



مقدمة إلى نماذج نضج البرمجيات
Software Maturity Models

المحاضرة الأولى

مفهوم نضج البرمجيات

أولاً: تعريف نموذج نضج البرمجيات :

نموذج النضج هو أداة تستخدمها الشركات وفرق تطوير البرمجيات لقياس مدى نجاح أعمالهم أو مشاريعهم ومدى قدرتهم على التحسين المستمر.

على عكس أدوات القياس الأخرى الموجهة نحو الأهداف، يمكن لنماذج النضج تقييم البيانات النوعية لتحديد المسار والأداء الطويل الأجل للشركة. تهدف النماذج إلى معرفة ما إذا كانت الشركات تنضج، مما يعني أنها تختبر وتنمو وتحسن باستمرار.

مفهوم نضج البرمجيات

ثانياً: أهمية نماذج نضج البرمجيات :

- ▶ توفر مراقبة أداء مرنّة يمكن أن تكشف عن معلومات قيمة حول صحة الشركة وإمكانياتها.
- ▶ على الرغم من أن **النماذج لا تعالج الكفاءات** بشكل مباشر، إلا أنها يمكن أن تحدد المجالات التي لا تتقيد فيها المنظمات بالمعايير وتحتاج لها تحديد الاستراتيجيات التي يمكن أن تحسن عملياتها وإجراءاتها من خلالها.
- ▶ تركز على التعلم والتحسين: من خلال تحديد المستوى الحالي لفريق أو شخص أو عملية، يمكن للنموذج أن يكشف عما يحتاج إلى التغيير لنقل الشركات إلى المستوى التالي.

مفهوم نضج البرمجيات

ثالثاً: أنواع نماذج النضج:

هناك عدة أنواع من نماذج النضج. تم تصنيفها حسب مجال استخدامها.
فيما يلي ثلاثة أنواع مختلفة من النماذج :

1. **نموذج نضج العمليات التجارية Business process maturity model** يستخدم نموذج نضج العمليات التجارية خمسة مستويات لتقدير نضج المنظمة. المستويات هي:

- **المستوى الأول Initial:** المستوى الأول هو أدنى مستوى. يصف ممارسات الإدارة غير المتسلقة أو الفرق التي تتفاعل مع الأزمات بدلاً من التنبؤ بها.
- **المستوى الثاني Managed:** يعرف هذا المستوى الفرق والشركات التي لديها أساس للإدارة، لكن الفرق الفردية داخل الشركة لا تزال تعمل بشكل معزول مع حد أدنى من التعاون أو الأدلة على دمج استراتيجيات التحسين.
- **المستوى الثالث Standardized:** المستوى الموحد أو إدارة العمليات يعني أن الشركة على دراية بعملياتها وتعمل نحو تحقيق الاتساق والتسليم المتساوي.
- **المستوى الرابع Predictable:** في هذا المستوى، تستخدم المنظمات بنيتها التحتية للعمليات وقدرات أصولها لتحقيق نتائج موثوقة من خلال التحكم في التباينات داخل مخرجاتها.
- **المستوى الخامس Optimizing:** في هذا المستوى، تعمل الشركات على التحسين المستمر والتركيز على الابتكار.



مفهوم نضج البرمجيات

ثالثاً: أنواع نماذج النضج:

2. نموذج نضج القدرات Capability maturity model

- صُمم نموذج نضج القدرات في البداية لتطوير البرمجيات، وتقدير نضج المنظمة أو أنظمة تطوير البرمجيات بمقارنتها بأفضل الممارسات في الصناعة.
- من خلال قياس النتائج وتعيين مستويات النضج، يمكن للشركات وفرق التطوير استخدام نماذجها لتقدير وعيها بعمليات الأعمال، وتقنيات الإدارة الفعالة، وال المجالات التي تحتاج إلى تحسين.
- تستخدم جميع نماذج النضج المستويات المذكورة في النموذج السابق مع بعض التباينات لوصف العمليات.
- في نماذج القدرات: تتعلق المستويات بشكل أوضح بعمليات التطوير.

