكذلك تقدير الفرق بينها.

- طريقة التجميع أو المراكمة (Accumulation Method) يجري التقدير في هذه الحالة من خلال التحقق من العمليات وتجميع الجهد المطلوب لها. من الضروري التحقق من جميع العمليات. تعتمد دقة التقدير على دقة العمليات التي يجري التحقق منها، لذلك من الضروري أن يجري تدقيق هذه العمليات في مرحلة مبكرة من إجرائية التطوير.

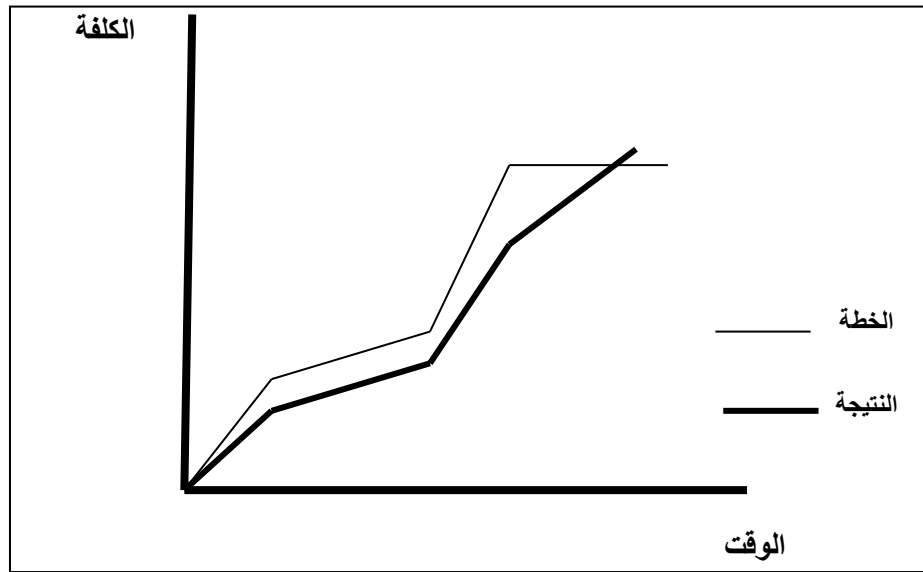
5. دقة تقدير كلفة البرمجيات

- تكون المعلومات المطلوبة لإجراء تقدير الكلفة في المراحل المبكرة من المشروع غامضة عموماً أكثر من المراحل اللاحقة
- يكون الفرق كبيراً بين قيمة التقدير والقيمة الفعلية
- في التقدير المعتمد على حجم البرنامج مثل نموذج CoCoMo يمكن حساب التقدير الدقيق في مرحلة تصميم البرنامج، حيث يكون منطق البرنامج قد أصبح واضحاً
- في تحليل نقطة الوظيفة الذي يعتمد على مواصفات خارجية مثل معلومات دخل وخرج وظيفة معينة، يكون التقدير أكثر دقة في المراحل المبكرة مثل مرحلة التحليل

6. إدارة الكلفة

يجب أن يجري، وعلى نحوٍ منتظم، التحقّق من القيم الفعلية للكلفة مقارنةً مع القيم المخطط لها، بحيث يجري تحديد فيما إذا كانت الكلفة تتغير حسب الخطة. كذلك يجب التحقّق من القيمة الفعلية لكلفة كل عنصر مقارنةً مع الكلفة المخطط لها لهذا العنصر، واتخاذ الإجراءات المناسبة عندما تتجاوز القيمة الفعلية القيمة المخطط لها.

في مشاريع تطوير البرمجيات، ستزداد الكلفة نتيجة لتوسّع نطاق البرمجية الناتج عن تغييرات المواصفات، أو نتيجة لكلف إضافية ناتجة عن تأخر في المهام. لاحظ الشكل التالي:



التحقق من الكلفة الفعلية لكل عنصر كلفة

لتحديد الكلفة الكلية، يجب التحقق من الكلفة الفعلية لكل عنصر كلفة. يمكن استخدام جدول شبيه بالجدول التالي لإدارة التفاصيل المتعلقة بكل عنصر:

شهر أيار		شهر نيسان				
النتيجة	الخطأ	النتيجة	الخطأ			
1,988	3,002	964	0	مخدّمات، حواسيب شخصية، أجهزة شبكات	أجهزة	كلفة أولية
196	200	98		Installation		
0	0	1,500	1,505	شراء برمجيات	برمجيات	
1,002	1,000	1,006	1,000	نفقات شخصية (بالساعة)		
0	0	0	0	Outsourcing		
				كلفة صيانة الأجهزة	أجهزة	كلفة جارية
				كلفة إدارة البرمجيات	برمجيات	

عندما تتجاوز القيم الفعلية القيم المقدّرة، وهذا ما هو متوقع، لن يجري إتمام المشروع ضمن حدود الميزانية إلا إذا جرى اتخاذ الإجراءات المناسبة. لذلك، من الضروري التحقق من العناصر المتعلقة بالكلفة وتحديد العوامل التي تسبب زيادة الكلفة ومن ثم اتخاذ الإجراءات المناسبة.

7. أساسيات نموذج CoCoMo

نموذج (CoCoMo 2.0)

نموذج تقدير برمجي جديد للكلفة والجدولة الزمنية، وهو مناسب للنماذج الجديدة في تطوير البرمجيات، مثل برمجيات الأعمال (Business Software)، البرمجيات الغرضية التوجه (Object-Oriented Software)، نماذج التطوير التطورية أو الحلزونية (Spiral or Evolutionary Development Models)، وغير ذلك.

غايات نموذج (CoCoMo 2.0)

- تطوير نموذج لتقدير الكلفة والجدولة الزمنية للبرمجيات
- تطوير قاعدة معطيات لكلفة البرمجيات وأدوات تدعم المقدرات اللازمة للتحسين المستمر للنموذج
- لتوفير إطار عمل تحليلي كمي (Quantitative Analytic Framework)، ووضع مجموعة من الأدوات والتقنيات لتقويم جهود تحسين تكنولوجيا البرمجيات، وذلك من خلال دورة حياة كلفة ووقت هذه البرمجيات

القطاعات المستقبلية لسوق البرمجيات

يبين المخطط التالي قطاعات سوق البرمجيات المستقبلية:

برمجيات المستخدم (End-User Programming)		
تكامـل الأنظمة (System Integration)	تركيب التطبيقات (Application Composition)	مولدات التطبيقات وأدوات التركيب (Application Generators) (and Composition Aids)
البنية التحتية (Infrastructure)		

نماذج CoCoMo 2.0 الخاصة بقطاعات سوق البرمجيات

- لا يحتاج قطاع برمجيات المستخدم إلى نموذج CoCoMo 2.0
- نموذج تركيب التطبيقات (Application Composition Model)
- مولدات التطبيقات، تكامل الأنظمة، والبنية التحتية
 - نموذج التصميم المبكر (Early Design Model)
 - نموذج البناء البعدي (Post-Architecture Model)

قياس حجم البرمجيات

يستخدم نموذج CoCoMo 2.0 ثلاثة مقاييس مختلفة لقياس حجم المشروع البرمجي:

- نقاط الغرض (Object Points)
- نقاط الوظيفة غير المضبوطة (Unadjusted Function Points)
- أسطر الترميز المصدري (Source Line Of Code SLOC)

8. نمذجة الجهد

الجهد المفترض أو المتصور (Nominal Effort)

يُعطى الجهد المفترض من أجل حجم مشروع ما بالعلاقة التالية:

$$PM \text{ nominal} = A \times (\text{Size})B$$

ويعبر عن هذا الجهد باستخدام الوحدة شخص/شهر (Person/Month PM). سنوضح بقية الرموز المستخدمة في العلاقة السابقة:

- يعرف نموذج CoCoMo عاملاً أسياً من أجل التوفير النسبي (Relative Economies) أو الخسارة النسبية (Relative Diseconomies) للامتداد (Scale) الذي نصادفه عندما يزداد حجم المشروع البرمجي. يجري تمثيل هذا العامل من خلال الأس (B)
- يُستخدم الثابت (A) لتمثيل التأثيرات الخطية (Linear Effects) على الجهد في المشاريع ذات الحجم المتزايد، ويكون (A=2.94)

عوامل الامتداد الأسية (Exponent Scale Factors)

يجري حساب العامل (B) باستخدام المعادلة التالية:

$$B = 0.91 + 0.01 * \sum_{i=1}^5 (W_i)$$

يبين الجدول التالي مستويات التقدير (Rating Levels) الخاصة بعوامل الامتداد الأسية في نموذج CoCoMo 2.0:

يجري جمع التقديرات الرقمية للمشروع (Wi) من أجل كل العوامل، وتستخدم لتحديد عامل الامتداد (B).

