تحدّث الدكتور في بداية المحاضرة عن استخدام الـAgile development في عملية الـMaintaining الصيانة للمنتجات البرمجية حيث لاينصح (لكن يمكن) استخدام الـAgile Methods في عملية الصيانة فإذا كانت هذه المنتجات الأخيرة قد تم تطويرها باستخدام الـAgile سنحتاج لعمليات هندسة عكسية **Reverse Engineering** حتى نستطيع استنباط الوظائف والمتطلبات التي تقوم بها هذه المنتجات لايجاد العطل وإمكانية التغيير فيه, أما اذا كانت قد طورت باستخدام الطرق الكلاسيكية فستقوم بهدم البنية **Structure** الخاصة بهذه المنتجات من **Documentations, Models...Etc.** وستتحول لمنتجات غير قابلة للصيانة فيما بعد.

Requirements Engineering

بداية ماهو تعريف المتطلبات **Requirements** ؟

هي مجموعة الوظائف والخدمات التي يقدمها النظام.

من يقوم بتحديد هذه المتطلبات ؟

الزبون **Customer**, مدير الشركة التي تحتاج النظام البرمجي **Managers** أو المستخدمون **End Users** (الموظفون في الشركة مثلاً الذي سيستخدمون البرنامج فعلياً), سنطلق عليهم **Stakeholders**...

ومهمتنا كـمهندسي برمجيات هي أيجاد الطرق والوسائل لجمع الوظائف والمعلومات التي من المتوقع أن ينفذها المنتج البرمجي وهذه العملية يطلق عليها Requirements Engineering وهي:

The broad spectrum of tasks and techniques that lead to an understanding of requirements is called *Requirements Engineering*. ***SE practitioner approach P120***

تأتي أهمية المتطلبات من أن أي سوء فهم أو تواصل في هذه المرحلة سيؤدي إلى نتائج سلبية ومن الممكن أن تؤدي إلى فشل المنتج البرمجي الذي بشكل عام يعتبر مكلف مادياً...

"The seeds of major software disasters are usually sown in the first three months of commencing the software project". ***Caper Jones***

سنكمل في هذه المحاضرة مع أنواع المتطلبات ثم مع الطريقة السليمة لربط المتطلبات وتنسيقها في وثائق تكون مرجعية للإنطلاق بالمرحلة القادمة في عملية تطوير المنتج البرمجي...

أنواع المتطلبات Requirements Types:

بدايةً إن عملية **تجميع وفهم وتثبيت** المتطلبات تتم في مرحلة الـ**Analysis** ولكن يمكن أن يكون البدء في تجميعها **Requirements Gathering** أو **Information Gathering** قد بدء من مرحلة الـ**Communication** لكن في مرحلة الـ**Analysis** يتم أخذ المعلومات المهمة والمفيدة في المشروع وتثبيتها وتوثيقها بحيث لا نستخدم المعلومات غير المهمة والتي يمكن أن تؤدي إلى فشل النظام لاحقاً وقد ينتقل أيضاً مجموعة من نشاطات هذه العملية إلى مرحلة الـ **Planning**.

وهنا ظهر ما يسمى **Feasibility Study** أي دراسة إمكانية نجاح المشروع وحسب نتيجة هذه الدراسة يتم أخذ القرار بإتمام المشروع من عدمه وذلك حتى لا يتم إضاعة الوقت.

ويمكن أن تندرج المتطلبات تحت الأصناف التالية:

- **User Requirements:**

تمثل الوظائف التي تم فهمها من الزبون أو المستخدم ونقوم بإعادة كتابتها وصياغتها بطريقة يفهمها المستخدم ولانستخدم فيها مصطلحات تقنية أو غير مفهومة بالنسبة لهم ونستفيد منها في التأكد من أننا قد فهمنا الوظائف التي طلبها المستخدم بشكل سليم.

