

مراجع المحاضرة : كتاب هندسة البرمجيات - تأليف د.أحمد شعبان دسوقي - أ.د. السيد محمود الريبيعي



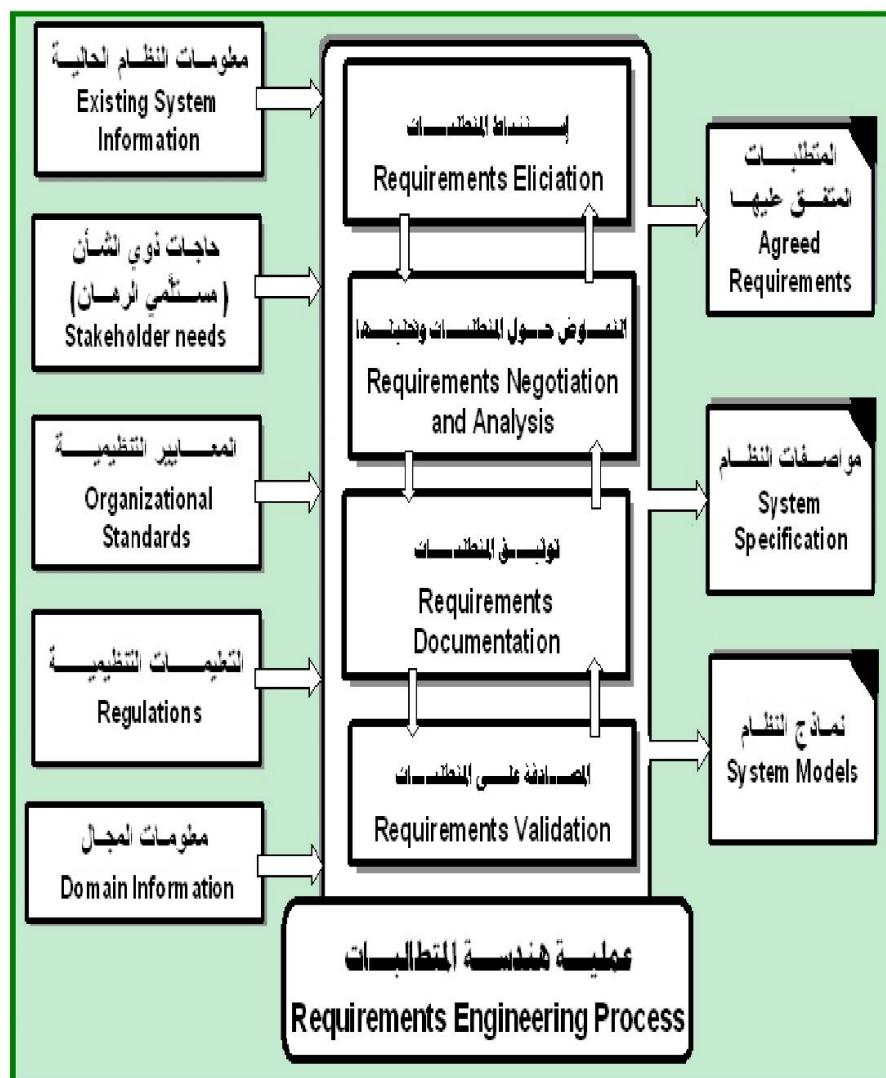
محتويات الوحدة

الصفحة	الموضوع
115	مقدمة
115	تمهيد
117	أهداف الوحدة
119	1. مفهوم هندسة المتطلبات
121	1.1 مراحل (أنشطة) هندسة متطلبات البرمجيات
129	2.1 مدخلات ومخرجات عملية هندسة متطلبات البرمجيات
131	2. مفهوم المتطلبات
137	3. أنواع المتطلبات
137	1.3 المتطلبات الوظيفية
138	2.3 المتطلبات غير الوظيفية
143	4. وثيقة مواصفات المتطلبات
145	1.4 مستويات المتطلبات
149	2.4 بنية مستند مواصفات المتطلبات
151	الخلاصة
153	للمحة مسبقة عن الوحدة التالية
154	إجابات التدريبات
156	مسرد المصطلحات
161	المراجع

والسمة الأخرى لتوصيف المتطلبات هي أنه من الضروري وضع الدليل الإرشادي حول كيفية التأكد من أن البرنامج أو النظام يلبي احتياجات المستخدمين ، وذلك من خلال استخدام أدوات القياس والاختبارات المختلفة.

١. مفهوم هندسة المتطلبات

تُعرف هندسة المتطلبات بأنها عملية (Process) تتضمن مجموعة من الأنشطة الهيكلية (Structured Set of Activities) التي يتم اتباعها لاستنتاج وثيقة متطلبات النظام والمصادقة عليها ، وتوثيقها ، والمحافظة عليها ، حيث تشمل هذه المتطلبات الخدمات (Services) التي يحتاجها العميل من النظام ، وكذلك تحديد القيود (Constraints) التي يجب أخذها في الاعتبار وعدم تجاوزها أثناء العمل على تطوير النظام. وكلمة "هندسة" (Engineering) هنا تدل ضمناً على أن هذه الأنشطة يجب أن تتم بصورة مُرتبة (Systematic) وقابلة للتكرار (Repeatable) لضمان أن تكون المتطلبات الناتجة كاملة (Complete) ومتاغمة (Consistent) ، ووثيقة الصلة بمتطلبات المستخدم (Relevant) ، وخلافه من مميزات المتطلبات الجيدة، والشكل رقم 4.1 يوضح أهم الأنشطة الهيكلية (المراحل) التي تتضمنها عملية هندسة المتطلبات ، وكذلك مدخلاتها ومخرجاتها.



شكل 4.1 : شكل توضيحي لعملية هندسة البرمجيات : أهم الأنشطة الهيكلية (المراحل) والمدخلات والمخرجات.

وسوف نركز هنا فقط على بعض الأنشطة المهمة التي تشملها عملية هندسة المتطلبات ، وهي كالتالي :

- استباط المتطلبات . (Requirements Elicitation)
- التفاوض حول المتطلبات وتحليلها (Requirements Negotiation and Analysis).
- توثيق المتطلبات . (Requirements Documentation)
- المصادقة على المتطلبات (Requirements Validation)

1.1 مراحل (أنشطة) هندسة متطلبات البرمجيات

① المرحلة الأولى : استباط المتطلبات : وهي مسؤولية كل من المحللين والمستخدمين بالحوار مع بعضهم البعض بحيث يحاول كل منهم أن يتفهم الآخر، وتؤدي هذه المرحلة إلى بيان غير رسمي للمتطلبات (Informal Statement of Requirements). وهذا يتطلب شكلاً من أشكال الاتصال المباشر والمهارات التي يجب أن يمارسها المحلل ليست تلك المهارات التقنية العادلة التي تتعلق بتطوير البرمجيات ، ولكن تشمل العديد من جوانب مهارات الاتصال الأخرى ، حيث تشمل هذه المهارات الاستماع إلى احتياجات المستخدمين وطرح عدة أسئلة تساعدهم على تحديد أهدافهم وعواقبهم ، وفي النهاية تسجيل وجهات نظرهم لمتطلبات النظام. ومن أهم أسئلة استباط المتطلبات ما يلي :

- ما هي الأسباب وراء تطوير النظام؟ .
- ما هي توقعات المستخدمين من النظام؟ .
- من هم المستخدمون وما هو مستوى خبراتهم العملية؟ .
- ما هي مزئيات المستخدمين حول كيفية استخدامهم للنظام؟ .
- ما هي خصائص البيئة التي يجب أن يستجيب لها النظام؟ .
- ما هي واجهات الاستخدام المتاحة والتي سيتم تصميماها؟ .