عوامل الامتداد (Wi)	منخفض جداً (5)	منخفض (4)	مفترض (3)	مرتفع (2)	مرتفع جداً (1)	مرتفع جداً جداً (0)

مثال 1

إذا كان لدينا مشروع برمجي ذو حجم (100 KLOC)، وتقدير مرتفع جداً جداً (0) من أجل جميع العوامل. سيكون لدينا:

- $W_i = 0$
- $B = 0.91$
- $E = PM = 2.94 * 1000.91 = 2.94 * 66 = 194 \text{ PM}$ (الجهد النسبي)

مثال 2

إذا كان لدينا مشروع برمجي ذو تقدير منخفض جداً (5) من أجل جميع العوامل. سيكون لدينا:

- $W_i = 25$
- $B = 1.16$
- $E = PM = 2.94 * 1001.16 = 2.94 * 209 = 614 \text{ PM}$ (الجهد النسبي)

التوفير والخسارة الناتجة عن الامتداد

- إذا كان ($B < 1.0$)، سيؤدي المشروع توفيراً نتيجة الامتداد. إذا تضاعف حجم المشروع، فإن الجهد اللازم سيكون أقل من ضعف الجهد السابق. تزداد إنتاجية المشروع (Project Productivity) بازدياد حجم المشروع
- إذا كان ($B = 1.0$)، سيكون هناك توازناً بين التوفير والخسارة الناتجة عن الامتداد. يستخدم هذا النموذج الخطي (Linear Model) من أجل تقدير كلفة المشاريع الصغيرة. يستخدم من أجل نموذج تركيب التطبيقات (Applications Composition Model)
- إذا كان ($B > 1.0$)، سيؤدي المشروع خسارة نتيجة الامتداد. هذا يعود بشكل رئيسي إلى سببين أساسيين، نمو عبء التواصل بين الأشخاص (Interpersonal Communications) ونمو عبء تكامل الأنظمة الضخمة (Large-System Integration Overhead)، ويكون للمشاريع الأضخم عدد أكبر من الأشخاص، وبالتالي سيكون هناك مسارات تواصل أكثر بين الأشخاص. إن دمج منتج صغير كجزء من منتج أضخم، يتطلب ليس فقط جهد تطوير المنتج الصغير، وإنما يتطلب جهد إضافي لتصميم ودمج واختبار واجهات هذا المنتج الصغير مع باقي المنتج

9. تدريبات:

1. من أهم الصعوبات التي تميز مشاريع تطوير البرمجيات عن غيرها من المشاريع، صعوبة التحقق وقياس جودة المنتج ؟ (اختر 1 من الإجابات)

- صح
- خطأ

2. من القياسات التي يبنى عليها حساب النقاط الوظيفية ؟ (احذف 1 من الإجابات)

- عدد مدخلات المستخدمين
- عدد الواجهات التخاطبية
- عدد أسطر الرماز المصدري
- عدد استفسارات المستخدمين

3. يستخدم نموذج CoCoMo المقاييس التالية لقياس حجم المشروع ؟ (احذف 1 من الإجابات)

- نقاط الغرض
- عدد الواجهات التخاطبية
- نقاط الوظيفة غير المضبوطة
- أسطر الرماز المصدري

الإجابة الصحيحة:

1. (صح)

(قد يتطلب التحقق من بعض الميزات وقتاً طويلاً، وقد لا تظهر بعض الثغرات إلا بعد فترة تشغيل طويلة للمنتج البرمجي)

2. (عدد أسطر الرمز المصدري)

(راجع الفقرة 3-b)

3. (نقاط الوظيفة غير المضبوطة)

(راجع الفقرة 7-e)

10. المراجع:

1. http://www.tutorialspoint.com/software_engineering/software_project_management.htm
2. http://csse.usc.edu/csse/event/1996/COCOMO/1_Boehm%20%20Tutorial.pdf



الفصل الثامن: إدارة المخاطر

كلمات المفتاحية:

إدارة المخاطر، الاستراتيجيات المنفصلة للمخاطرة، الاستراتيجيات الفاعلة للمخاطرة، الخسارة، مخاطر المشروع، المخاطر التقنية، مخاطر الأعمال، المخاطر المعروفة، المخاطر القابلة للتنبؤ، المخاطر غير القابلة للتنبؤ، تحديد المخاطرة، المخاطرة العمومية، المخاطرة الخاصة بالمنتج، قائمة تفقد عناصر المخاطرة، مكونات المخاطرة البرمجية، توقع المخاطرة، أثر المخاطرة، احتمال المخاطرة، جدول المخاطرة، مراقبة المخاطرة، تخفيف المخاطرة، تجنب المخاطرة.

ملخص:

في هذا الفصل نتعرف على المخاطر في مشاريع تطوير البرمجيات، وخطة إدارة المخاطر والتي توضع عادة في مرحلة التخطيط للمشروع.

أهداف تعليمية:

يهدف هذا الفصل إلى التعريف بـ:

- تعريف المخاطر المحتملة في إدارة مشاريع تطوير البرمجيات
- التعرف على عناصر خطة إدارة المخاطر في مشاريع تطوير البرمجيات

1. إدارة المخاطر:

المقصود بالمخاطر في مشاريع تطوير البرمجيات، هو احتمال الخسارة نتيجة لعدم القدرة على تحقيق الهدف ضمن الموارد المتاحة في أي مرحلة من مراحل تطوير المنتج. عوامل الخطورة لا يمكن إلغاؤها ولكن يمكن إدارتها، وخطة إدارة المخاطر هي عامل حاسم جداً في نجاح مشاريع تطوير البرمجيات كما سنرى لاحقاً. وبالرغم من أن هناك العديد من الاقتراحات لتعريف المخاطرة البرمجية (Software Risk)، إلا أن هناك اتفاق عام على أن المخاطرة تتصف دوماً بصفيتين:

- **عدم اليقين (Uncertainty)**

قد يحصل الحدث المميز للمخاطرة وقد لا يحصل، أي لا يوجد مخاطر احتمالها 100%.

- **الخسارة (Loss)**

إذا أصبحت المخاطرة حقيقية فإن التبعات أو الخسائر الناتجة عن ذلك ستحصل.

2. مفاهيم أساسية:

تقدير حجم المشروع:

من أصعب المهام التي تواجه مدير المشروع، هي تقدير الحجم النهائي للمشروع، والذي يؤثر بدوره على تقدير زمن التنفيذ والكلفة.

من المؤشرات التي يمكن تقدير حجم المشروع انطلاقاً منها: عدد الوثائق الناتجة عنه (التقارير والاحصائيات)، عدد الواجهات التخاطبية، عدد اجراءات العمل المطلوب أتمتها.

إلا أن الوثائق الناتجة عن المشروع تختلف في الحجم والإخراج وصعوبة التوليد، ونفس الأمر ينطبق على الواجهات التخاطبية. وإجراءات العمل يمكن أن تكون بسيطة أو مركبة، والنتيجة أن هذه العوامل الثلاث مجتمعة يمكن أن تعطي تقديراً لحجم العمل، وإذا أضفنا لها خبرة مدير المشروع وقدرته على التنبؤ بالمشكلات فيمكن لمدير المشروع وضع خطة زمنية للمشروع بأقل قدر من الارتياح.

على الرغم من ذلك يبقى تقدير حجم المشروع والوقت اللازم لتنفيذه من الأخطار المحتملة، ولا بد لمدير المشروع من تتبع التنفيذ أولاً بأول وإعادة تقييم وقت التنفيذ والقدرة على الالتزام بالخطة الزمنية، ولا بد له من وضع خطة إدارة مخاطر يلجأ للتلو المقتوحة فيها عند ظهور أي تأخير.

من الاجراءات المحتملة والممكن ادراجها كحلول للتأخير في خطة إدارة المخاطر، زيادة حجم كادر العمل (يجب أن يكون لدى مدير المشروع قاعدة معطيات بأشخاص محتملين بكفاءات مناسبة ممكن ضمهم لفريق العمل عند الحاجة)، أو زيادة ساعات العمل (عمل إضافي)، أو الاستعانة باستشاريين لتقديم حلول لمشاكل تؤخر سير العمل.