> **User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate.** ***SE by Sommerville P83 | our course, slide 3***

- **System Requirement:**

وهي مستنتجة من الـUser requirement حسب فهم المحلل Analyst للمتطلبات مع تفصيل أكثر ووضع الخطوط العريضة للحل لتكون مرجع يمكن الرجوع لها عند الحاجة .

> **System requirements are more detailed descriptions of the software system's functions, services, and operational constraints. The system requirements document (sometimes called a functional specification) should define exactly what is to be implemented. It may be part of the contract between the system buyer and the software developers.**
>
> ***SE by Sommerville P83 | our course, slide 3***

مثال مطروح في السلايدات يمكن استخدامه لفهم متطلبات المستخدمين ومتطلبات النظام بشكل أفضل:

نلاحظ هنا أن الـuser requirement تكتب بشكل تجريدي بحيث أي شخص يقرأها يستطيع أن يفهمها بينما الـsystem requirement تأتي لتفصيل وشرح ما كتب في الـuser requirement وموجهة للأشخاص التقنيين بشكل أكبر...

User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

Domain Requirement:

هي المتطلبات والقيود المشتقة من مجال عمل هذا النظام وليست مشتقة من متطلبات المستخدمين فعلى سبيل المثال في نظام برمجي طبي معين يوجد متطلبات (من وزارة الصحة مثلاً) على النظام تحققيها:

Insulin Pump System that delivers insulin on demand include the following domain requirement:

- **The System safety should be assured according to standard IEC 60601-1: Medical Electrical Equipment – Part 1 General requirement for Basic Safety and Essential Performance.**

http://www.SoftwareEngineering-9.com/Web/Requirements/DomainReq.html

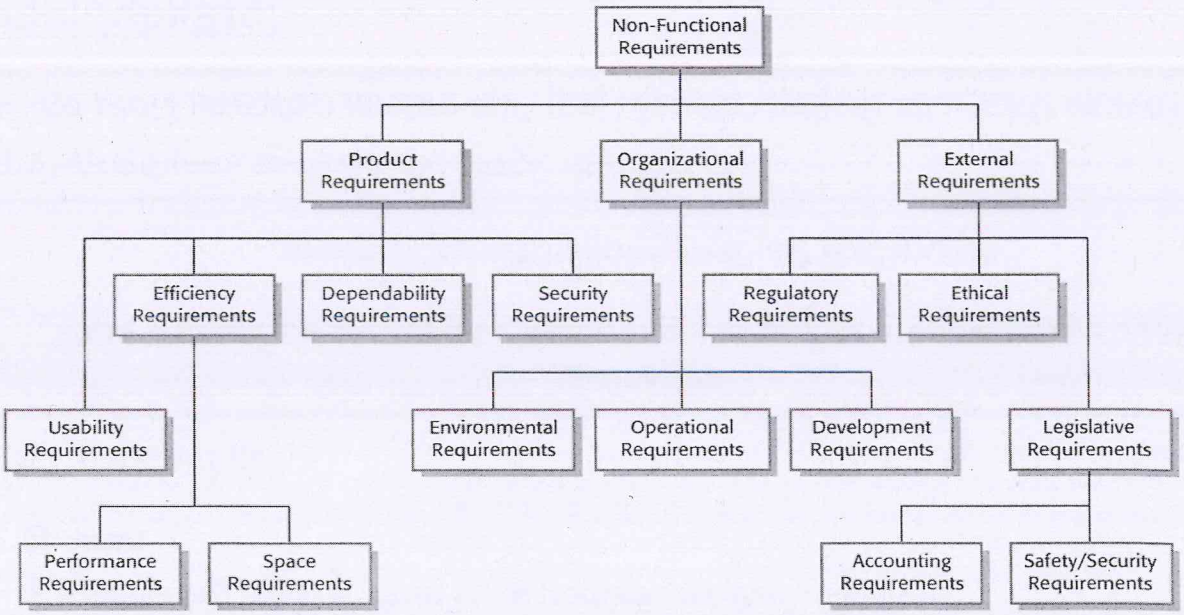
Functional Requirements:

هي مجموعة المتطلبات والوظائف والـ Actions التي يقدمها النظام وهي تعتمد على طبيعة هذا النظام والمستخدمين المتوقعين لاستخدامه وعدم الدقة فيها يمكن أن يخلق بعض المشاكل كما في اختلاف تفسير المتطلبات بين المستخدم والمطورين.