- ما هي الوظائف التي سيؤديها النظام من وجهة نظر المستخدمين وبلغتهم العامة؟ .
- ما هي القيود و الأجهزة ، والبرمجيات ، و التواهي الاقتصادية والإجرائية التي يجب أن يستجيب إليها النظام؟ .
- مَاذَا سَيَكُونُ الشكل النهائى للنظام ، شكل النماذج الأولية (Prototypes) ، وشكل النظام النهائي الذي سوف يتم على أساسه الإنتاج الشامل ذو الكميات الكبيرة (Mass Production) .

ومن أهم الإرشادات التي يجب مراعاتها في هذه المرحلة ، ما يلي :

- الأخذ في الاعتبار السياسات والتنظيمات الداخلية للمؤسسة في عملية تطوير البرمجيات ، وكذلك الاهتمامات المهنية لها .
- استشارة كل من له علاقة بعملية تطوير البرمجيات (System Stakeholders) .
- تسجيل منابع المتطلبات (Requirements Sources) .
- تعريف بيئه تشغيل النظام (System's Operating Environment) .
- دراسة القيود المفروضة من قبل نطاق النظام (System Domain Constraints) .
- تسجيل الأسباب المنطقية للمتطلبات (Requirements Rationale) .
- تجميع المتطلبات من وجهات نظر متعددة.
- استخدام أساليب / أنشطة (Techniques / Activities) مختلفة لاستبيان المتطلبات ، والتي من أهمها :
- **أسلوب المقابلات (Interviews Technique)** : وهي عملية اتصال مباشر بين المستخدمين ومن لهم صله مباشرة بالنظام (Stakeholders) ومصممي أو مهندسي البرمجيات، ويطلق عادة على المهندسين الذين يتعاملون مع الاحتياجات بمهندسي المتطلبات أو بمحلي النظم ، وسوف نستخدم في هذه الوحدة مصطلح " محلل " . أما بالنسبة للمستخدمين فإنهم يُعرفون أحياناً بالعملاء أو بالزبائن، وبالطبع هنالك فارق ما بين المستخدم والعميل ، فمثلاً تقوم منشأة أو منظمة (العميل) بشراء نظام معلومات

لموظفيها (المستخدمين المعنيين) كي يستخدمونه في أعمالهم ، ومن ثم سوف نستخدم في هذه الوحدة مصطلح "المستخدم" . ويتم خلال هذا الاتصال مناقشة المتطلبات على هيئة سؤال وجواب ، ويستخدم أسلوب المقابلات في استبيان المتطلبات أو التحقق منها.

• **أسلوب استخلاص البيانات (Data Mining Technique)** : وهو عبارة عن نشاط بحثي خلال أوعية البيانات المختلفة (وثائق (Documents) ، و نماذج (Forms) ، و قواعد بيانات (Database) ، و ملاحظات بريد إلكتروني (E-Mail Notes) ، ...و خلافه). بعرض الحصول على معلومات تفصيلية استخراجها وعلى أساسها يتم تحديد متطلبات النظام المطلوب تطويره.

• **أسلوب المحلل كمتدرب (Analyst as Apprentice)** : في هذا الأسلوب يتم مشاركة المحلل مع المستخدمين النهائين للنظام باعتباره متدرباً للقيام بأعمالهم بعرض إعطاء المحلل فكرة مفصلة ودقيقة عن المهام الخاصة التي يقوم بها المستخدم النهائي للنظام ، والذي هو في النهاية سوف يستخدم النظام المطلوب تطويره.

• **أسلوب التحليل الاجتماعي والمشاهدة (Observation and Social Analysis)** : وهو أسلوب مشابه لأسلوب "المحلل كمتدرب" ، ولكن المحلل يجلس هنا مع المستخدم النهائي لمشاهدته أثناء أدائه لعمله الحالي أو أثناء عمله على نسخة قديمة من النظام المراد تطويره، والغرض من هذا الأسلوب هو اكتساب معلومات تفصيلية عن آلية العمل التي يستخدمها المستخدم النهائي مع تحديد أو معرفة أية آليات ذات كفاءة تم تطويرها واستخدامها من قبله والتي تختلف مواصفات المهام الخاصة به.

② **المرحلة الثانية : التفاوض حول المتطلبات و تحليلها** : وهي مرحلة يتم فيها تصنيف المتطلبات وتنظيمها في مجموعات واكتشاف العلاقات بينها و بين المتطلبات الأخرى ، حيث يقوم المحلل بتبسيط الأفكار حول المعلومات المعطاة أثناء المرحلة الأولى، وهو يحول أفكار المستخدمين حول النظام إلى تقديم تمثيل منظم للنظام كما

يراه. وهذا ربما يسبب بعض التعقيدات (ربما يكون هناك عدد من المستخدمين المختلفين لديهم آراء مختلفة عن احتياجاتهم). حيث إنه في مستهل كل مشروع يوجد على الأقل رأيان هما وجهة نظر مستخدم ووجهة نظر مطور. وربما يكون هناك اختلاف في الرأي بين العملاء ومستخدميهم ، فمثلاً سيهتم المستخدمون بأن يكون النظام مريحاً لهم وذا إمكانات عالية . ربما مبالغ فيها . والتي يمكن أن تؤدي إلى زيادة التكلفة ، وربما سيرفضون أي تسهيلات في النظام لمراقبة معدلات أعمالهم وهو ما قد يرفضه العميل نفسه وربما . أيضاً . لا يصل المستخدمون إلى رأي متجانس فمنهم ما يعتقد أن الحاسوب الآلي يمكن عمل أي شيء فتكون مطالبهم عالية ، وأخرون بسطاء في الاتجاه المعاكس يؤمنون بأن الحاسوب الآلي يمكن أن ينفذ فقط معظم الأعمال العادلة بصورة آلية ومن هنا يبرز دور المحلل باعتباره مفاوضاً لحل هذه الاختلافات للحصول على الموافقة من كل المستخدمين والعملاء على مواصفات متطلبات نهائية. وتؤدي هذه المرحلة إلى المتطلبات المتفق عليها (Agreed Requirements) من مختلف الأشخاص والجهات ذات العلاقة بتطوير النظام (Stakeholders).

وبصفة عامة ، يمكن تلخيص دور المحلل في هذه المرحلة في النقاط التالية:

- المساعدة على حل الاختلافات المحتملة في الآراء بين المستخدمين أنفسهم وبينهم وبين العملاء.
- إسداء النصيحة للمستخدمين بما هو ممكن تنفيذه تقنياً وما هو غير ممكن.
- التفاوض مع المستخدمين والعملاء والحصول على مواصفات متطلبات نهائية.