الكلفة:

عند تقدير كلفة المشروع، لا بد من أن نأخذ بعين الاعتبار العوامل التالية:

- تقدير حجم الشروع
- جودة المنتج النهائي
- العتاد المادي اللازم
- الأدوات البرمجية اللازمة والترخيص المطلوبة
- المهارات الشخصية المطلوبة في فريق العمل
- نفقات السفر والإقامة في حال وجودها
- الاتصالات
- التدريب والدعم الفني

عوامل الخطورة:

يمكن تصنيف عوامل الخطورة في مشاريع تطوير البرمجيات ضمن ثلاث فئات:

- مخاطر المشروع (Project Risks)
 - عدم الالتزام بالخطة الزمنية للمشروع.
 - الانزياح في الجدول الزمني للمشروع يؤدي إلى زيادة الكلفة.
 - الكادر، الموارد، الزيون، ومشاكل المتطلبات.
- المخاطر التقنية (Technical Risks)
 - تهدد جودة ودقة وتوقيت البرمجية المطلوب إنتاجها
 - يصبح تحقيق البرمجية صعباً أو مستحيلاً
 - تحدّد المشاكل المحتملة في التصميم، التحقيق، الواجهات، الاختبار، والصيانة
 - عوامل أخرى للمخاطرة في هذا الإطار
 - غموض المواصفات (Specification Ambiguity)
 - عدم اليقين التقني (Technical Uncertainty)
 - التقادم التقني
 - ظهور التقنيات الجديدة

- مخاطر الأعمال (Business Risks)
 - بناء منتج أو نظام ممتاز لا يرغب فيه أحد (مخاطرة السوق)
 - بناء منتج لم يعد متناسباً مع الإستراتيجية العامة لأعمال الشركة (مخاطرة إستراتيجية)
 - بناء منتج لا يعرف فريق المبيعات كيفية بيعه أو تسويقه
 - فقدان دعم الإدارة العليا بسبب تغيير اهتمامها أو إجراء تغيير في الأشخاص (مخاطرة إدارية)
 - فقدان توفر الميزانية أو الموظفين (مخاطرة الميزانية)

ويمكن تصنيف عوامل الخطورة حسب إمكانية توقعها ضمن ثلاث فئات:

- المخاطر المعروفة (Known Risks)

هي المخاطر التي يمكن اكتشافها بعد تقييم متأنٍ لخطة المشروع، ولبينة الأعمال والبيئة التقنية التي يجري فيها تطوير المشروع، والمصادر الموثوقة الأخرى للمعلومات (مثل: تاريخ توريد غير معقول، افتقار للمتطلبات الموثقة أو لنطاق البرمجية، أو بيئة تطوير فقيرة)؛
- المخاطر القابلة للتنبؤ (Predictable Risks)

تستخلص من الخبرة السابقة، مثل تغيير العاملين، اتصالات ضعيفة مع الزبون، تخفيف جهود الموظفين مع تخديم طلبات الصيانة المستمرة.
- المخاطر غير القابلة للتنبؤ (Unpredictable Risks)

وهذه تشبه "جوكر" ورق اللعب، يمكن أن تحدث، وقد تحدث فعلياً ولكن يصعب فعلياً تعيين هويتها مقدماً.

أهم عوامل الخطورة والتي تتكرر في كثير من مشاريع تطوير البرمجيات:

- خروج المطورين ذوي الخبرة من فريق العمل، ودخول أعضاء جدد
- تغير البنية التنظيمية للمشروع
- تغير المتطلبات الوظيفية
- التقدير السيء لزمان التنفيذ
- تغير الأدوات والبنى التحتية

3. خطة إدارة المخاطر:

يمكن تلخيص محتوى خطة إدارة المخاطر، بأربعة موضوعات هي:

- **تحديد المخاطر**
تحديد المخاطر هي إجرائية لتوصيف التهديدات التي تعترض خطة المشروع (التقديرات، الجدول الزمني، تحميل الموارد، ...). عندما يقوم مدير المشروع بتعيين هوية المخاطر، فإنه يكون قد خطا الخطوة الأولى باتجاه تجنبها عند الإمكان والتحكم فيها أيضاً عند الضرورة.
- **تقييم المخاطر**
تقييم احتمال حدوث كل مخاطرة وتأثيرها على المشروع.
- **وضع الإجراءات المضادة**
تحديد المخاطر ذات التأثير الكبير على المشروع، ومحاولة التخفيف منها ووضع الإجراءات المناسبة لذلك.
- **مراقبة وإدارة المخاطر**
مراقبة آلية التعامل مع المخاطر، وتوثيق جميع المعلومات المتعلقة بذلك وبالمشاكل التي قد تنتج عن مخاطرة معينة، واعتماد نظام سليم للتواصل بين أعضاء فريق المشروع.

4. تحديد المخاطر:

إحدى طرائق تحديد المخاطر هي أن نقوم بإنشاء قائمة حصر لعناصر المخاطرة (Risk Item Checklist)، من البنود الممكن ادراجها في هذه القائمة:

المخاطر المتعلقة بالحجم الكلي للبرمجية المراد بناؤها أو تعديلها.	Product Size	حجم المنتج أو المشروع	PS
المخاطر المتعلقة بالقيود التي تفرضها الإدارة أو يفرضها مكان التسويق.	Business Impact	أثر الأعمال	BI
المخاطر المتعلقة بمستوى تمرس الزبون وقابلية المطور للاتصال بالزبون بطريقة مخططة زمنياً.	Client Characteristics	مزاي الزبون	CC
المخاطر المتعلقة بالدرجة التي وصل إليها تعريف الإجراءات البرمجية والتي تتبّعها مؤسسة التطوير.	Process Definition	تعريف إجراءات العمل	PD
المخاطر المتعلقة بتوفر جودة الأدوات المستخدمة لبناء المنتج.	Development Environment	بيئة التطوير	DE
المخاطر المتعلقة بتعقيد النظام المطلوب بناؤه ومستوى التقنيات المطلوب توفيرها في النظام.	Required Technology	التقنيات المطلوبة	RT
المخاطر المتعلقة بخبرة مهندسي البرمجيات الذين سيقومون بالعمل فيما يخص خبرتهم التقنية الكلية وخبرتهم في المشروع.	Staff Size and Experience	حجم الكادر وخبرته	SS

مخاطر حجم المشروع:

- قائمة حصر عناصر المخاطرة
 - الحجم التقديري للمنتج مقياساً باستخدام عدد أسطر الترميز (LOC) أو نقاط الوظيفة (FP)
 - درجة الثقة في تقدير الحجم التقديري للبرمجية
 - الحجم التقديري للمنتج مقياساً بعدد البرامج والملفات والمناقلات
 - الانزياح النسبي (Relative Shift) في حجم المنتج عن وسطي المنتجات السابقة
 - حجم قاعدة المعطيات (Database) التي يُنشئها أو يستخدمها المنتج
 - عدد مستخدمي المنتج
 - عدد التغيرات المتوقعة في متطلبات المنتج، قبل التسليم، وبعد التسليم
 - مقدار البرمجيات التي أُعيد استخدامها
- يجب في كل حالة من هذه الحالات إجراء مقارنة معلومات المنتج المطلوب تطويره بالخبرة السابق
- إذا حصلت نسبة انزياح كبيرة، أو كانت الأرقام متماثلة ولكن النتائج السابقة أقل بكثير من المقبول، فإن احتمال المخاطرة عالٍ جداً

مخاطر أثر الأعمال:

- قائمة حصر عناصر المخاطرة
 - أثر هذا المنتج في عائدات الشركة
 - وضوح هذا المنتج في نظر الإدارة العليا
 - إمكانية الالتزام بالمواعيد النهائية للتسليم (هل هذه مواعيد معقولة ومنطقية؟)
 - عدد الزبائن الذين سيستخدمون هذا المنتج وانسجام احتياجاتهم معه
 - درجة تمرُّس المستخدم النهائي
 - مقدار وجودة توثيق المنتج اللازم توفيرها وتسليمها للزبون
 - القيود الحكومية على بناء المنتج (إن وجدت)
 - الكلفة المترتبة على التسليم المتأخر
 - الكلفة المترتبة على منتج فيه عيوب
- يجب في كل حالة من هذه الحالات إجراء مقارنة معلومات المنتج المطلوب تطويره بالخبرة السابقة
- إذا حصلت نسبة انزياح كبيرة، أو كانت الأرقام متماثلة ولكن النتائج السابقة أقل بكثير من المقبول، فإن احتمال المخاطرة عالٍ جداً

الزبون:

- تختلف الاحتياجات من زبون لآخر:
فزبون يعرف ما يريد، وآخر يعرف ما لا يريد. وزبون يبذل قصارى جهده لمعرفة التفاصيل، على حين يكتفي زبون آخر بالوعد حتى لو كانت غير دقيقة.
- تختلف الشخصية من زبون لآخر:
فزبون يستمتع بكونه زبوناً، ويستمتع بالمفاوضات وبالنتائج السيكولوجية للمنتج الجديد، بينما يفضل آخر عدم كونه زبوناً. وزبون يقبل سعيداً أي شيء يُسلم إليه، ويجعل أسوأ منتج هو الأفضل، في حين يشنكي آخر بأسى عندما يفنقر المنتج إلى الجودة. ويبيد البعض تقديره عندما تكون الجودة عالية، وقلة منهم يشنكون لمجرد الشكوى بقطع النظر عن الأسباب.
- تختلف الصلة بالموردين من زبون لآخر:
فزبون يعرف جيداً المنتج والمنتج، وآخر لا يتواصل مع المنتج إلا بمراسلات مكتوبة، وبعض المكالمات الهاتفية المختصرة.
- غالباً ما يناقض الزبون نفسه:
فهو يريد الحصول على كل شيء "البارحة" ومجاناً. غالباً ما يعاني المنتج من تناقض الزبون مع نفسه.
- قائمة حصر عناصر المخاطر المتعلقة بالزبون
 - هل عملت مع الزبون في الماضي؟
 - هل يمتلك الزبون فكرة واضحة عن المطلوب؟ هل قضى الزبون وقتاً كافياً لكتابتها؟ أو هل كتبها بالأصل؟
 - هل سيوافق الزبون على قضاء وقت في اجتماعات رسمية لجمع المتطلبات بهدف تحديد نطاق المشروع؟
 - هل الزبون مستعد لتأسيس قنوات اتصال سريعة مع المطور؟
 - هل الزبون متمرس تقنياً في مجال المنتج؟
 - هل الزبون مستعد ليدع موظفك يقومون بعملهم، أي هل سيقاوم الزبون نفسه بألا يقف فوق رأسك خلال الأعمال التقنية التفصيلية؟
 - هل يفهم الزبون الإجرائية البرمجية (Software Process) التي يتبناها الفريق لتطوير المنتج البرمجي؟
- إذا كان الجواب عن أيٍّ من الأسئلة السابقة نفيًا، فيجب إجراء تقصُّ آخر لتقييم احتمال المخاطرة

تعريف اجرائيات العمل:

- هل تدعم إدارتك العليا سياسة مكتوبة تؤكد أهمية الإجرائية القياسية لهندسة البرمجيات؟
- هل طوّرت مؤسستك وصفاً مكتوباً للإجرائية البرمجية التي ستطبق في هذا المشروع؟
- هل يحدّد أعضاء الكادر المكلف بالعمل الإجرائية البرمجية كما هي موثقة ومستعدّون لاستخدامها؟
- هل تُستخدم الإجرائية البرمجية لمشاريع أخرى؟
- هل طورت مؤسستك أو طلبت سلسلة من الدورات التدريبية في هندسة البرمجيات للمديرين والكادر التقني؟
- هل تُقدّم معايير منشورة لهندسة البرمجيات لكل مطوّر برمجيات أو مدير برمجيات؟
- هل تجرى المراجعات التقنية الرسمية (Formal Technical Reviews) لتوصيف المتطلبات والتصميم والترميز بشكل منتظم؟
- هل تُجرى المراجعات التقنية الرسمية للإجراءات الاختبار وحالات الاختبار بشكل منتظم؟
- هل توثّق نتائج كل مراجعة تقنية رسمية ، بما في ذلك الأخطاء الموجودة والموارد المستخدمة؟
- هل توجد آلية ما لضمان توافق العمل المنفّذ ضمن المشروع مع قياسات هندسة البرمجيات؟
- هل استُخدمت آلية للتحكم في تغيير متطلبات الزبون التي تؤثر في البرمجية؟
- هل يوجد توثيق لبيان العمل، توصيف متطلبات البرمجية، وخطة تطوير البرمجية، وغيرها، وذلك لكل عقد فرعي؟
- هل يُتبع إجراء ما لمتابعة ومراجعة أداء المتعاقدين الفرعيين؟
- هل تُستخدم طرائق محدّدة لتحليل البرمجيات؟
- هل تُستخدم طريقة محدّدة لتصميم المعطيات وتصميم بنية البرمجية؟
- هل كُتِبَ أكثر من 90% من ترميز برنامجك بلغة عالية المستوى؟
- هل عُرِّفت واستخدمت مصطلحات محدّدة لتوثيق الترميز؟
- هل تُستخدم طرائق محدّدة لتصميم حالات اختبار البرمجية؟
- هل تُستخدم أدوات برمجية لدعم فعاليات التخطيط والمتابعة؟
- هل تُستخدم أدوات برمجية لدعم إجرائية تحليل البرمجيات وتصميمها؟
- هل تُستخدم أدوات لإنشاء نماذج أولية (Prototype) برمجية؟
- هل تُستخدم أدوات برمجية لدعم إنتاج الوثائق وإدارتها؟
- هل تُجمع مقاييس الجودة لجميع المشاريع البرمجية؟
- هل تُجمع مقاييس الإنتاجية لجميع المشاريع البرمجية؟

- إذا أُجيب بالنفي عن معظم هذه الأسئلة، فإن الإجرائية البرمجية ضعيفة ويكون احتمال المخاطرة عالياً. يجب الانتباه إلى أن قائمة تفقّد عناصر المخاطرة المتعلقة بالإجرائية قد تختلف من مشروع إلى آخر ولكن النقاط المحددة سابقاً تعتبر أكثر شيوعاً وأهمية بين المشاريع البرمجية

بيئة التطوير:

- تدعم بيئة هندسة البرمجيات فريق العمل والإجرائية والمنتج، إذا كان هناك عيوب في هذه البيئة، فقد تكون مصدر مخاطرة كبيرة
- قائمة حصر عناصر المخاطر المتعلقة ببيئة التطوير
 - هل تتوفر الأدوات المناسبة لإدارة المشروع البرمجي؟
 - هل تتوفر الأدوات المناسبة لإدارة الإجرائية البرمجية؟
 - هل تتوفر الأدوات المناسبة للتحليل والتصميم؟
 - هل تتوفر مترجمات (Compilers) أو مولّدات ترميز (Code Generator)، مناسبة للمنتج المراد بناؤه؟
 - هل تستخدم البيئة قاعدة معطيات أو مخزن (Repository)؟
 - هل جميع الأدوات البرمجية المستخدمة متكاملة مع بعضها؟
 - هل جرى تدريب أعضاء فريق المشروع على الأدوات التي تلائم مهام كل عضو؟
 - هل هناك خبراء محليين للإجابة عن الأسئلة المتعلقة بالأدوات؟
 - هل هناك توثيق كافٍ لهذه الأدوات؟
 - هل المساعدة المتاحة (بطريقة ما، عبر الهاتف، الإنترنت، ...) كافية؟
- إذا كانت الإجابة عن معظم هذه الأسئلة بالنفي، فإن بيئة تطوير البرمجيات ضعيفة واحتمال المخاطرة عالٍ جداً

التقنيات المطلوبة:

- قائمة حصر بنود المخاطر المتعلقة بالتكنولوجيا
 - هل التكنولوجيا التي ستبنى جديدة على المؤسسة؟
 - هل تحتاج متطلبات الزبون إلى بناء خوارزميات جديدة أو تقنية دخل أو خرج جديدة؟
 - هل للبرمجيات واجهة مع أجهزة جديدة أو غير مجرّية؟
 - هل للبرمجيات التي ستبنى واجهة مع نظام قواعد معطيات لم تثبت بعد وظيفته وأداؤه في مجال التطبيق المراد بناؤه؟
 - هل تحتاج متطلبات المنتج إلى واجهة مستخدم خاصة؟
 - هل تحتاج متطلبات المنتج إلى استخدام طرائق جديدة للتحليل والتصميم والاختبار؟
 - هل تحتاج المتطلبات إلى استخدام طرائق غير تقليدية لتطوير البرمجيات، مثل الطرائق الصورية (Formal Methods)؟ والطرائق المعتمدة على الذكاء الاصطناعي (Artificial Intelligence)، والشبكات العصبونية (Neural Networks)؟
 - هل تضع المتطلبات قيود أداء فائقة على المنتج؟
 - هل الزبون غير متأكد من إمكانية تحقيق الوظيفة المطلوبة؟
- إذا كان الجواب على أي من هذه الأسئلة بالإيجاب (نعم)، فيجب إجراء تحليل إضافي لتقييم احتمال المخاطرة

حجم الكادر وخبرته:

- قائمة حصر عناصر المخاطر المتعلقة بحجم وخبرة الكادر
 - هل يتوفر أفضل العاملين؟
 - هل يمتلك العاملون المزيج المناسب من المهارات؟
 - هل عدد العاملين كافٍ؟
 - هل العاملون ملتزمون بالعمل طوال مدة المشروع؟
 - هل سيعمل بعض العاملين جزئياً في هذا المشروع؟
 - هل لدى العاملين التوقعات أو التصورات الصحيحة عن العمل الذي يعملون فيه؟
 - هل تلقى العاملون التدريب الضروري؟
 - هل سيكون تغيير العاملين منخفضاً إلى درجة تسمح بالاستمرار؟
- إذا كان الجواب على أي من هذه الأسئلة بالنفي، فيجب إجراء المزيد من التحليل والتقصي لتقييم احتمال المخاطرة.