هذه المتطلبات يجب أن تكون **كاملة Complete** أي يجب أن تصف كل متطلبات النظام وأن تكون **متناسقة Consistent** أي أنها يجب أن تكون بعيدة عن التضارب مع بعضها.

Non-functional Requirements:

هي المتطلبات التي لا تتصل مباشرةً بالخدمات أو الوظائف التي يقدمها النظام وعلى الأغلب لن يطلبها المستخدم حرفياً أي يجب أن تتحقق ضمن المشروع بشكل بديهي مثل التقيّد بالوقت المحدد والأمان واستخدام معايير محددة, وغالباً ما يتم تطبيقها على **كامل** النظام وليس على جزء أو خدمة أو ميزة منه, وفي حال عدم تطبيق أحد هذه المتطلبات يمكن أن يجعل النظام عديم الفائدة. فعلى سبيل المثال الفشل في نظام قيادة الطيار الآلي لتحقيقه للمتطلبات الاعتمادية **Reliability Requirements** سيجعل منه نظام غير آمن ولن يتم استخدامه.



ويمكن أن تصنف الـ **Non-Functional Requirements** إلى 3 أنواع وهي:

❖ Product Requirements:

هي الصفات التي تحدد أو تقيد تصرف المنتج البرمجي ويجب أن يتمتع بها الـ **Final Product** مثل: **Execution Speed, Reliability, Security Requirements , Usability Requirements**

❖ Organizational Requirements:

هي المتطلبات الناتجة عن اتباع السياسات والإجراءات التنظيمية لكلا المؤسستين (الزبون وفريق التطوير) مثل :

Developing Environment, Programming Language, Process Standards ...etc.

❖ External Requirements:

هي المتطلبات المشتقة من عوامل خارجية عن النظام البرمجي وعمليته التطويرية مثل:

- **Interoperability Requirements** المتطلبات التوافقية مع أنظمة برمجية أخرى يتم استخدامها (كـنظام بنكي).
- **Legislative Requirements** المتطلبات التشريعية التي تضمن تنفيذ النظام البرمجي ضمن القوانين.
- **Ethical Requirements** متطلبات أخلاقية تضمن أن يتم قبول النظام من قبل مستخدميه.

Stakeholders:

كي يتم تجميع المتطلبات السابقة على أكمل وجه يجب التواصل مع أشخاص محددين والذين نطلق عليهم اسم **Stakeholders** ويمكن تعريفهم بـ :

الأشخاص المعنيين والمستفيدين من نجاح النظام.

Anyone who benefits in a direct or indirect way from the system which is being developed.

ويمكن تصنيفهم إلى:

- ☑ **Users**
- ☑ **Developers يمكن أن يشمل الـ designer – analyst - supporter**
- ☑ **System Administrators**
- ☑ **Testers**
- ☑ **Support Staff**
- ☑ **Maintainers**
- ☑ **System managers**
- ☑ **System owners**

Requirements Engineering Process:

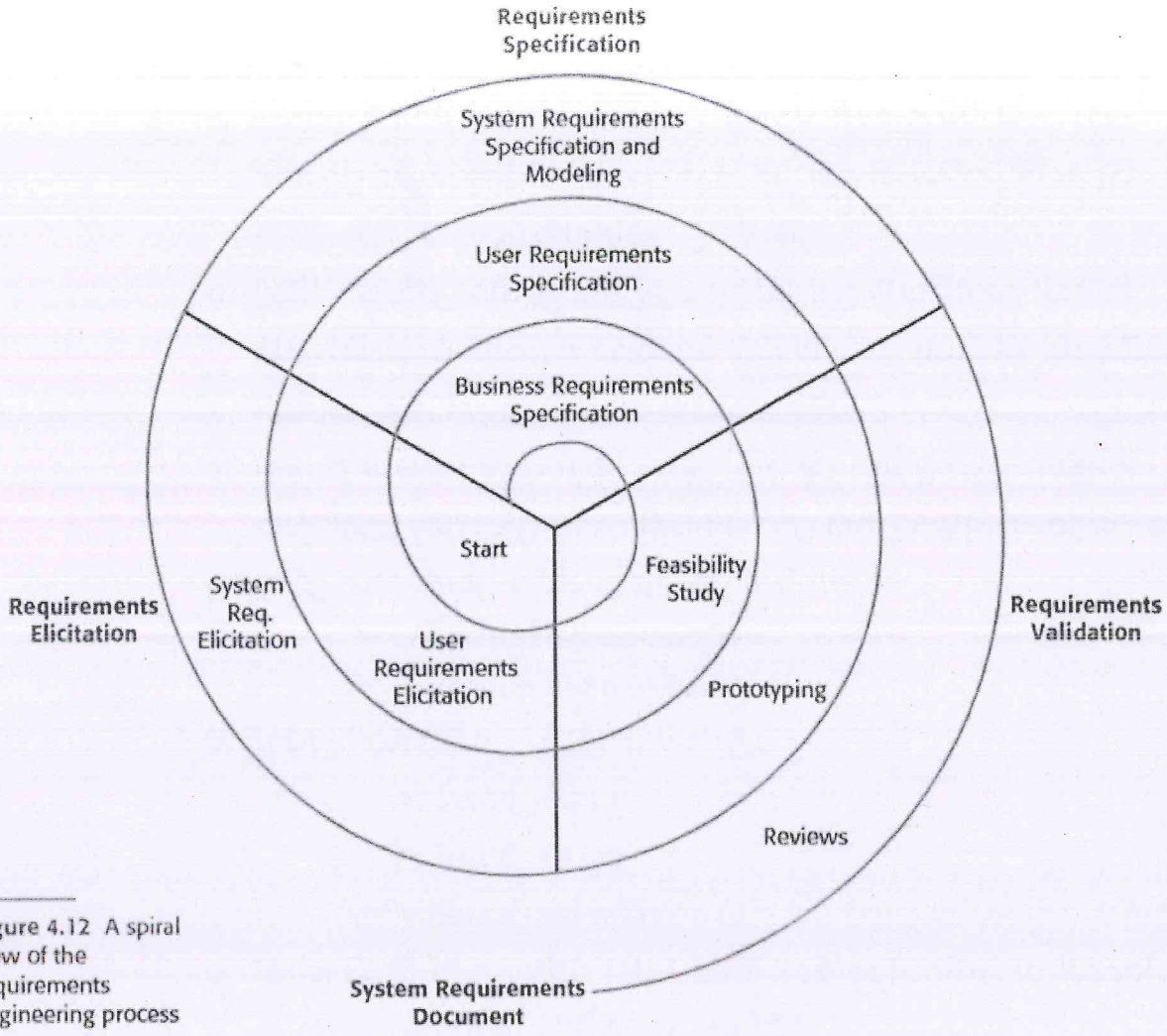


Figure 4.12 A spiral view of the requirements engineering process

كما قلنا سابقاً إن عملية تجميع وفهم المتطلبات هي عملية صعبة وأن أي خطأ في فهم المتطلبات سيؤدي إلى فشل النظام ولذلك يوجد عدد من العمليات العامّة التي يجب القيام بها لضمان عملية تجميع وفهم المتطلبات الأساسية للنظام وهي:

- **Requirements Elicitation:**

هي المنحى المبدئي/البنائي وتتألف من الـ **work tasks** التالية:

1. **Requirements discovery:**

هي المرحلة التي يتم فيها عملية تجميع المتطلبات وبما أن أساس هذه العملية هو **التجميع Gathering** لذلك هنالك عدة أنواع للتجميع منها:

← Interviews المقابلات:

عبارة عن مجموعة من المقابلات التي تتم لأشخاص محددين بغاية تجميع المعلومات الخاصة عن موضوع معين.