ومن أهم الإرشادات التي يجب مراعاتها في هذه المرحلة ، ما يلي :

- تعريف حدود النظام (System Boundaries).
- التخطيط للصعوبات المتوقعة وكذلك لطرق حلها.
- استخدام أساليب و أنشطة (Techniques / Activities) مختلفة لإجراء عمليات التفاوض والتحليل ، والتي من أهمها :

- **قوائم تدقيق المتطلبات (Requirements Checklists)** : و تستخد لتأكد من عدم نسيان أي من المتطلبات في جميع مراحل هندسة المتطلبات ، وكذلك التأكد من وضوح المتطلبات و قابليتها للاختبار و عدم وجود أية تعارضات أو تكرارات أو مرادفات بين المتطلبات.
- **تعيين أولوية المتطلبات (Requirements Prioritization)** : وهي تتطلب من مبدأ أنه من الجائز عدم تحقيق كامل متطلبات الـ "Stakeholders" عند بناء النظام ، ويجب تحديد أوليات تنفيذ المتطلبات من قبل الـ "Stakeholders" حسب الأهمية ، وذلك لتسهيل عملية اختيار المتطلبات الضرورية في حالة وجود مقايضة (Trade-Off) بين ما ينجز من متطلبات وبين القيود المفروضة على النظام مثل التكلفة و مواعيد التسليم النهاية.
- **النمذجة غير الاصطلاحية (Informal Modeling)** : وهي تتضمن العديد من الوسائل مثل الوصف النصي (Text Description) ، والصور الغنية بالمعلومات (Rich Pictures) لوصف النظام بطريقة واضحة ومرنة ومفهومه بحيث تُمكِّن الـ "Stakeholders" من استكشاف خفايا النظام والتحاور فيما بينهم وكذلك فيما بينهم وبين فريق التطوير بدون الحاجة إلى معرفتهم بأساليب النمذجة المختلفة.
- **النمذجة شبه الاصطلاحية (Semi-Formal Modeling)** : بالإضافة إلى النمذجة غير الاصطلاحية تُعد النمذجة شبه الاصطلاحية من أهم الأنشطة في عملية توثيق المتطلبات حيث تتضمن العديد من الوسائل التي تستخدم لتدوين البيانات باستخدام مجموعة من الرموز (الاصطلاحات) (Notations) ذات دلالات ومفاهيم معينة مثل مخططات سريان البيانات "DFD" (Data Flow Diagrams) ، و خرائط الحالة (State Charts) ، و لغة النمذجة الموحدة (Unified Modeling Language "UML") . والهدف من النمذجة شبه الاصطلاحية هو نفس الهدف من النمذجة غير الاصطلاحية ألا وهو تطوير بيان يصف النظام بطريقة واضحة

ومنه ومفهومه لتسهيل عمليات الاتصال والتحليل ومساعدة المحلل والمهتمين بالنظام "Stakeholders" على فهم النظام ، ولكنها تختلف في الوسائل التي تستخدم في عمليات التدوين حيث إنها تستخدم مجموعة من الرموز المعروفة سابقاً والتي لها دلالات ومفاهيم وقواعد معينة لوصف النظام.

• عملية التدرج التحليلي (Analytic Hierarchy Process) : وهي عملية تساعد عند اتخاذ قرار اختيار ما بين المتطلبات ، حيث تتم عملية الاختيار المعقّدة من خلال تقسيمها إلى مجموعة من المقارنات البسيطة بين المتطلبات المراد الاختيار فيما بينها وتستخدم عملية التدرج التحليلي في حالة صعوبة تحديد أوليات تنفيذ المتطلبات أو في حالة اختلاف والمهتمين بالنظام "Stakeholders" في اختيار المتطلبات الضرورية.

③ المرحلة الثالثة : توثيق المتطلبات: وهي كتابة بيان واضح غالباً بلغة طبيعية مما يتوقع أن يقدمه النظام للمستخدمين ، وتؤدي هذه المرحلة إلى مسودة الوثيقة النهائية للمتطلبات (Draft Requirements Document) .

ومن أهم الإرشادات التي يجب مراعاتها في هذه المرحلة (بدون ترتيب) ، ما يلي :

- تعريف الهيكل القياسي المستخدم في عملية التوثيق.
- شرح كيفية استخدام الوثيقة.
- تضمين ملخص عن المتطلبات.
- تعريف المصطلحات المستخدمة خلال الوثيقة.
- تصميم الوثيقة بحيث تكون مقروءة للجميع ، وتساعد القارئ على إيجاد المعلومة ، وسهلة التعديل.
- استخدام لغة بسيطة وموজزة.
- استخدام الرسوم التوضيحية على نحو ملائم وواضح.
- وصف المتطلبات بصورة كمية (Quantitatively) إن أمكن ذلك.

- استخدام أساليب / أنشطة (Techniques / Activities) مختلفة لإجراء عمليات التوثيق والتوصيف ، والتي من أهمها : **النمذجة غير الاصطلاحية (Informal Modeling)** ، بالإضافة إلى **النمذجة شبه الاصطلاحية (Semi-Formal Modeling)** ،
- **النمذجة الاصطلاحية (المنهجية) (Formal Modeling)** : وهي تشمل مجموعة من الأنشطة التي تمكن فريق التطوير من وصف متطلبات النظام والتحقق من صحة عمله عن طريق تدوين رياضي دقيق يعتمد على استخدام التحليل الرياضي المنهجي (الصوري) (Formal Mathematical Transformation) ، وتتم تنفيذ جميع عملياتها بطريق آلية ، مع إمكانية إعداد الرسوم البيانية المساعدة للمواصفات كدليل على تحقيقها.. وفي حين تهدف النمذجة غير الاصطلاحية والنمذجة شبه الاصطلاحية إلى تطوير بيان يصف النظام لتسهيل عمليات الاتصال والتحليل ومساعدة المحلل والمهتمين بالنظام "Stakeholders" على فهم النظام ، نجد أن هدف النمذجة الاصطلاحية هو تطوير بيان وصفي للنظام يساعد على التحقق المطلوب من سلوك وخصائص النظام الضرورية.
- **تابعية المتطلبات (Requirements Traceability)** : تشمل على مجموعة من الروابط بين متطلبات النظام وأنواع أخرى من المعلومات والتي من أهمها : **مصدر المتطلب (Requirement Source)** ، و**أسباب المتطلب (The Rationale)** ، **Behind the Requirement** ، وخلافه.
- **إدارة تغيير المتطلبات (Requirements Change Management)** : وهي العملية التي من خلالها يتم مراجعة التغييرات المقترحة في المتطلبات والتأكد من قبولها من "Stakeholders" قبل تضمين هذه التغييرات في وثيقة توصيف المتطلبات ، وبالتالي قبل تنفيذها في النظام. وإدارة تغيير المتطلبات تتضمن أيضاً إدارة التغييرات في النظام وأخطاء النظام الناتجة من التغييرات في المتطلبات.
- **إعادة استخدام المتطلبات (Requirements Reuse)** : وتعني إعادة استخدام مواصفات المتطلبات (أو جزء منها) التي سبق تطويرها في مشاريع سابقة ، مثل :

إعادة استخدام القوالب (Templates) الخاصة بالمواصفات العامة ، أو استخدام مواصفات معينة سبق تطويرها لمشاريع مماثلة.

