Business Requirements أو User Requirements

Information Finding أو Information Gathering أو Requirements Discovery

Grouping أو Clusters أو Requirements Classification

Arrangement أو Requirements prioritization

what the system should do أو Function Requirements

what the system should have أو Non-Functional Requirements

وكنّا قد بدأنا في المحاضرة السابقة بالـRequirements Engineering والتي هي مجموعة من الطرق والنشاطات في عملية جمع متطلبات المنتج البرمجي وتعرفنا على المراحل التي تمر بها عملية " هندسة المتطلبات " وقد توقفنا عند مرحلة Requirements Analysis والآن سندخل بها بشكل تفصيلي لكن قبل ذلك طرح الدكتور سؤال هام كنا قد ذكرناه في المحاضرة السابقة وهو:

في أي مرحلة تتم عملية الـRequirements Engineering؟

تشمل المراحل التي لها علاقة بالـ Requirements بشكل أساسي ولكن بنسب مختلفة وهذه المراحل هي :

Communication – Planning – Analysis

والآن نكمل بشرح باقي المراحل الأساسية للـ Requirements Engineering :

Elicitation – Analysis (specification) – Validation – Management

2- Requirements Analysis (Specification):

هي المرحلة الخاصة بتوثيق الخصائص والـ Requirements للمشروع بشكل موجه لكلاً من الزبون Customer والمطوّر Developer .

كما أنها تسمح للمحللين Analysts بتوضيح المتطلبات الأساسية التي يعتمد عليها المشروع وتعطي إمكانية بناء نماذج تصف تصور المستخدم للمشروع والوظائف التي يقوم بها والمشاكل التي يواجهها والعلاقة بينهما, وهذه النماذج تعطي صورة عن سلوك النظام والمعطيات التي يتعامل معها وكل ذلك من خلال عملية (Documentation - التوثيق).

عند القيام بعملية التوثيق يجب الوضع بعين الاعتبارعدة نقاط مهمة وهي:

a. متطلبات المستخدم يجب أن تكون مفهومة للمستخدمين النهائيين والزبائن الذين من الممكن ألا يكون لديهم أي معرفة تقنية من قبل.
b. System Requirements يتم كتابتها بشكل أكثر تفصيلاً كما يمكن أن تحوي على معلومات تقنية أكثر من الـ User requirements.
c. المتطلبات يمكن أن تكون جزء من العقد لمطوري النظام لذلك من المهم أن تكون كاملة قدر الإمكان.

وبالتالي يجب اتباع طرائق توثيق معيارية ودقيقة ومضمونة لأن أي خطأ يحدث في هذه المرحلة سيؤدي إلى فشل المشروع.

إذاً ماهي الطرق اللازمة لتوثيق هذه المعلومات والمتطلبات ؟

من أهم الطرق التي لدينا هي الطرق الكتابية أو السردية الـ Textual والتي يمكن أن تصنف تحت:

☑ **Natural Language:**

تعني توثيق ما تم جمعه في تقرير منسق ومكتوب بلغة طبيعية Natural Language حيث يمكن أن نعطي تعابير وتوضيحات كثيرة وكل متطلب يمكن صياغته في جملة واحدة مرقمة.

إلا أن هذه الطريقة غير محبذة لأنها في بعض الأحيان قد تكون غير واضحة لكلاً من المستخدمين و المطورين, كما أن المتطلبات الوظيفية والغير وظيفية يمكن أن تندمج مع بعضها في هذا التقرير.

☑ **Structured Natural Language:**

ظهرت لمعالجة المشاكل التي تظهر عند استخدام الـ **Natural Language**,حيث تم اقتراح وضع بنية أو قالب **Template** موحد وكل جزء من هذا النموذج يعبر عن جزء من المتطلبات على أن يتم الالتزام بهذا القالب في كتابة التقارير.
وفي حال وجود نقص أو خطأ يمكن معالجة ذلك إمّا بإضافة ملحق **Appendix** أو بتغيير القالب وهذا يعتبر من مساوئ الـ**Structured Natural Language .**

☑ **Design Description Languages:**

مع أن استخدامها تراجع إلى حدا ما إلا أنها كانت قادرة على كتابة المتطلبات بلغة خاصة مثل اللغات البرمجية أي لها مصطلحات وتعابير خاصة بها فقط وعلى الرغم من أنها قادرة على التوصيف بشكل أكبر إلا أنها تحتاج إلى أشخاص محددين للتعامل معها, كما يجب مواكبة تطور مراحل هذه اللغة التي من الممكن أن تكون غير معممة على جميع الشركات وبالتالي تم إهمالها.