5. تقييم المخاطر:

مكونات المخاطرة البرمجية:

يمكن تعريف مكونات المخاطرة البرمجية على النحو التالي:

- مخاطرة الأداء (Performance Risk)
- وهي درجة الشك في أن يتوافق المنتج مع المتطلبات المطلوبة منه، وأن يناسب الاستخدام المخطط له
- مخاطرة الكلفة (Cost Risk)
- وهي درجة الشك في أن يتحقق الالتزام بموازنة المشروع
- مخاطرة الدعم (Support Risk)
- وهي درجة الشك في أن تكون البرمجيات سهلة التصحيح والتكيف والتحسين
- مخاطرة الجدول الزمني (Schedule Risk)
- وهي درجة الشك في أن يحافظ على الجدول الزمني للمشروع وأن يُسَلَّم المنتج في موعده

ومن المفيد أن يقوم مدير المشروع بتعريف مكونات المخاطرة البرمجية (الأداء، الكلفة، الدعم، الجدول الزمني).

ويُصنَّف أثر كل من مكونات المخاطرة، في واحدة من أربع فئات تأثير:

- تافه (Negligible)
- هامشي (Marginal)
- حرج (Critical)
- كارثي (Catastrophic)

ويبين الجدول التالي النتائج الممكنة للأخطاء (الأسطر ذات الرقم 1) أو الإخفاق في تحقيق الخرج المطلوب (الأسطر ذات الرقم 2) . يجري اختيار فئة التأثير اعتماداً على التوصيف الأكثر ملائمة للوصف في الجدول:

		الأداء	الدعم	الكلفة	الجدول الزمني
كارثي	1	الإخفاق في تحقيق المتطلبات ينتج عنه إخفاق المهمة		ينتج عن الإخفاق كلف زائدة وتأخير في البرنامج بقيم تقديرية تفوق \$500K	
	2	تراجع كبير عائد لعدم تحقيق الأداء التقني	برمجيات غير مستحبة أو غير قابلة للدعم	قصور مالي كبير، احتمال تجاوز الميزانية	موعد تسليم غير قابل للتحقيق
حرج	1	يؤدي الإخفاق في تحقيق المتطلبات إلى تراجع أداء النظام إلى درجة يصبح فيها نجاح المهمة موضع شك		ينتج عن الإخفاق تأخيرات تشغيل و/أو زيادة في الكلفة بقيمة تقديرية من \$100K إلى \$500K	
	2	بعض التخفيض في الأداء التقني	تأخيرات ثانوية في تعديلات البرمجيات	بعض النقص في الموارد المالية، وتجاوز محتمل لها	انزلاق محتمل في تاريخ التسليم
هامشي	1	الإخفاق في تحقيق المتطلبات ينتج عنه تراجع في المهمات الثانوية		انزلاق في الكلفة، انزلاق للجدول الزمني قابل للاستدراك بقيمة تقديرية من \$1K إلى \$100K	
	2	تخفيض أصغري إلى صغير في الأداء التقني	دعم استجابي للبرمجيات	موارد مالية كافية	جدول زمني معقول قابل للتحقيق
تافه	1	الإخفاق في تحقيق المتطلبات يخلق أثراً غير مناسب أو غير قابل للتشغيل		ينتج عن الخطأ أثر جزئي في الكلفة و/أو الجدول الزمني بكلفة تقديرية أقل من \$1K	
	2	لا تخفيض في الأداء التقني	برمجيات قابلة للدعم بسهولة	اقتصاد محتمل في الموازنة	موعد تسليم مبكر قابل للتحقيق

توقُّع المخاطرة (Risk Projection)

يقوم توقُّع المخاطرة (ويسمى أيضاً تقدير المخاطرة Risk Estimation) بمحاولة تقدير المخاطرة بطريقتين:

- الأرجحية (Likelihood)
 - احتمال أن تكون المخاطرة حقيقية.
 - العواقب (Consequences)
 - النتائج أو التبعات الناتجة عن المشاكل المتعلقة بالمخاطرة عند وقوعها.
- و ينفَّذ مدير المشروع بالتعاون مع المديرين الآخرين والكادر التقني أربع فعاليات لتوقُّع المخاطرة:
- تأسيس مقياس يعطي الأرجحية المتوقعة للمخاطرة
 - وصف عواقب المخاطرة
 - تخمين أثر المخاطرة في المشروع والمنتج
 - ملاحظة الدقة الإجمالية لتوقُّع المخاطرة حتى لا يكون هنالك مستقبلاً أي سوء فهم

إنشاء جدول المخاطرة

يزوِّد جدول المخاطرة (Risk Table) مدير المشروع بتقنية بسيطة لتوقُّع المخاطرة. لاحظ الجدول التالي والذي يمثِّل جزءاً من جدول المخاطرة لمشروع ما:

المخاطر	الفئة	الاحتمال	الأثر	RMMM
تقييم الحجم قد يكون منخفض جداً	PS	60%	2	
عدد المستخدمين اكبر من المخطط له	PS	30%	3	
إعادة استخدام أقل من المخطط لها	PS	70%	2	
المستخدم النهائي لا يتقبَّل النظام ويقاومه	BU	40%	3	
سيجري التشدد في الموعد النهائي للمشروع	BU	50%	2	
سيحدث فقدان التمويل	CU	40%	1	
سيغيّر الزبون المتطلبات	PS	80%	2	
لن تحقق التكنولوجيا التوقعات	TE	30%	1	
نقص في التدريب على الأدوات	DE	80%	3	
الكادر غير خبير	ST	30%	2	
التغيير في الكادر سيكون كثيراً	ST	60%	2	
...	

- يبدأ فريق المشروع بسرد جميع المخاطر بمساعدة قائمة تفقّد عناصر المخاطرة
 - تصنف كل مخاطرة في العمود الثاني من الجدول (مثلاً: PS تعني مخاطرة حجم المشروع، BU يعني مخاطرة الأعمال)
 - يمكن تقدير قيمة احتمال حدوث كل مخاطرة من قبل كل عضو من أعضاء الفريق، ومن ثم أخذ وسطي القيم الافردية للوصول إلى إجماع (موافقة جماعية) على قيمة واحدة للاحتمال
 - يقيّم بعد ذلك أثر كل مخاطرة، قيم الأثر لها المعاني التالية (1- كارثي، 2-حرج، 3-هامشي، 4- تافه)
 - تحديد فئة الأثر (Impact Category) ثم يجري توسيط فئات كل من مكونات المخاطرة الأربعة (الأداء والدعم والكلفة والجدول الزمني) لتحديد قيمة الأثر الكلي
- عند اكتمال ملء الأعمدة الأربعة الأولى من جدول المخاطرة يفرز الجدول اعتماداً على الاحتمال والأثر. تنتقل المخاطر ذات الاحتمال الأعلى والأثر الأعلى إلى قمة الجدول، وتسقط المخاطر ذات الاحتمال الأقل إلى أسفله. يحقق هذا ترتيباً من الدرجة الأولى لأولوية المخاطر.
- إن لأثر المخاطرة واحتمالها وقماً متميزاً على اهتمام الإدارة. ولكن يجب أن لا يأخذ عامل مخاطرة (Risk Factor) أثره كبير واحتمال حدوثه صغير جداً الكثير من وقت الإدارة. في حين أنه يجب الانتباه إلى المخاطر الشديدة الأثر التي احتمال حدوثها كبير أو متوسط، وكذلك المخاطر التي لها أثر ضئيل واحتمال حدوث كبير، وذلك في الخطوات التالية لإدارة المشروع.
- يحتوي العمود المشار إليه بـ "RMMM" (Risk Mitigation, Monitoring and Management) مؤشراً إلى تخفيف المخاطرة وخطة المراقبة والإدارة التي طوّرت بهدف التعامل مع المخاطر.
- يمكن تحديد احتمال المخاطرة بإجراء تقديرات إفرادية ثم تطوير قيمة واحدة متفق عليها. ومع أن هذه الطريقة قابلة للتطبيق، فقد طوّرت طرائق أكثر تطوراً لتحديد احتمال المخاطرة. يمكن على سبيل المثال تقييم سواقات المخاطرة (Risk Drivers) وفق مقياس كيفي للاحتمال له القيم التالية: مستحيل، غير محتمل، متكرّر. ويمكن إرفاق قيمة رياضية للاحتمال مع كل قيمة كيفية (مثال: احتمال بقيمة 0.7 إلى 1 تعني مخاطرة محتملة جداً).