④ **المرحلة الرابعة المصادقة على المتطلبات :** وهي مسؤولية مختلف الأشخاص والجهات ذات العلاقة بتطوير النظام (Stakeholders). وتؤدي هذه المرحلة إلى إنتاج تقرير عن المصادقة عن المتطلبات (Requirements Validation Report) ، ويُعد هذا التقرير حجر الزاوية في عملية هندسة المتطلبات ، حيث إنه بناءً على هذا التقرير أما أن تقبل مسودة الوثيقة النهائية للمتطلبات المنتجة في المرحلة السابقة وتصبح وثيقة نهائية ، أو يتم إعادة تكرار العملية مرة أخرى ، كما هو موضح بالشكل رقم 4.1. وسوف نتناول التحقق والمصادقة بصفة عامة . إن شاء الله . في الوحدة القادمة ، ولكن سوف نذكر هنا . لجعل الموضوع متكاملاً . بعض من الأساليب والأنشطة (Techniques / Activities) التي يتم استخدامها لإجراء عمليات المصادقة على المتطلبات ، والتي من أهمها هما النموذجة الأولية (Prototyping) بنوعيها : النموذجة الأولية الملقاة جانباً (Throw-Away Prototyping) ، والنماذج الأولية الارتقائية (Evolutionary Prototyping) ، وقد تم مناقشتها في الوحدة السابقة ، وكذلك استخدام قوائم تدقيق المتطلبات (Requirements Checklists) والتي تم شرحها سابقاً خلال هذه الوحدة ، هذا بالإضافة إلى :

- **فحص المتطلبات (Requirements Reviews) :** وتشتمل هذه العملية على فحص الـ "Stakeholders" مواصفات المتطلبات للتأكد من أنها تفي بجميع متطلباتهم ، ومن ثم الاجتماع مع مهندسي فريق التطوير لإطلاعهم على نتائج الفحص والتي تُعد نوعاً من التغذية العكسية (Feedback) التي تساعد فريق التطوير على تلافي القصور الموجود في مواصفات المتطلبات ، وتنتهي هذه العملية بعد موافقة الـ "Stakeholders" على مواصفات المتطلبات المقدمة إليهم .

أسئلة تقويم ذاتي

عدد أهم الأسئلة التي تساعده في استنباط المتطلبات ، وما هي أهم الإرشادات التي يجب مراعاتها في هذه المرحلة .



2.1 مدخلات وخرجات عملية هندسة متطلبات البرمجيات

لإتمام عملية هندسة المتطلبات بطريقة تضمن استنتاج متطلبات تفي وبصورة كاملة باحتياجات كل من المستخدم والعميل يجب توافر بعض المدخلات الضرورية والموضحة بالشكل رقم 4.1 ، وهي كالتالي :

- **معلومات النظام الحالية (Existing System Information)** : وهي معلومات عن الأداء الوظيفي للنظام المطلوب توصيف متطلباته ، وكذلك الأنظمة الأخرى التي يتفاعل معها هذا النظام.
- **حاجات ذوي الشأن (أصحاب المصلحة) (Stakeholder Needs)** : وهي تمثل وصف متطلبات ذوي الشأن أو أصحاب المصلحة (Stakeholder) من تطوير النظام لدعم الأعمال القائمين بها، وبصفة عامة يُعرف ذوي الشأن أو أصحاب المصلحة (Stakeholder) بأنهم أفراد أو مجموعات معينة لها تأثير هام أو يمكن أن تتأثر أو لها مصلحة مباشرة بـ "مجموعة من الأنشطة" المتعلقة بتطوير منتج ما ، أي أنهم هم الأشخاص المهتمين بإنجاز عملية تطوير المنتج ، ويتم تعينهم بشكل طبيعي بناءً على أدوارهم أو وظائفهم وليس بصفتهم الشخصية ، ومن أهم هؤلاء : مهندسو البرمجيات (Software Engineers) المسئولين عن تطوير النظام ، و المستخدمون النهائين (System End-Users) الذين سوف يستخدمون النظام بعد تسليمه ، والمدراء المسؤولون عن المستخدمين النهائين للنظام (Managers of System End-Users) ، والمنظمون الخارجون (External Regulators) الذين يتأكدون من مطابقة النظام

المطور للمواصفات الصحيحة ، و الخبراء (Experts) الذين يعطون المعلومات الأساسية عن نطاق تطبيق النظام.

- **المعايير التنظيمية (Organizational Standards)** : تستخرج هذه المتطلبات من السياسات والإجراءات التي تتبعها مؤسسة العميل في تطوير البرمجيات ، مثل المعايير القياسية المستخدمة في المنظمة والمعنية بتنفيذ وتطوير الأنظمة وإدارة جودتها (من أهم المعايير المستخدمة في مثل هذه الحالات معيار وتعليمات IEEE ، و المواصفات والمعايير العسكرية ، ومعايير ISO) ، ومتطلبات التنفيذ (مثل نوع اللغة البرمجية التي سوف تستخدم) ، وطريقة التصميم والبناء الهيكلي ، ومتطلبات التسليم.
- **التعليمات التنظيمية (Regulations)** : وهي تمثل التعليمات والقيود المفروضة على النظام مثل تعليمات الصحة ، والأمن التي تطبق على النظام.
- **معلومات المجال (Domain Information)** : تستخرج هذه المعلومات من نطاق التطبيق الخاص بالنظام وليس من منطلق احتياج المستخدمين ، حيث تعكس هذه المعلومات خصائص وقيود هذا المجال مثل كيفية تنفيذ العمليات الحسابية الخاصة بالنظام (Computations).

أما مخرجات عملية هندسة البرمجيات ، وكما هي موضحة بالشكل رقم 4.1 فهي كما يلي :

- **المتطلبات المتفق عليها (Agreed Requirements)** : وهي عبارة عن وصف لمتطلبات النظام التي اتفق عليها ذوو شأن أو أصحاب المصلحة (Stakeholder) من تطوير النظام.
- **مواصفات النظام (System Specification)** : وهي عبارة عن وصف تفصيلي للمتطلبات الوظيفية للنظام.
- **نماذج النظام (System Models)** : وهي مجموعة من النماذج التي تصف النظام

مثل : نموذج سريان البيانات (Data-Flow Model) ، و نموذج العمليات (Process Model) ، و خلافه ، والتي تصف النظام من وجهات نظر مختلفة.

2. مفهوم المتطلبات

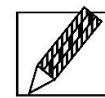
المتطلبات : تُعرف بأنها توصيف النظام الذي ينتج خلال عملية هندسة البرمجيات لخدمات النظام والقيود المفروضة عليه ، والتي يتفق عليها ذوو الشأن أو أصحاب المصلحة (Stakeholder) من تطوير النظام، والمتطلبات تتراوح بين كونها بيانات ملخصة عن الخدمات والقيودو بين كونها مواصفات تفصيلية تصف جميع الجوانب الهامة للنظام وبطريقة واضحة وموصلة ومتربطة وتأخذ بعين الاعتبار تحديد نوعية البيانات التي يجب معالجتها وكيفية إجراء هذه المعالجات وخصوصاً الحسابية منها (Computation) ، والوظيفة والأداء المطلوبين والسلوك المتوقع من النظام ، والواجهات المطلوبة مع الأنظمة الأخرى ، والقيود المفروضة على التصميم ، ومؤشرات التحقق من صحة العمل المطلوبة لتعريف النظام الناجح.