ولدينا أيضاً من الطرق الأخرى :

☑ **Graphical Notations: الطريقة الرسومية**

التعبير عن المتطلبات باستخدام الرسم أفضل من استخدام الكتابة حيث يؤمن الرسم ميزتين أساسيتين وهما:

- **Compression of The Data** ضغط البيانات:

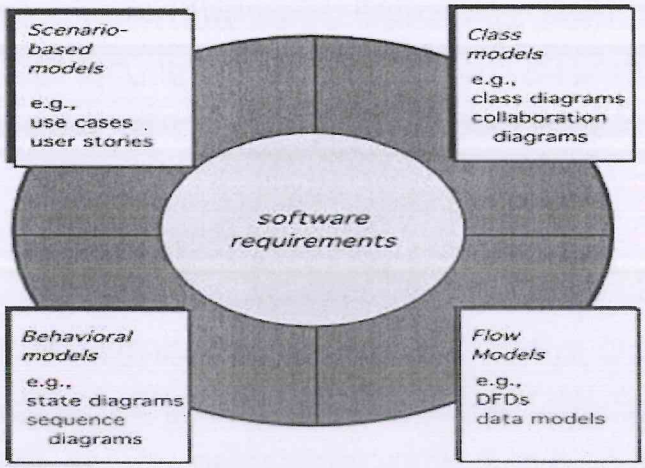
أي عندما نقوم بشرح فكرة باستخدام الكتابة يمكن أن تستغرق ثلاث أو أربع صفحات بينما إذا عبرنا عنها باستخدام الرسم يمكن أن يأخذ حجم الرسمة أو الصورة فقط ويوصل الفكرة نفسها وبالتالي يتم تخفيف الضغط عن المستخدم والمطور.

- الرسم يوصل المعلومة بشكل دقيق بشرط أن توفر معلومات شاملة وأن تكون الرموز مفهومة المعنى.

وبالتالي يجب استخدام رموز محددة ومعيارية يمكن أن يفهمها كل من يقرأها في جميع أنجاء العالم ولذلك تم الاتفاق على مايسمى

UML (Unified Modeling Language)

وهي لغة نمذجة موحدة ومعيارية تستخدم للتعبير عن الرموز والأشكال والمخططات بحيث يفهمها الجميع.



ويمكن النظر إلى المتطلبات (من ناحية النمذجة والرسم) من أربع وجهات نظر **Perspectives** :

Scenario-Based Models: النماذج المعتمدة على السيناريوهات مثل الـ **Use cases** و الـ **User stories**.

Behavioral Models: النماذج السلوكية أي التي تعتمد على سلوك الكانات **Entities** في النموذج لدينا.

Flow Models: النماذج التي تعتمد على تدفق البيانات.

Class Models: النموذج التركيبي.

وجهات النظر هذه **Perspectives** سنتعرف عليها بالتفصيل وستكون عناوين للمحاضرات القادمة بإذن الله.

☑ Mathematical Specifications:

وتسمى أيضاً الـ Formal Language حيث يتم وضع توصيف للمتطلبات بطريقة تصفها بشكل دقيق فلا يمكن أن يتواجد توصيف غيره عن طريق استخدام الرياضيات المنطقية Logical Math أو التعابير الرياضية Expressions أو النظريات Theories.
فاذا كانت لدينا القدرة في استخدام التعابير الرياضية للتعبيرعن جميع المتطلبات يمكن استخدام Auto Code-Generation توليد الكود البرمجي بشكل آلي وهذا موضوع بحثي قيد التطوير حالياً و يطلق عليه Model Driven Development.