تقييم أثر المخاطرة

تؤثر ثلاثة عوامل في النتائج المحتملة لحدوث مخاطرة وهي: طبيعة المخاطرة، ونطاقها، وتوقيتها.

• طبيعة المخاطرة

تشير طبيعة المخاطرة إلى المشاكل المحتملة الحدوث عند وقوعها. على سبيل المثال، تُعيق واجهة خارجية لأجهزة الزبون عُرُفت تعريفاً سيئاً (مخاطرة تقنية) التصميم المبكر والاختبار، ويُحتمل أن تُؤدي لاحقاً إلى مشاكل في تكامل النظام.

• نطاق المخاطرة

يجمع نطاق المخاطرة شدة المخاطرة (كم هي جديّة؟) إلى توزعها الكلي (كم جانباً من المشروع سيتأثر؟ أو كم زبوناً سيتأذى؟).

• توقيت المخاطرة

يتعلق توقيت المخاطرة بالسؤال: متى ستحدث؟ وكم من الوقت سيستمر الشعور بأثرها؟.

• تحديد النتائج الكلية للمخاطرة

قد يريد مدير المشروع في معظم الحالات أن تقع "الأشياء السيئة" أبكر ما يمكن، ولكن في بعض الحالات كلما تأخر ذلك كان أفضل. ينصح أحد مناهج تحليل المخاطرة بالخطوات التالية لتحديد النتائج الكلية للمخاطرة:

- حدّد الاحتمال الوسطي لقيمة حدوث كل مكون من مكونات المخاطرة
- حدّد أثر كل مكون بالاعتماد على المعايير المبيّنة
- أكمل جدول المخاطرة وحلّل النتائج

تطبّق تقنيات توقع المخاطرة وتحليلها تكرارياً مع تقدم المشروع البرمجي. يجب على فريق المشروع أن يعيد النظر في جدول المخاطرة بشكل منتظم، وإعادة تقويم كل مخاطرة ليحدّد متى قد تؤدي ظروف جديدة إلى تغيير احتمال حدوثها وأثرها. وقد يكون ضرورياً نتيجةً لهذه الفعالية إضافة مخاطر جديدة إلى الجدول أو إزالة بعض المخاطر التي لم تعد ذات علاقة، وكذلك تغيير التوضع النسبي للبعض الآخر الباقي.

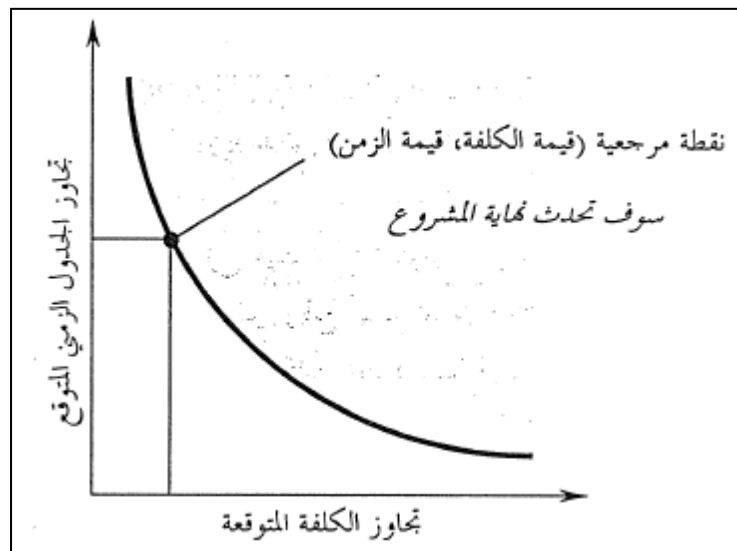
نفحص، خلال تقييم المخاطرة، دقة التقديرات التي أجريت أثناء توقُّع المخاطرة، ونحاول ترتيب أولويات المخاطر التي كُشِف عنها والبدء بالتفكير بطرائق لضبط و/أو تجنب المخاطر التي يُحتمل حدوثها. يتوقَّر لدينا قبل البدء بتقييم المخاطرة مجموعة من الثلاثيات من الشكل $[ri, li, xi]$ ، حيث (ri) تمثل المخاطرة، (li) هو أرجحية (احتمال) المخاطرة، و (xi) هو أثر المخاطرة.

- **المستوى المرجعي للمخاطرة (Risk Referent Level)**

يجب تعريف المستوى المرجعي للمخاطرة حتى يكون التقييم مفيداً. في حالة معظم المشاريع البرمجية، تمثل أيضاً مكونات المخاطرة (الأداء والكلفة والدعم والجدول الزمني) مستويات مرجعية للمخاطرة. أي أن هناك مستوى ل: انحطاط الأداء، أو تجاوز الكلفة، أو صعوبة في توفر الدعم، أو انزلاق الجدول الزمني، أو مزيج من الأربعة، سوف يؤدي إلى إيقاف المشروع. يتوقف العمل إذا سبب مزيج من المخاطر مشاكل تؤدي إلى تجاوز واحد أو أكثر من المستويات المرجعية.

- **النقطة المرجعية (Reference Point)**

في سياق تحليل المخاطر البرمجية، يكون لمستوى المخاطرة نقطة مفردة ، تسمى النقطة المرجعية (Referent Point) أو نقطة الانكسار (Break Point)، يكون فيها القرار باستمرار المشروع أو إنهائه (حين تكون المشاكل كبيرة) مقبولاً بنفس القدر. إذا أدى مزيج من المخاطر إلى مشكلة تتسبب بتجاوز الكلفة والجدول الزمني فسيكون هناك مستوى ما، سيتسبب (عند تجاوزه) بإنهاء المشروع. يبيِّن الشكل التالي المستوى المرجعي للمخاطرة:



يكون لقرارات الاستمرار أو الإنهاء عند النقطة المرجعية الوزن ذاته. نادراً ما يمكن تمثيل المستوى المرجعي على المخطط البياني بخط أملس. بل يكون في معظم الحالات منطقة فيها مساحات من الشك (أي غالباً ما تكون محاولة التنبؤ بقرار الإدارة اعتماداً على مزيج من قيم مرجعية هي محاولة مستحيلة). لهذا نقوم خلال تقييم المخاطرة بالخطوات التالية:

- تُعرّف مستويات مرجعية للمخاطرة في المشروع
- نحاول تطوير علاقة بين كل ثلاثية [ri, li, Xi] وكل من المستويات المرجعية
- نتنبأ بمجموعة من النقاط المرجعية التي تعرّف منطقة الإنهاء، محدودة بمنحنٍ أو مساحات من الشك
- نحاول التنبؤ بكيفية تأثير خلطات من المخاطر في مستوى مرجعي ما

6. وضع الإجراءات المضادة ومراقبة المخاطر:

- استراتيجية التعامل مع المخاطرة
 - تجنّب المخاطرة
 - مراقبة المخاطرة
 - إدارة المخاطرة والتخطيط للطوارئ
- تخفيف المخاطرة (Risk Mitigation)
 - إذا اعتمد الفريق البرمجي منهجاً فاعلاً لإدارة المخاطرة، فإن التجنّب هو دوماً أفضل إستراتيجية، ويتحقق ذلك بتطوير خطة لتخفيف المخاطرة (Risk Mitigation):
 - الاجتماع بالعاملين حالياً لتحديد أسباب التغيير (مثل ظروف عمل سيئة، أجور منخفضة، سوق عمل منافسة)
 - التصرّف لتخفيف الأسباب التي تقع ضمن سيطرة الإدارة قبل بدء المشروع
 - الافتراض، حالما يبدأ المشروع، بأن التغيير سيحدث، وتطوير تقنيات لضمان الاستمرار عندما يغادر بعض العاملين
 - تنظيم فرق المشروع بحيث يجري توزيع المعلومات حول كل فعالية تطوير توزيعاً مناسباً
 - تعريف مقاييس للتوثيق، ووضع آليات للتحقق من أن الوثائق تطوّر في الوقت المناسب
 - إجراء مراجعات لكامل العمل بحيث يكون هناك أكثر من شخص معتمد لهذا العمل
 - تعيين عنصراً احتياطياً لكل تقني ذو دور هام

● مراقبة المخاطرة

تبدأ فعاليات مراقبة المخاطرة مع تقدّم المشروع. فيراقب مدير المشروع العوامل التي قد تقدّم دلالة فيما إذا كانت المخاطرة ستصبح أكثر أو أقل احتمالاً. يمكن مراقبة العوامل التالية في حالة التغيير الكبير للعاملين:

- السلوك العام لأعضاء الفريق اعتماداً على ضغوط المشروع؛
- العلاقات الداخلية الشخصية بين أعضاء الفريق؛
- المشاكل المحتملة في التعويضات والفوائد؛
- توافر العمل ضمن الشركة وخارجها.