وتحدد معرفة ما يؤديه النظام وكيفية أداء النظام لذلك ، وهو ما يُعرف "بالعلاقة ما بين التوصيف والتطبيق" ، أحد أكثر المسائل الجدلية أثناء عملية تحديد وتحليل المتطلبات، فيرى البعض أن توصيف المتطلبات يجب أن يتتجنب مسائل التطبيق لعديد من الأسباب أهمها أنه من غير المتوقع أن يستوعب المستخدم المسائل الفنية للتطبيق ، وكذلك عدم وضع العقبات التي قد تعيق تطوير المشروع، ومن جانب آخر يرى البعض أنه من غير الممكن الفصل بين التوصيف والتطبيق وأنهما متداخلان ومتربطان في العديد من الطرق والأساليب الرئيسية. ففي تلك الطريقة ، يُعد استيعاب وتوثيق أعمال ومهام النظام المتاح فعلها (وقد يكون ذلك نظام يدوي أو نظام قائمة

على الحاسب الآلي أو على كيهما) يعد أول مهمة في تطوير النظام الجديد، وبالتالي فإن تطبيق أحد الأنظمة يعد الجزء الأساسي في التوصيف لوضع نظام جديد. فمثلاً إذا افترضنا أنه لدينا الرغبة في تطوير ووضع نظام آلي لأعمال الاختبارات ، فإحدى الطرق أن نقوم بالدراسة والتوثيق لكافة الإجراءات الحالية لأعمال الاختبارات الآلية منها واليدوية وعلى كافة المستويات ، وبعد الانتهاء من إنجاز هذه المهمة تأتي الخطوة الثانية كي نقرر الإجراءات يجب حosisتها (أي إدخالها حاسوبياً) ، والتأكد من إمكانية تحقيق المطلوب من الناحية الفنية، وأخيراً فإن توصيف النظام الجديد مشتق من تصميم (تطبيق) النظام السابق أو القديم، وتبدو هذه الطريقة من التطوير منطقية وعملية جداً ، ولكن لا تعني أن التطبيق والتوصيف متداخلان.

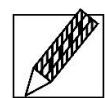
تدريب (1)

ما الفرق بين المتطلبات والتصميم ؟



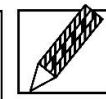
تدريب (2)

ما الفرق بين متطلبات النظام ومتطلبات البرمجيات ؟



تدريب (3)

ما الفرق بين متطلبات العميل و متطلبات المطور ؟



وللقاء مزيد من الضوء على مفهوم المتطلبات سوف نذكر بإيجاز بعضًا من المقارنات البسيطة التالية :

ما الفرق بين متطلبات العميل ومتطلبات المطور؟

- متطلبات العميل : تُعد متطلبات إجبارية ، ويجب أن تتحقق في المنتج النهائي .
وكمثال لمتطلبات العميل فإن البيانات يجب أن تخزن في ملف وبدون استخدام أي قواعد بيانات خارجية.
- متطلبات المطور : تُعد متطلبات داعمة لتوقعات وطلبات العملاء ، ولنشاطات وخيارات التصميم ، وتشاً هذه المتطلبات . عادة . من القيود المفروضة من بيئة التطوير على النظام، وكمثال لمتطلبات المطور التي تدعم طلب العميل السابق فإن البيانات يجب أن تخزن في ملف فهرس متتالي لمناسبة هذا النوع من الملفات للنظام الذي يتم تطويره.

ولكن لماذا نحتاج المتطلبات ؟

- لتقدير أفضل للاحتياج وأسبابه لدى كل من المستخدمين والعملاء وال محللين.
- للحصول على المعلومات الضرورية والكافية لمراحل التصميم والتنفيذ اللاحقة.
- لاكتشاف المتطلبات المتناقضة (Inconsistent Requirements) والمشاكل المحتمل ظهورها أثناء مرحلة التنفيذ ، وذلك في مرحلة مبكرة.
- تقدير التكلفة ، ومواعيد التنفيذ ، والموارد المطلوبة ، وخلافه.
- لتفادي المشاكل التي قد تنشأ في مراحل التصميم والتنفيذ والصيانة ، والتي من أهمها ما يلي :

- ◆ المتطلبات لا تعكس الأهداف الرئيسية التي من أجلها تم تطوير النظام.
- ◆ المتطلبات متناقضة أو ناقصة.
- ◆ زيادة التكلفة في حالة إحداث تغييرات للمتطلبات بعد الانفاق عليها.
- ◆ الفهم الخاطئ لمتطلبات المستخدمين والعميل من قبل فريق التطوير.

- ♦ تجاوز الميزانية المقررة أو استغرق وقت أطول في عملية التنفيذ من التوقعات المبدئية لها.
- ♦ معاناة فريق التطوير من إعادة العمل بسبب أخطاء أو نقص في المتطلبات.
- ♦ افتقار الأنظمة الناتجة إلى بعض الوظائف أو لها وظائف غير مستعملة.
- ♦ ارتفاع تكاليف صيانة النظام.
- ♦ صعوبة تقويم تكلفة التغييرات أو تطوير النظام.

ولمن يتم تحديد المتطلبات ؟ وفيما تستخدمها ؟ :

- يتم تحديد المتطلبات للاستفادة منها من قبل العديد من الأشخاص بمختلف خلفياتهم ومعرفتهم التقنية و على سبيل المثال العملاء ، والمستخدم النهائي ، و المحللون ، و المصممون ، و المهندسون ، و المديرون.
- **عملاء النظام ومستخدموه:** يستخدمون المتطلبات لفحصها للتحقق من مناسبتها لتلبية احتياجاتهم والمصادقة عليها و كذلك لإدخال التعديلات المطلوبة في حالة الضرورة.
 - **محلي النظم :** وهو يستخدمون المتطلبات للتحقق من صحتها وتكاملها وتوافقها مع باقي مستندات المشروع.
 - **مدراء المشروع :** وهو يستخدمون المتطلبات لغرض التخطيط للنظام و تطويره والتأكد من تسليم المشروع في الوقت المحدد وبمواصفات ومقاييسها الجودة المطلوبة.
 - **مهندسي التصميم والتطوير :** وهو يستخدموا المتطلبات لتصميم وتطوير النظام بما يتلاءم معها ويحققها بالكامل.
 - **مهندسو اختبار النظام :** وهو يستخدمون المتطلبات لتطوير مقاييس الاختبار التي سوف يستخدمونها في التحقق والمصادقة على صحة النظام.
 - **مهندسو صيانة النظام :** وهو يستخدمون المتطلبات لمساعدتهم على فهم النظام والعلاقة بين أجزائه المختلفة لإجراء عمليات الصيانة اللازمة بكل سهولة وكفاءة وبأقل التكاليف.

ما هي الخصائص المميزة للمتطلبات الجيدة؟

- قابلة للإثبات (Verifiable) : يجب أن ثُعَرَفَ المتطلبات بكلمات دقيقة في المعنى يسهل التحقق منها وقابلة للإثبات ، وليس مُعرَفة بكلمات غير دقيقة يصعب التتحقق من تنفيذها مثل : مقاومة (Resistant) ، كافية (Sufficient) ، مفرطة (Excessive) ،الخ.
- واضحة ومفهومه (Unambiguous and Understandable) : أي أن جميع كلماتها وجملها تحتمل معنى واحداً فقط.
- كاملة (Complete) : تغطي جميع المهام المطلوبة من حيث : مبادئ الصيانة والتشغيل ، بيئات التشغيل ، والقيود المفروضة على كل مهمة ، وخلافه ، بحيث يمكن تنفيذها بدون حاجة المصممين والمطوريين للاستفسار عن أي شيء فيها.
- صحيحة (Correct) : تؤدي في النهاية إلى تحقيق كامل متطلبات المستخدمين والعميل بصورة صحيحة.
- محددة ومحضرة (Definitive and Abstractive) : تحدد ما هو المطلوب ولماذا ، وذلك بكل دقة ووضوح وباختصار من خلال الإجابة الدقيقة على الأسئلة التي تبدأ بكلمة "ما هو/هي" (What) ، وكذلك الأسئلة التي تبدأ بكلمة "لماذا" (Why) ، وليس الإجابة على الأسئلة التي تبدأ بكلمة "كيف" (How).
- متوافقة (Consistent) : متسقة مع المتطلبات الواردة في مستندات أخرى لغير المشروع.
- قابلة للتعديل (Modifiable) : مكتوبة بطريقة منظمة بحيث يمكن تعديلها أو تغييرها بسهولة ويسر.