مفهوم نضج البرمجيات

ثالثاً: أنواع نماذج النضج:

3. نموذج نضج الأجيال Agile ISO maturity model

على عكس نماذج النضج الأخرى، يعمل نموذج منظمة المعايير الدولية (ISO) على القضاء على بعض الانتقادات الموجهة لنماذج النضج. من خلال:

- توحيد المستويات Standardizing the levels.

- تحدد النماذج الأجيال توقيعات أكثر وضوحاً تحددها هيئة دولية.

بعض نماذج النضج ليست مفصلة بما يكفي لتوفير تقييمات مفيدة. تعمل النماذج الأجيال الموحدة Standardized على مكافحة ذلك من خلال التركيز الأقل على التطوير المتسلسل والمزيد على المراحل الفردية من دورة حياة التطوير. تستخدم العديد من منهجيات الأجيال Sprints، والتي تركز فرق التطوير وتقسم المشاريع إلى مراحل أكثر قابلية للتحقيق. يتيح اتخاذ نهج تكراري في التطوير لفرق تقييم المجالات الأكثر تفصيلاً وتحديد المجالات التي تحتاج إلى تحسين بشكل أكثر فعالية.



مفهوم نضج البرمجيات

رابعاً: قياس نضج البرمجيات:

قياس نضج البرمجيات يمكن أن يتم من خلال استخدام نماذج نضج البرمجيات المختلفة التي تساعد في تقييم مدى كفاءة وفعالية العمليات والأنظمة داخل المؤسسة. يؤخذ بعين الاعتبار الخطوات التالية:

.1 اختيار نموذج نضج مناسب:

- نموذج نضج القدرات (CMMI) يقيس مدى قدرة المؤسسة على تطوير البرمجيات بشكل متكرر وفعال.
- نموذج نضج العمليات التجارية (BPMM) يقيس مدى نضج العمليات التجارية والتكامل بين الفرق.
- نموذج نضج الأ杰يل (Agile Maturity Model) يقيس مدى تبني منهجيات أ杰يل في تطوير البرمجيات.

.2 تحديد مستوى النضج:

- المستوى الأول (Initial) يشير إلى عمليات غير منضبطة وردود فعلية للأزمات.
- المستوى الثاني (Managed) يوجد بعض التنظيم في العمليات ولكنها غير متكاملة بين الفرق.
- المستوى الثالث (Standardized) العمليات متناسقة وموحدة عبر المؤسسة.
- المستوى الرابع (Predictable) العمليات مكررة ويمكن التنبؤ بنتائجها.
- المستوى الخامس (Optimizing) المؤسسة تركز على التحسين المستمر والابتكار.

.3 تقييم العمليات الحالية:

- تحليل العمليات الحالية وتحديد مدى توافقها مع معايير النموذج المختار.
- استخدام استبيانات ومقابلات مع الفرق المختلفة لجمع البيانات اللازمة.

مفهوم نضج البرمجيات

رابعاً: قياس نضج البرمجيات:

.4 تحليل النتائج:

- مقارنة النتائج مع المعايير المحددة في النموذج لتحديد مستوى النضج الحالي.
- تحديد نقاط القوة والضعف في العمليات الحالية.

.5 وضع خطة للتحسين:

- بناء خطة عمل لتحسين العمليات ورفع مستوى النضج.
- تحديد الأهداف والمهام وتوزيع الأدوار والمسؤوليات.

.6 المتابعة والتقييم الدوري:

- مراجعة العمليات بشكل دوري لضمان تنفيذ التحسينات.
- إعادة تقييم مستوى النضج بمرور الوقت للتحقق من التقدم والتحسين المستمر.

باستخدام هذه الخطوات، يمكن للمؤسسات تحسين كفاءة وفعالية عمليات تطوير البرمجيات وتوجيه الجهود نحو تحقيق مستويات أعلى من النضج والابتكار.