يجب على المدير إضافةً إلى مراقبة العوامل المذكورة آنفاً، أن يراقب فعالية خطوات تخفيف المخاطرة

● إدارة المخاطرة والتخطيط للمخاطر

تفترض إدارة المخاطرة والتخطيط للطوارئ أن جهود التخفيف قد أخفقت، وأن المخاطرة أصبحت حقيقية.

قد يعيد مدير المشروع مؤقتاً- تركيز الموارد (وإعادة ضبط الجدول الزمني للمشروع) باتجاه تلك الوظائف التي اكتمل فيها عدد العاملين، ممكناً بذلك القادمين الجدد الذين يجب إضافتهم إلى الفريق من أن يصلوا إلى السرعة المطلوبة في العمل. يطلب من الأفراد الذين سيغادرون أن يتوقفوا عن جميع الأعمال وأن يقضوا أسابيعهم الأخيرة في طور نقل المعرفة (Knowledge Transfer Mode).

7. تدريبات:

1. من أهم الصعوبات التي تميز مشاريع تطوير البرمجيات عن غيرها من المشاريع، صعوبة التحقق وقياس جودة المنتج ؟ (اختر 1 من الإجابات)

- صح
- خطأ

2. المخاطر القابلة للتنبؤ تستخلص من الخبرات السابقة، وخاصة لدى مدير المشروع ؟ (اختر 1 من الإجابات)

- صح
- خطأ

3. من المخاطر التي يمكن إدراجها تحت تصنيف أثر الأعمال: (اختر الإجابة الخاطئة)

- أثر هذا المنتج في عائدات الشركة
- هل جميع الأدوات البرمجية المستخدمة متكاملة مع بعضها؟
- عدد الزبائن الذين سيستخدمون هذا المنتج وانسجام احتياجاتهم معه
- القيود الحكومية على بناء المنتج (إن وجدت)

4. عند تحديد عوامل الخطورة في حجم الكادر وخبرته، وفي حال كانت المخاطرة موجودة، فإن الحلول الممكنة والتي تدرج عادة في الخطة:

- وجود مرشحين محتملين كبداية بنفس الخبرة ويمكن توظيفهم
- التعاقد مع خبرات خارجية متاح وسلس
- امكانية التعاقد الضمني (Subcontracting) متاح وممكن
- جميع الخيارات السابقة صحيحة

5. تقدير أرجحية المخاطرة يتم بـ:

- بناءً على رأي مدير المشروع
- وسطي تقديرات فريق العمل
- من التقييمات السابقة لمشاريع مشابهة
- بالتشاور مع الزبون

الإجابة الصحيحة:

1. (صح)
(قد يتطلب التحقق من بعض الميزات وقتاً طويلاً، وقد لا تظهر بعض الثغرات إلا بعد فترة تشغيل طويلة للمنتج البرمجي)
2. (صح)
(قدرة مدير المشروع على التنبؤ بالمخاطر هي مؤشر على خبرته، ومنها تقييم الزبون و القدرة على التواصل معه، وتوضيح متطلباته بشكل غير قابل للتأويل)
3. (هل جميع الأدوات البرمجية المستخدمة متكاملة مع بعضها؟)
(يندرج هذا البند تحت عنوان بيئة التطوير)
4. (جميع الخيارات السابقة صحيحة)
(يجب على مدير المشروع التفكير في هذه البدائل والأخذ بعين الاعتبار امكانية الاستعانة بها عند الضرورة)
5. (وسطي تقديرات فريق العمل)
(من المفيد تضمين رأي أعضاء فريق العمل في المخاطر المحتملة، فهذا يرفع من وعيهم للمخاطر وأثرها ويحفزهم على بذل الجهد لتلافيها)



الفصل التاسع: التوثيق - ادارة الجودة – الدروس المستفادة

كلمات المفتاحية:

إدارة الجودة، تعقب الرماز المصدري، اختبار المجتزئات، اختبارات التكامل، الاختبار الكلي للنظام.
Software quality assurance, Defect tracing, Unit testing, Integration test,
System Testing, Source code tracing

ملخص:

يناقش هذا الفصل أهم القضايا المتعلقة بإدارة جودة المشروع، التوثيق والدروس المستفادة.

أهداف تعليمية:

يهدف هذا الفصل إلى التعريف بـ:

- شرح مفهوم التحقق من الجودة وإدارة الجودة
- التوثيق في مشاريع تطوير البرمجيات
- الدروس المستفادة وأهمية توثيقها في مشاريع تطوير البرمجيات

1. التوثيق:

التوثيق هو من المهام التي يكرهها ويتجنبها معظم المطورين، علماً أن المستفيد الأساسي منه هو المطورون أنفسهم، وقد أثبتت التجربة أن مشاريع تطوير البرمجيات تسير بشكل أسرع وأفضل مع التوثيق الجيد، وأن التعديلات والتحسينات على المنتج البرمجي أسهل بكثير في حال توافر التوثيق الجيد له.

وفيما يلي سنعرض أهم الوثائق المستخدمة في مشاريع تطوير البرمجيات، علماً أن جزءاً كبيراً منها يستخدم في حالات وفي حالات أخرى لا أهمية له، وذلك حسب منهجية الإدارة المستخدمة في المشروع.

يمكن تصنيف الوثائق إلي نوعين:

- وثائق إجراءات العمل (Process documents)
- وثائق المنتج (Product documents)

وثائق إجراءات العمل (Process documents)

هذه الوثائق هي التي تضبط ايقاع العمل، وهي غالباً ما تعبر عن مرحلة مجزة من العمل، فمثلاً يبدأ المشروع ببيان للهدف (Vision statement)، وينتهي بتقرير نهاية المشروع (Project closure report)، ومرحلة التحليل تنتهي بوثيقة توصيف المتطلبات (SRS). من هذه الوثائق:

- بيان الهدف (Vision statement):
وفيه نجد تعريفاً للمشروع وأهدافه، وتأطيراً له، وهو يفيد في ترتيب الأولويات وقرار ما يمكن تضمينه أو استبعاده في خطة تنفيذ المشروع.
- فهرس المشروع (Project charter):
وفيه نجد المتطلبات وميزات المشروع كخطوط عريضة، وتعاد مع مستوى أعمق من التفصيل في خطة تنفيذ المشروع، وقراءته تعطي فكرة عامة عما سيتم تنفيذه وهو موجه للمستويات الأعلى في الإدارة.

- **وثيقة توصيف المتطلبات (Software requirements and specification):**
يشار لها اختصاراً ب SRS، وهي قائمة بالمتطلبات الوظيفية وغير الوظيفية المراد أن يحققها النظام، بالإضافة إلى شروط العمل والقيود المفروضة من قبل الزبون.
هذه الوثيقة هي الأهم من بين وثائق المشروع، وهي المرجع للاختار والتسليم، وهي تمثل المرحلة الأولى في التخاطب ما بين الزبون والمطورين، وبتبدأ عادة بنسخة مختصرة تتم المراجعة عليها وتفصيلها تباعاً بالتعاون مع الزبون.
تتضمن هذه الوثيقة عادة قائمة بميزات النظام، وعرض لطرق وسناريوهات العمل في النظام.
يمكن التعبير عن محتوى هذه الوثيقة من خلال حالات استخدام (Use cases).
في آجاليل يستعاض عن هذه الوثيقة بقائمة ميزات المنتج (Product backlog)، وقد شرحنا عن هذه الوثيقة في الفصول السابقة (ورقيات آجاليل).