- **جزأة (Partitioned) :** مقسمة إلى أجزاء متراكبة تتاسب مع تدرج مستوى هيكلية النظام (System Hierarchy) ، يمكن تتبعها (Traceable) وفهمها بسهولة ويسر.
- **قابلة للاستخدام (Usable) :** قابلة للتنفيذ من قبل المصممين بسهولة ووضوح ما هي الفوائد الناتجة من تحديد المتطلبات بصورة جيدة؟
- خفض أو إزالة إعادة أنشطة التصميم وتسلیم المنتج البرمجي.
- تحسين الجودة.
- التحكم في المشروع في جميع مراحله.
- تخفيض النزاعات مع العميل طوال وقت تنفيذ النظام.
- تخفيض التكاليف.
- الإبقاء بالجدوى الزمنية لتنفيذ المشروع.
- إزالة احتمال حدوث فهم غير دقيق للمتطلبات.
- زيادة التواصل والتفاهم والتعاون والمساندة بين العميل وفريق التطوير.
- إزالة الاعتماد على أناس محددين في عملية تطوير المنتج لتخفيض احتمال المخاطرة عند تركهم لأعمالهم.

أسئلة تقويم ذاتي

ما هي الخصائص المميزة للمتطلبات الجيدة؟.

ما هي أهم الفوائد المترتبة على تحديد المتطلبات بالصورة الجيدة؟



3. أنواع المتطلبات (Types of Requirements)

بصفة عامة يمكن تقسيم المتطلبات إلى نوعين رئيسيين هما :

- متطلبات وظيفية (Functional Requirements).
- متطلبات غير وظيفية (Non-Functional Requirements).

1.3 المتطلبات الوظيفية (Functional Requirements)

هي عبارة عن وصف للخدمات التي يجب أن تقدمها البرمجية بالتفصيل ، وكيفية استجابتها لمدخلات محددة ، وكيفية سلوكها في حالات محددة ، وتعتمد هذه المتطلبات . في المقام الأول . على نوع البرمجية نفسها وعلى المستخدمين النهائيين لها ، وكذلك على نوع النظام الداخلة في تكوينه والمتطلبات الوظيفية - في بعض الحالات - ربما تتضمن ماذا يجب آلا يفعله النظام ، مثل : المتطلبات الخاصة بالأمان (Security Requirements).

وتحدد المتطلبات الوظيفية هي الجوهر الأساسي لمواصفات المتطلبات أنها تبين ماذا يجب أن يفعله النظام حيث إنها تتميز بالأفعال التي تؤدي عملاً ما ، والأمثلة هي :

- النظام يجب أن يُظهر عناوين الكتب التي كتبها مؤلف معين.
- النظام يجب أن يُظهر باستمرار درجة حرارة كل المكائن.
- يجب أن يكون المستخدم قادرًا على البحث في جميع قاعدة البيانات الفعلية الأولية أو اختيار مجموعة فرعية منها.
- كل مدخلة يجب أن يخصص لها معرف مفرد (ORDER_ID) والتي تمكن المستخدم من التعامل معه بسهولة من حيث البحث والتخزين وخلافه.

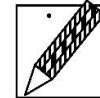
2.3 المتطلبات غير الوظيفية

. (Non-Functional Requirements)

هي عبارة عن وصف للقيود المفروضة على الخدمات التي يجب أن يقدمها النظام ، مثل التكلفة ، و تاريخ التسلیم ، و كمية الذاكرة المتوفّرة ، و كمية التخزين المدعومة المتوفّر ، و وقت الاستجابة ، و لغة البرمجة و طريقة التطوير المستخدمة (Particular CASE Development System) ، و أحجام البيانات ، و اعتمادية النظام ، و قدرات وحدات الإدخال والإخراج ، و خلافه . وقد تكون المتطلبات غير الوظيفية أكثر أهمية من المتطلبات الوظيفية لأنّه إذا لم يتم تحقيق المتطلبات غير الوظيفية فلا فائدة للنظام ، وعادة تهتم المتطلبات غير الوظيفية بالخصائص النوعية (Qualitative Features) الخاصة بالمنتج البرمجي والتي يمكن قياسها مثل الاعتمادية (Reliability) ، و قابلية التقليل (Portability) ، و المثانة (Robustness) (قدرته على متابعة العمل في جميع الظروف حتى تلك غير المتوقعة) ، والأمان (Ease of Use) ، و سهولة الصيانة (Security) ، و السرعة (Speed) ، و الحجم (Size) ، و خلافه . والجدول رقم 4.1 يبيّن أهم تلك الخصائص وطرق قياسها .

تدريب (4)

في الحقيقة ، ليس من السهل التمييز بين المتطلبات الوظيفية وغير الوظيفية ، بمعنى أنه يمكن أن تظهر متطلبات وظيفية وكأنها متطلبات غير وظيفية ، اشرح هذه العبارة مع إعطاء مثال على ذلك



جدول رقم 4.1 : أهم خصائص وطرق قياس المتطلبات غير الوظيفية

الخاصية	القياس
الاعتمادية	تقاس بـ : متوسط الوقت الذي يحدث من بعده عطل أو فشل في النظام ، و احتمالية عدم جاهزية النظام ، و معدل حدوث عطل أو فشل في النظام ، و جاهزية النظام للعمل .
المتانة	تقاس بـ : الوقت اللازم لتشغيل النظام بعد حدوث عطل أو فشل ، النسبة المئوية للخسارة نتيجة فشل النظام ، و احتمالية تخريب البيانات (Data Corruption) بسبب فشل النظام .
الحجم	ويقاس بسعة رقاقة الذاكرة العشوائية المطلوبة لتشغيل النظام (كيلو بايت) . وكذلك يمكن أن يقاس بعدد سطور التعليمات البرمجية التي يتكون منها النظام .
السرعة	تقاس بـ : عدد الإجراءات التي تعالج في الثانية ، وقت الاستجابة للمستخدم أو لحدث ما ، وقت تجديد الشاشة .
سهولة الاستخدام	تقاس بـ : وقت التدريب اللازم لإتقان العمل على النظام ، وعدد أشكال المساعدة الفورية أثناء العمل على النظام .
قابلية النقل	تقاس بعدد أنظمة التشغيل التي يمكن تشغيل النظام من خلالها بدون أي تعديلات تذكر في التعليمات البرمجية الخاصة بالنظام .