مفهوم نضج البرمجيات

خامساً: الفرق بين العمليات الناضجة العمليات غير الناضجة:

الفرق بين البرمجيات العمليات الناضجة العمليات غير الناضجة يكمن في عمليات التنظيم، الكفاءة، القدرة على التكرار، والتحسين المستمر. يمكن تلخيصها في الجدول التالي:

العملية	العمليات الناضجة	العمليات غير الناضجة
التنظيم والهيكلة	تتمتع بعمليات محددة ومعرفة بوضوح، مع وجود وثائق وإجراءات محددة لكل عملية. تتبع المؤسسة معايير وقواعد ثابتة لتنفيذ العمل.	تفتقر إلى التنظيم والهيكلة، وتعتمد على الاستجابة الفردية للمواقف والأزمات بدون وجود إجراءات ثابتة أو توثيق مناسب.
التكرار والاتساق	قابلة للتكرار والاتساق عبر جميع المشاريع والفرق. يمكن التنبؤ بنتائج العمليات لأن الإجراءات تتبع معايير محددة.	تعتمد على ردود فعل غير منتظمة وغير متوقعة. تختلف نتائج العمليات بشكل كبير بسبب عدم وجود معايير ثابتة.
التحسين المستمر	تضمن آليات لتحليل الأداء وتحديد المجالات التي تحتاج إلى تحسين. يتم تنفيذ تحسينات مستمرة بناءً على البيانات والتقييمات الدورية.	تفتقر إلى التركيز على التحسين المستمر. عادةً ما تظل العمليات كما هي دون مراجعة أو تحسينات منهجية.

مفهوم نضج البرمجيات

خامساً: الفرق بين العمليات الناضجة العمليات غير الناضجة:

العملية	العمليات الناضجة	العمليات غير الناضجة
القدرة على التنبؤ والسيطرة	تتمتع بقدرة عالية على التنبؤ بالنتائج والتحكم في العمليات. يمكن إدارة المخاطر بشكل فعال بفضل الإجراءات المحددة والمعايير الثابتة.	تفقر إلى القدرة على التنبؤ والسيطرة، مما يؤدي إلى مخاطر عالية وفشل متكرر في تحقيق الأهداف المطلوبة.
الاعتمادية والجودة	تنتج برامج ذات جودة متدنية وغير موثوقة، بسبب عدم اتباع معايير الجودة وعدم وجود عمليات مراجعة صارمة.	تنتج برامج ذات جودة متدنية وغير موثوقة، بسبب عدم اتباع معايير الجودة وعدم وجود عمليات مراجعة صارمة.
التعاون والتكامل	تشمل تعاوناً فعالاً وتكاملاً بين الفرق المختلفة، مما يعزز التنسيق وتحقيق الأهداف المشتركة.	قد تعاني من عزلة الفرق وغياب التعاون، مما يؤدي إلى تكرار الجهود والموارد وعدم تحقيق الأهداف بكفاءة.

مفهوم نضج البرمجيات

سادساً: لمحّة تاريخية عن نماذج نضج البرمجيات:

تُعد نماذج نضج البرمجيات (Software Maturity Models) أدوات هامة في مجال تطوير البرمجيات، حيث تُستخدم لقياس وتحسين عمليات تطوير البرمجيات داخل المؤسسات. وفيما يلي لمحّة تاريخية عن تطور هذه النماذج:

1. السبعينيات والثمانينيات:

- بداية الوعي بأهمية العمليات: بدأت المؤسسات تدرك أهمية وجود عمليات منظمة لتطوير البرمجيات، حيث كان ذلك نتيجة للتحديات التي واجهتها من حيث الجودة والجدول الزمني والتكلفة.
- مبادرات مختلفة: بدأت بعض المؤسسات مثل IBM بتطوير نماذج داخلية لتحسين عملياتها.

2. أواخر الثمانينيات:

- تطوير نموذج CMM: في عام 1986، طورت مؤسسة كارنيجي ميلون (Carnegie Mellon) بالتعاون مع وزارة الدفاع الأمريكية نموذج نضج القدرات (Capability Maturity Model) - CMM. كان الهدف من هذا النموذج هو تحسين جودة البرمجيات من خلال تحسين عمليات تطوير البرمجيات.

3. التسعينيات:

- إصدار CMM: تم إصدار الإصدار الأول من نموذج نضج القدرات (CMM) في عام 1991، وقد لقي استحساناً كبيراً واعتمده العديد من المؤسسات لتحسين عملياتها.

- CMMI - Capability Maturity Model Integration - CMMI: في أواخر التسعينيات، بدأ تطوير نموذج التكامل لنضج القدرات (CMMI) كمحاولة لتوحيد مختلف نماذج CMM الفرعية (مثل للهندسة ونموذج CMM لإدارة المشاريع).