- **خطة ادارة المشروع (Project management plan):**
يشار لها اختصاراً ب PMP، وهي عرض للمخرجات المتوقعة من المشروع، ويتم تجميعها على شكل كتل تنتهي كل منها بنقطة علام، مع وصف للاجراءات الكفيلة بتحقيق هذه المخرجات.
خطة ادارة المشروع يتم التعديل عليها باستمرار، وقليل من المشاريع تبدء بمتطلبات واضحة، مفصلة وكاملة.

- **خطة الاصدار (Release plan):**
تتضمن وصفاً لمتجزئات النظام، التي يلتزم فريق العمل بانجازها ضمن الاطار الزمني المحدد.

- **خطة المرحلة (Iteration plan or Sprint plan):**

تتضمن ميزات النظام التي يلتزم فريق العمل بانجازها ضمن الاطار الزمني للمرحلة.

من الوثائق المستخدمة في ادارة مشاريع تطوير البرمجيات أيضاً:

■ شروط نجاح المشروع (Project success criteria)

■ تقرير نهاية المشروع (Project closure report)

وثائق المنتج (Product documents)

- هيكلية وتصميم المنتج (Architecture & Design):
وفيه نجد التصميم الكامل للنظام، ومن أهم الأدوات لصياغته هي مخططات UML. عادة يبدأ بتعريف هدف النظام ومنه نستنتج اللاعبين الأساسيين (Main actors)، ومن ثم:
 - حالات الاستخدام (Use cases)
 - حالات الاستخدام المفصلة (Detailed use cases)
 - مخططات الصفوف (Class diagrams)
 - مخططات التتابع الزمني (Sequence diagrams)
 - وأخيراً مخططات التجوال (Windows navigation diagrams)

- معايير كتابة الرمز المصدري (Coding standards):
هي قائمة بمعايير تقييم جودة الماز المصدري، تتيح الاستعانة بها من قبل المطورين لتذكر معايير التقييم عند البرمجة، وعند مراجعة الرمز المصدري لمطور آخر (Peer review).
من الأمثلة على محتوى هذه الوثيقة:
 - طرق التسمية (Naming conventions)
 - طريقة اضافة التعليقات التوضيحية (Comments Style)
 - منهجية معالجة الأخطاء في الاجرائيات (Error trapment)وعادة يتم صياغة هذه الوثيقة على شكل قائمة أسئلة (Checklist)، على الشكل التالي:
 - هل تم التقيد بطريقة التسمية المعتمدة ؟
 - هل تم اضافة تعليقات توضيحية كافية ؟
 - هل تم التقيد باضافة جزء ملاحقة أخطاء في كل إجرائية ؟
 -

- خطة الاختبار (Test Plan):
هذه الوثيقة مهمة جداً، وهي مرجع لاستلام المنتج ووضعه قيد الاستخدام الفعلي، وتتضمن جميع الاختبارات التي يجب أن يحققها النظام لإعلان انتهائه، بالإضافة إلى اختبارات الأداء والجودة.
خطة الاختبار عادة يتم التعديل عليها مع تطور المشروع، والسبب هو التغييرات في المتطلبات والتعديلات والتحسينات التي تتطلب بدورها اختباراً.

من المفيد أن تضاف إلى سيناريوهات الاختبار في هذه الوثيقة:

- اسم المختبر
- تاريخ الاختبار
- التعليق في حال فشل الاختبار
- تاريخ التصحيح لكل خطأ
- طريقة تصحيح لكل خطأ

ويمكن أن تعاد الاختبارات الموثقة في الخطة أكثر من مرة، ومن قبل أكثر من شخص. خطة الاختبار هي المرجع للمطورين عندما يقومون بمراجعة النظر (Peer review).

• دليل المستخدم (User manual):

نجد فيه عادة:

- شرح مختصر عن النظام
- متطلبات التشغيل البرمجية والمادية
- تنصيب و تشغيل النظام
- شرح لعناصر التحكم (قوائم الخيار، أزرار التنقل، ...)

ومن ثم شرح لطريقة عمل البرنامج ووظائفه، إما من خلال سيناريوهات العمل (كيف نضيف بيانات زبون جديد، كيف نقوم بعملية اخراج من المستودع، ...)، أو من خلال شرح الوظائف في كل واجهة تخاطبية من النظام.

يضاف إلى ما سبق تسهيلات لاستخدام الدليل، ك فهرس مفصل للمحتويات، وكلمات مفتاحية مثلاً.

2. جودة المنتج:

هي بالتعريف إلى أي درجة يحقق المنتج المتطلبات الصريحة والضمنية المطلوبة.

مثال عن المتطلبات الضمنية:

- عدم وجود أخطاء غير معالجة
- توافق المنتج وطريقة استخدامه مع توقعات الزبون وظروف عمله
- الأداء وخاصة في الحالات الحدية

الجودة عادة تبطاء من سرعة تقدم المشروع، وتزيد من كلف التطوير والصيانة، وقد ترفع كلفة المشروع إلى حدود غير مقبولة، وهذا يعني أن الجودة يجب ضبطها وتعريف مستويات لها، والوقوف عند المستوى المناسب.

والسوية المطلوبة من الجودة توضع في خطة مع بدء المشروع، تسمى خطة الجودة، ويلتزم بها كامل فريق العمل.

خطة الجودة:

- يجب أن تكون موثقة
- يمكن اسناد معام التحقق من الجودة إلى فريق عمل مستقل عن فريق التطوير
- من الأمور التي يتم ذكرها في خطة الجودة:

توثيق الخطأ، مصدره، متى تم اكتشافه، متى تم حله، كيف تم حله.	Defect tracing	ملاحقة الأخطاء
كل مجتزء يجب اختباره على حدى، فور انجازه.	Unit testing	اختبار المجتزئات
المرور على الرماز المصدري سطرًا سطرًا.	Source code tracing	تتبع الرماز المصدري
كل عمل يقوم به شخص يتم اختباره من قبل شخص آخر.	Technical review	المراجعة التقنية
اختبار واجهات الربط بين الأجزاء المنفذة.	Integration test	اختبارات التكامل
تشغيل النظام بغاية اكتشاف الأخطاء فيه، واختبار جميع سيناريوهات العمل.	System test	الاختبار الكلي للنظام

3. الدروس المستفادة:

- الدروس المستفادة هي جزء من توثيق المنتج
- تتضمن الدروس التي تعلمها كل من أعضاء فريق العمل، واقتراحاته لتحسين طرق العمل
- الفائدة منها في المشاريع التالية والمستقبلية
- لا تتضمن وثيق الدروس المستفادة سلبيات العمل فقط، بل يمكن أن نذكر فيها التجارب التي أعطت نتائج جيدة
- يتم بناء هذه الوثيقة والتعديل عليها في جميع مراحل تطوير المنتج

Lesson Learned No.	Date	Name	Role	Lesson Description	Recommended Items to add to the process	Recommended Items to remove from the process	PM Process	System Development Phase	Knowledge Area
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									

- هذه الوثيقة تفيد في بناء دليل عمل لشركة تطوير البرمجيات، وهي تفيد في نقل الخبرات بين أعضاء فريق العمل

4. تدريبات:

1. من العناصر التي ترد في خطة ضبط الجودة ؟ (اختر 1 من الإجابات)

- التحقق من الرماز المصدري.
- اختبارات التكامل
- اختبار المتجزئات
- الاختبار الكلي للنظام
- كل ما سبق

2. الوثيقة التي تتضمن مخططات حالات الاستخدام (Use case diagrams) ؟ (اختر 1 من

الإجابات)

- خطة الاحتمار
- هيكلية وتصميم المنتج
- دليل الاستخدام
- دليل ميزات المنتج

3. يفضل أن يقوم بمهمة التحقق من الجودة، فريق عمل مستقل عن المطورين ؟ (اختر 1 من

الإجابات)

- صح
- خطأ

4. الدروس المستفادة توضع عادة في تقرير نهاية المشروع ؟ (اختر 1 من الإجابات)

- صح
- خطأ

الإجابة الصحيحة:

1. (كل ما سبق)
(راجع الفقرة 1)
2. (هيكلية وتصميم المنتج)
(راجع الفقرة 1-b)
3. (صح)
(راجع الفقرة 2)
4. (خطأ)
(الدروس المستفادة، ولما لها من أهمية توضع في وثيقة مستقلة، ويمكن الرجوع لها في مشاريع أخرى)

5. المراجع:

- IEEE Standard for Software Test Documentation