ويمكن تصنيف المتطلبات غير الوظيفية ، كما هو موضح بالشكل 4.2 ، على النحو التالي :

- **متطلبات المنتج (Product Requirements)** : وهي المتطلبات التي تحدد سلوك المنتج البرمجي النهائي مثل : سرعة التنفيذ (Execution Speed) ، الاعتمادية (Reliability) الخ.
- **المتطلبات التنظيمية (Organizational Requirements)** : وهي المتطلبات الناتجة عن السياسات والإجراءات التنظيمية (Organizational Policies and Procedures) الخاصة بالشركة أو المؤسسة التي يطورها لها المنتج البرمجي ، مثل : المعايير القياسية المستخدمة في عملية التطوير (Process Standards) ، ومتطلبات التنفيذ (Implementation Requirements) الخ.

- **المتطلبات الخارجية :** (External Requirements) : وهي المتطلبات التي تنشأ من عوامل خارجية للنظام ولعملية التطوير الخاص بها ، مثل المتطلبات التشريعية فمثلاً يجب أن لا يفشي النظام أية معلومات شخصية عن العملاء بغض النظر عن أسمائهم وأرقام هواتفهم) إلى مشغلي النظام.

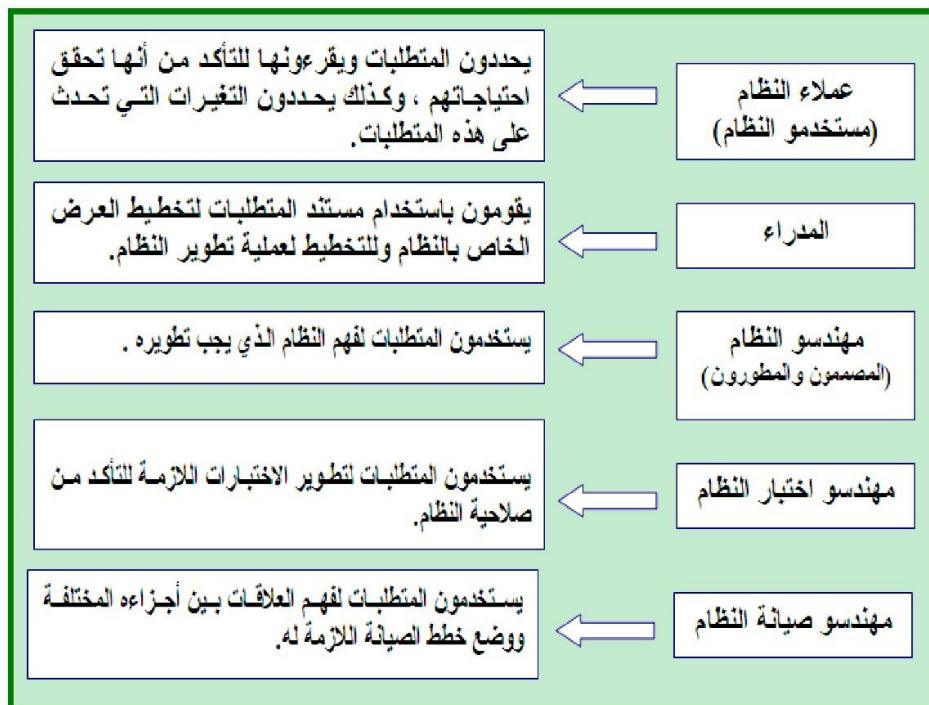
وقد تكون المتطلبات غير الوظيفية صعبة جداً بحيث لا يمكن وصفها بدقة ، ولكن المتطلبات غير الدقيقة (الغامضة) من الصعب التأكد من صحتها ، فعلى سبيل المثال عند تحديد الهدف من النظام بالقول " يجب أن يكون النظام سهل الاستعمال من قبل مراقبين ذوي خبرات ، وكذلك يجب أن يكون منظماً بحيث يقلل من أخطاء المستخدمين " بالتدقيق في منطوق هذا الهدف نجد أنه غامض وغير قابل للقياس والتحقق من تفيذه ، وحيث إن تحديد أهداف النظام مفيدة للمطورين لأنها تظهر الرغبات والاهتمامات الحقيقية لمستخدمي النظام ، لذا يجب أن يكون منطوق الهدف واضحاً وقابلأً للقياس ، فمثلاً يمكن تعديل المنطوق السابق إلى منطوق واضح وقابل للقياس بالقول : " يجب أن يكون المراقبين من ذوي الخبرة قادرين على استخدام جميع وظائف النظام بعد تدريب حوالي ساعتين وبعد هذا التدريب فإن متوسط عدد الأخطاء التي يرتكبها المستخدمين من ذوي الخبرة يجب أن لا تزيد عن خطأين في اليوم " .

4. وثيقة مواصفات المتطلبات

(Requirements Specification Document)

المنتج النهائي لعملية هندسة المتطلبات هو وثيقة مواصفات المتطلبات ، وهي ضرورية لنجاح وتطوير أي مشروع تطوير برمجية، فإذا لم نحدد بوضوح ما يستطيع أن يفعله النظام فكيف يمكننا أن نطور البرمجية وبأي ثقة. وكيف نأمل أن يتتوافق المنتج النهائي مع احتياجاتنا؟ . ووثيقة المواصفات هي الوثيقة التي يمكن الرجوع إليها عند تقويم أي تطوير ، وهي عنصر رئيسي لأي عقد ارتباط شرعي، ومن أهم العوامل التي يجب مراعاتها عند كتابة وثيقة مواصفات المتطلبات هو الأخذ بعين الاعتبار مستخدم هذه الوثيقة (أنظر شكل 4.3) ، لكي يفهمها كل من له صلة في تطوير النظام باختلاف مستوياتهم من حيث الخلفيات العلمية والخبرات العملية ، وخصوصاً المستخدمين والمطورين.

وبرغم أن كلاً من هؤلاء يشتكون في الهدف المشترك وهو الوصف الواضح لمتطلبات النظام ، فإنهم سيضطرون لاستعمال لغات مختلفة ، حيث إن خلفياتهم العلمية وخبراتهم العملية مختلفة، فمثلاً سيفضل المستخدمون وصفات غير تقنية بلغة طبيعية ، وبالرغم من أن تلك اللغات تكون ممتازة في القصائد وكتابة الخطابات ولكنها فقيرة وضعيفة في عرض المواصفات الدقيقة والمتكاملة والواضحة ، من الناحية الأخرى فإن المحللين لكونهم أصحاب خبرة تقنية سيفضلون استخدام رموز دقة (ريما رياضية) لكي يصفوا النظام.