مفهوم نضج البرمجيات

سادساً: لمحات تاريخية عن نماذج نضج البرمجيات:

4. الألفية الجديدة:

- إصدار CMMI: في عام 2002، تم إصدار نموذج CMMI كنسخة موحدة ومحسنة من نموذج CMM، وأصبح المعيار الأكثر استخداماً لتحسين عمليات تطوير البرمجيات.
- التوسع في النطاق: لم يعد استخدام نماذج نضج البرمجيات مقتصرًا على البرمجيات فقط، بل امتد ليشمل مجالات أخرى مثل إدارة المشاريع، والهندسة، وإدارة الخدمات.

5. التحديثات المستمرة:

- إصدارات جديدة: تم إصدار نسخ محدثة من CMMI بمرور الوقت، مثل CMMI Version 1.3 في عام 2010 و CMMI Version 2.0 في عام 2018. هذه الإصدارات تتضمن تحسينات بناء على المستخدمين وتطورات الصناعة.
- التركيز على التحسين المستمر: أصبحت نماذج نضج البرمجيات تركز بشكل أكبر على التحسين المستمر وتقديم قيمة حقيقية للمؤسسات.

6. الوقت الحاضر:

- الاعتماد الواسع: تُستخدم نماذج نضج البرمجيات اليوم على نطاق واسع في مختلف الصناعات لتحسين الجودة، وتقليل التكاليف، وزيادة الكفاءة.
- التكامل مع الأساليب الحديثة: تتكامل نماذج نضج البرمجيات مع أساليب تطوير البرمجيات الحديثة مثل Agile و DevOps لضمان تحقيق أفضل الممارسات وتحسين الأداء.



نموذج التكامل لنضج القدرات (Capability Maturity Model Integration – CMMI)

تعريفه:

هو إطار عمل طوره معهد الهندسة البرمجية (SEI) التابع لجامعة كارنيجي ميلون والمستخدم على نطاق واسع لتحسين وقياس العمليات داخل المؤسسات. يهدف CMMI إلى تعزيز كفاءة وجودة عمليات تطوير البرمجيات وإدارة المشاريع والخدمات، بالاعتماد على مجموعة من الممارسات المثبتة والمعترف بها دولياً.

أهداف CMMI

1. **تحسين الجودة والكفاءة:** يسعى CMMI إلى تحسين جودة البرمجيات والمنتجات والخدمات من خلال تطبيق ممارسات وعمليات قياسية وفعالة.
2. **تحديد مستوى النضج العملياتي:** يوفر CMMI إطاراً لقياس مدى نضج العمليات داخل المؤسسات، حيث يمكن تصنيف العمليات على مستويات منخفضة إلى عالية بناءً على قدرة المؤسسة على تنفيذها بشكل متكرر ومنتظم.



نموذج التكامل لنضج القدرات (Capability Maturity Model Integration – CMMI)

أهداف CMMI

3. تعزيز التحسين المستمر: يشجع CMMI على تنمية ثقافة التحسين المستمر داخل المؤسسات، من خلال تقديم آليات وممارسات لقياس الأداء وتحديد الفجوات وتطبيق استراتيجيات للتحسين.
4. تعزيز التكامل والتوحيد: يعتمد CMMI على مجموعة موحدة من الممارسات والمعايير التي تساعد على توحيد العمليات داخل المؤسسة وتحقيق التكامل بين الأقسام المختلفة.
5. المساعدة في تحقيق أهداف الأعمال: يهدف CMMI إلى دعم المؤسسات في تحقيق أهدافها التجارية من خلال تحسين إدارة المشاريع، والتحكم في الجودة، وتحسين الكفاءة العملية.



نموذج التكامل لنضج القدرات (Capability Maturity Model Integration – CMMI)

مكونات CMMI:

1. عمليات التقييم والتحسين (Process Improvement)

تشمل هذه المكونة تطوير وتنفيذ استراتيجيات لتحسين العمليات داخل المؤسسة، بما في ذلك تقييم العمليات الحالية وتحديد نقاط الضعف والتحسين المستمر.