بعد أن تعرفنا على بعض الطرق في توثيق المتطلبات كيف سيكون شكل المستند Document الذي سينتج لدينا ؟

سنعرض الآن اقتراح لهذا المستند :

Preface: صفحات بداية (يختلف عن المقدمة) يمكن أن يحوي على نسخة المستند الحالي Version 1 أو Version 2

تسمى Revision History.

Introduction: مقدمة تتحدث عن النظام المطلوب والشركة التي تطلبه والـ Business Outcome المتوقع منه.

Glossary: مجموعة المصطلحات والكلمات المستخدمة في المستند.

User Requirements Specification: الجزء الخاص بمتطلبات المستخدم User Requirements

System Architecture: نظرة عامة لبنية النظام.

System Requirements Specification: الجزء الخاص بمتطلبات النظام System Requirements.

System Models: النماذج والمخططات التي ستوثق كل ماذكر سابقا".

Appendices: الملحقات.

Index: الفهرس.

في سورية يتم التوجه نحو المستندات الـLess Models More Text أما الشركات فتتوجه نحو More Models Less Text.

3- Requirements Validation:

يتم في هذه المرحلة التأكد من أن جميع التفاصيل والمتطلبات التي يريدها المستخدم قد تم توثيقها حيث على المتطلبات تحقيق مايلي:

Validity: المتطلبات مطابقة لما يريد الزبون.

Consistency: التكامل.

Completeness: الكمالية.

Realism: واقعية الحلول والمتطلبات والقدرة على انجازه.

Verifiability: إمكانية الوصول لكود برمجي يحقق المتطلبات.

ويمكن تنفيذ هذه المرحلة من خلال عدّة طرق :

- المراجعات Reviews :

حيث يقوم بالتحقق من الشروط السابقة مجموعة من الأشخاص من قسم ضمان الجودة Quality Assurance يعتبرون مستوى عالي من المطورين من حيث الاختصاص والخبرة وسنطلق عليهم Reviewers.

- النماذج الأولية Prototypes:

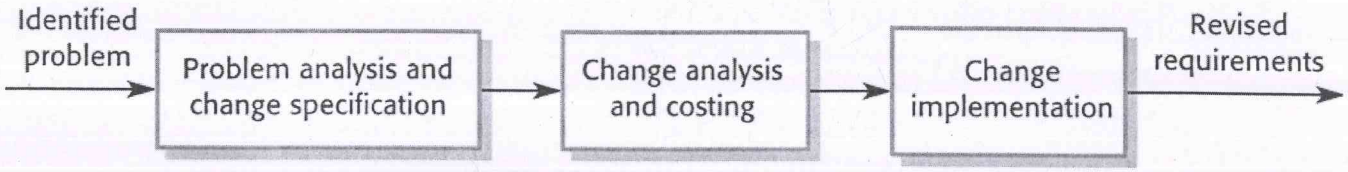
يتم وضع نموذج أولي للمتطلبات وعرضه على المستخدم للتعبير عن رضاه من المتطلبات.

- Test-Case Generation:

يتم استخراج حالات اختبار Test Cases يمكن تطبيقها على الكود البرمجي لاحقاً, أي التحقق من قدرة المستند على إعطاء حالات اختبار Testability وتتم عملية الاستخراج من قبل فريق الـQuality Assurance أيضاً.

4- Requirements Management:

هي إدارة عملية جمع وتصفية **Refinement** ونقاش المتطلبات ومن أهم مراحل الـ**Management** هي القدرة على معالجة التعديلات والتغييرات **Changes** التي تظهر مع تقدّم عملية التطوير وتحديد أهمية هذا التغيير وإمكانية تلبيته ويتم ذلك بـ:



Problem Analysis and Change Specification: يتم تحليل هذا التغيير والتحقق من أنه متطلب أساسي قابل للتحقيق.

Change Analysis And Costing: ندرس عملية إصافة هذا التغيير وكم سيكلف من الوقت ومن التكاليف الخاصة بالتعديل من ثم مناقشتها مع الزبون

Change Implementation: تضمين هذا التغيير في المستند الجديد.