شكل 4.3 : مستخدمو وثيقة مواصفات المتطلبات

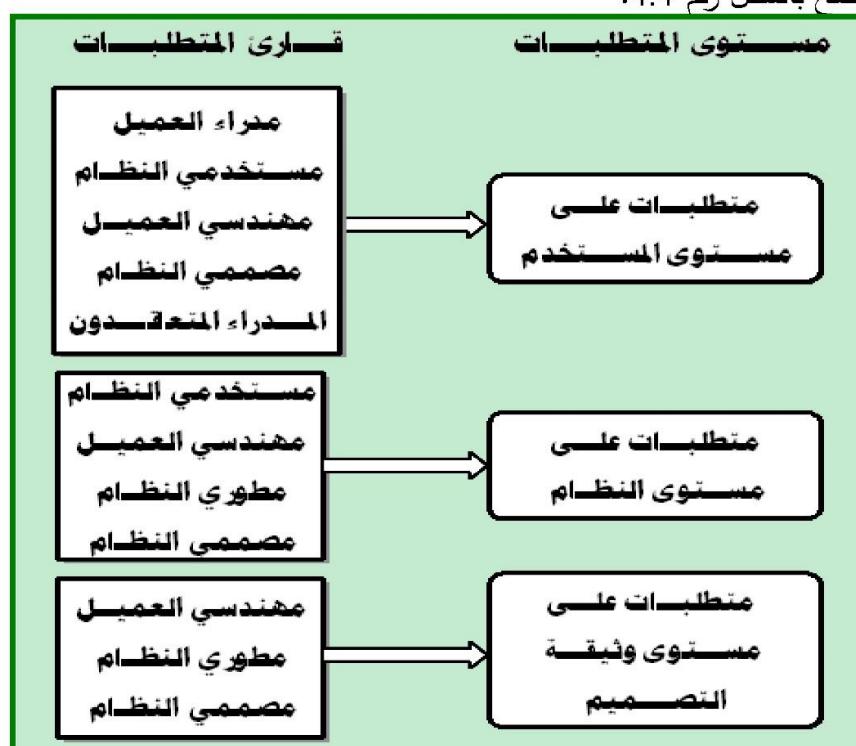
هناك رموز متعددة لكتابة المواصفات:

- الكتابة غير الرسمية (Informal) : وهي تتم باللغة الطبيعية حيث ستستخدم بوضوح وبعانية قدر الإمكان.
 - الكتابة الرسمية (Formal) : حيث تستعمل الرموز الرياضية بكل دقة وباختصار.
 - الكتابة شبة رسمية (Semiformal) : حيث تتم باستعمال مزيج من اللغة الطبيعية مع رموز وجداول ورسوم مختلفة.
- ومعظم هذه الرموز لديها أصول في الطرق لتصميم برمجية حيث تشمل الشفرة الكاذبة (Pseudo Code) ، ورسوم تدفق البيانات (Flow Charts) ، وخلافه. وفي

الوقت الحالي فإن معظم مواصفات البيانات تكتب بلغة طبيعية مع رموز شبه رسمية مثل رسومات تدفق البيانات المنشورة حيث تستخدم أحياناً لتوضيح نص اللغة الطبيعية.

1.4 مستويات المتطلبات (Levels of Requirements)

لتوصيف متطلبات النظام بصورة جيدة يفهمها كل من يستخدم هذه الوثيقة ب مختلف مستوياتهم الوظيفية وخبراتهم العملية يجب تقسيم وثيقة مواصفات المتطلبات إلى مستويات مختلفة من التفاصيل لكي تتناسب مع الاهتمامات الخاصة لكل من يقرأها ، مثل المستخدم النهائي والمدير التنفيذي والمطور ، وخلافه ، كما هو موضح بالشكل رقم 4.4.



شكل 4.4 : شكل توضيحي لأهم مستويات المتطلبات وقارئها.

ومن أهم تلك المستويات ما يلي :

▪ مطلبات المستخدم (User Requirements) .

مطلبات المستخدم للنظام ينبغي أن تصف المطلبات الوظيفية وغير الوظيفية لكي تكون مفهومة لمستخدمي النظام بدون المعرفة التقنية المفصلة له، ويجب أن تشمل سلوك النظام الخارجي وتجنب بقدر الإمكان خواص تصميم النظام، ويراعى عند كتابة مطلبات المستخدم أن لا تستعمل المصطلحات البرمجية (نظم الترميز) الرسمية أو وصف المطلبات بواسطة وصف تفاصيل النظامو ينبغي أن تكتب مطلبات المستخدم باللغة البسيطة ، وبجداوٍ وأشكال بسيطة ورسوم بيانية واضحة.

وقد تظهر العديد من المشاكل المختلفة عندما تكتب المطلبات بجمل لغة

طبيعية في نص المستند للأسباب التالية :

① **نقص الوضوح (Lack of Clarity)** : أحياناً يكون من الصعب استخدام اللغة الطبيعية بطريقة دقيقة وواضحة بدون جعل الوثيقة مفصلة وصعبة القراءة.

② **التشویش بين المطلبات (Requirements Confusion)** : حيث أنه يكون من الصعب التمييز بوضوح بين المطلبات الوظيفية وغير الوظيفية ، وكذلك يبين أهداف النظام ومعلومات التصميم.

③ **دمج المطلبات (Requirements Amalgamation)** : حيث إنه يتم التعبير عن عدة مطلبات مختلفة كمتطلب واحد.

ولقليل سوء الفهم عند كتابة مطلبات المستخدم ، يوصي البعض باتباع التوجيهات البسيطة التالية :

① تحديد شكل قياسي لكتابية المطلبات والتأكد من أن كل تعريف للمتطلب يناسب إلى ذلك الشكل و توحيد الشكل يجعل المحفوظات أقل احتمال والمتطلبات أسهل للفحص، فعلى سبيل المثال يعرض الشكل المتطلب الأولى بخط كبير، والتفاصيل بخطوط مختلفة متضمنة أسباب كل متطلب ومعلومات عن مقترن المتطلب (مصدر المتطلب) ، لكي تعرف من تستشير في حال تغير المطلبات.

مرجع المحاضرة : كتاب هندسة البرمجيات - تأليف د.أحمد شعبان دسوقي - أ.د.السيد محمود الريبي

- ② استعمل لغة على نفس النهج ، وينبغي أن تميز دائمًا بين المتطلبات الإجبارية والمفضّلة المتطلبات الإجبارية هي المتطلبات التي يجب على النظام أن يُدعمها ، أما المتطلبات المفضّلة هي المتطلبات المرغوب فيها ولكنها غير أساسية.
- ③ استخدم إبراز النص (أسود عريض ، مائل أو لون) لتنقّط الأجزاء الأساسية من المتطلبات.
- ④ تجنب بقدر الإمكان ، استعمال المصطلحات العلمية التقنية لعلوم الحاسوب الآلي ، وفي حالة استخدامها يجب تعريفها بدقة ووضوح.

▪ متطلبات النظام (System Requirements) .

تُعد متطلبات النظام امتداد لمتطلبات المستخدم والتي يستخدمها مهندسو البرمجيات كنقطة انطلاق لتصميم النظام حيث تضيف تفاصيلًا وشراً لكيفية تناول متطلبات المستخدم ، ويمكن استخدامها كجزء من العقد لاستخدام النظام ، لذا يجب أن تكون كاملة ومتواقة مع النظام الكلي.

وعوما فإن متطلبات النظام تصف ببساطة سلوك النظام الخارجي والقيود الموضوعة عليه ولا يجب أن تطرق إطلاقا إلى كيف يتم تصميم النظام. وتستخدم اللغة الطبيعية لكتابه كل من مواصفات متطلبات النظام وكذلك متطلبات المستخدم حيث تُعد مواصفات النظام أكثر تفصيلا عن متطلبات المستخدم.

تدريب (5)

اشرح الصعوبات في استعمال اللغة الطبيعية لوصف المتطلبات.



وعوما فإنه يحبذ كتابة متطلبات المستخدم بلغة يفهمها غير المتخصصين في حين يمكن كتابة متطلبات النظام ، بحيث تشتمل على أنماط مختلفة كالمبينة بالجدول رقم 4.2 ، مثل استخدام : قوالب للغة الطبيعية ، ولغات وصف برمجية ، والبيانات الرسومية ، والتمثيل الرياضي.