2. إدارة العمليات (Process Management)

تركز على تطوير وتوثيق عمليات إدارة العمليات داخل المؤسسة، بما في ذلك تحديد المعايير والإجراءات وتقديم التوجيهات لفرق العاملة.

3. التدريب والتطوير (Training and Development):

تشمل تلك المكونة توفير التدريب والتطوير المستمر لفرق العاملة، وتعزيز المعرفة والمهارات اللازمة لتحقيق أهداف CMMI.

4. دعم القرارات (Decision Support):

توفر أدوات وتقنيات لدعم اتخاذ القرارات الاستراتيجية والتكتيكية داخل المؤسسة، بما في ذلك تحليل البيانات وإعداد التقارير والتوجيهات.



نموذج التكامل لنضج القدرات (Capability Maturity Model Integration – CMMI)

أبعاد CMMI

1. القدرة (Capability):

تعبر عن قدرة المؤسسة على تنفيذ العمليات بشكل متكرر ومتوقع وفقاً لمعايير محددة، وتقاس عادة بمستويات النضج المختلفة (مثل Initial، Managed، Defined، Quantitatively Managed، Optimizing).

2. العمليات (Processes):

تعبر عن العمليات الرئيسية التي يجب أن تكون موجودة ومنظمة داخل المؤسسة، وتشمل توثيقها وتحديدها وتوفير التدريب والدعم اللازم لتنفيذها بشكل فعال.

3. المستوى (Level):

يعبر عن مستوى النضج للعمليات داخل المؤسسة، ويتم تحديده بناءً على مدى تنفيذ العمليات وفقاً لمعايير معينة، حيث يتقدم المستوى كلما تم تحقيق المزيد من التحسين والتكامل.

4. التقييم والتحسين (Assessment and Improvement):

يشمل هذا البعد تقييم أداء العمليات وتحديد المجالات التي تحتاج إلى تحسين، وتطبيق استراتيجيات التحسين المستمر لضمان تحقيق الأهداف المحددة.



نموذج التكامل لنضج القدرات (Capability Maturity Model Integration – CMMI)

مستويات CMMI

نموذج التكامل لنضج القدرات (CMMI) يتكون من خمسة مستويات رئيسية لتقدير وتحسين العمليات داخل المؤسسات. هذه هي المستويات الخمسة:

1. المستوى الأول Initial (الابتدائي):

في هذا المستوى، لا تكون العمليات محددة بشكل كافٍ، وتتسم بالعشوائية وعدم الثبات. الفرق يتفاعل عادةً مع المشاكل والأزمات بدلاً من التنبؤ بها مسبقاً.

2. المستوى الثاني Managed (المدار):

يتم في هذا المستوى تحديد وتوثيق العمليات الأساسية داخل المؤسسة، مع التركيز على إدارة العمليات بشكل فعال. لكن الفرق والأقسام داخل المؤسسة قد يعملون بشكل معزول مع الحد الأدنى من التعاون والتحسين المشترك.

3. المستوى الثالث Defined (المحدد):

في هذا المستوى، تكون العمليات محددة بشكل كامل وموثقة بشكل مناسب، ويتم تنفيذها بشكل منظم ومنظم. هذا المستوى يعكس وجود إدارة العمليات التي تعمل على تطويرها وتحسينها بشكل مستمر.



نموذج التكامل لنضج القدرات (Capability Maturity Model Integration – CMMI)

مستويات CMMI

4. المستوى الرابع Quantitatively Managed (المدار بشكل كمي):

يعتمد هذا المستوى على تحليل البيانات الكمية للعمليات وإدارتها بشكل كفاء، مما يساعد على التحكم في التباينات داخل الإنتاج وتحقيق نتائج موثوقة وقابلة للتنبؤ.

5. المستوى الخامس Optimizing (المحسن):

يعبر هذا المستوى عن تحقيق الفرق والمؤسسات لأقصى استفادة من العمليات، من خلال التركيز على التحسين المستمر والابتكار. يتم تطبيق أفضل الممارسات والتقنيات لتحقيق الأهداف الاستراتيجية بشكل متواصل.

باستخدام هذه المستويات، يمكن للمؤسسات تحديد مدى نضجها وقدرتها على تحقيق الأهداف المحددة، وتحديد الخطوات اللازمة لتحسين وتطوير عملياتها بشكل مستمر.