- **خطوات الـInterview :**

a. التحضير للمقابلة بشكل أساسي وتحديد المعلومات التي يجب الحصول عليها وبالتالي يجب تحديد من هم الأشخاص الذين يجب أن أحصل على هذه المعلومات من خلالهم.

b. إعطاء الشخص ملخص كامل وسريع عن نوع الأسئلة مع الحفاظ على أخلاقيات العمل و خصوصية هذا الشخص بحيث إذا أراد إنهاء المقابلة في أي وقت يريده لا نمنعه ذلك.

c. تصميم الأسئلة بحيث أحصل على أكبر قدر ممكن من المعلومات خلال أقصر زمن, وأن يبقى هذا الشخص ضمن سياق المقابلة كما يمكن ملاحظة أن هناك نوعين من الأسئلة:

- ✓ **Open-ended questions**
- ✓ **Closed-ended questions**

d. توثيق المقابلة عن طريق عدة طرق:

- ✓ تسجيل صوت
- ✓ تسجيل فيديو
- ✓ تسجيل ملاحظات

e. تقييم المقابلة من خلال إعادة كتابة المعلومات التي تم الحصول عليها ثم مراجعتها والتأكد من صحتها.

- **مساوئ الـInterview :**

i. الشخص الذي نجري معه المقابلة يكن أن يستخدم مصطلحات ومفردات من الممكن أن تكون غير مفهومة لنا.

ii. الكلفة الكبيرة للوقت المستغرق في المقابلات.

← Surveys الاستبيانات:

مجموعة من الأسئلة يتم توزيعها على مجموعة من الأشخاص ثم يتم تحليل النتائج والحصول على المعلومات من خلال تحليل هذه النتائج وبالتالي يتم التخلص من مشكلة إضاعة الوقت في **Interviews.**

- مساوئ الـSurveys :

عدم الحصول على نتائج صحيحة 100% وذلك بسبب عدم شعور الشخص الذي يجري الاستبيان بمسؤوليته عن المعلومة التي يعطيها.

← Observation المراقبة:

المراقبة شئ مزعج للبعض وبالتالي عندما يشعر أي شخص أنه تحت المراقبة سوف يقوم بأفعال لم يعتد على القيام بها وهذا يؤدي إلى نتائج غير صحيحة وتستخدم عادة في الـprototype لأنها تستخدم عندما يكون المستخدم ليس لديه قدرة على التعبير.

2. **Requirements classification and organization:**

تصنيف المعلومات التي تم جمعها وتنظيمها بشكل متماسك في إطار معين يخدم عملية تطوير النظام.

3. **Requirements prioritization and negotiation:**

ترتيب الأولويات بين المهمات التي يجب تنفيذها بحيث يضمن عدم حدوث تضارب أو تقاطع بينها ويتم تحديد هذه الأولويات بين المستخدم و المطورين.

4. **Requirements specification:**

يتم توثيق المتطلبات التي تم استنباطها من الـ **User Requirements** بشكل مبدئي **initial** وهذه الوثائق ليست النهائية فهي تختلف عن وثائق وتوصيف المرحلة القادمة التي ستكون أكثر تفصيلاً.

- **Requirements Analysis (Specification):**

يتم وضع توثيق رسمي للمتطلبات يطلق عليه **System Requirements Specification SRS** حيث من الممكن أن يحوي مجموعة من النماذج التصويرية **Graphical Models** أو النماذج الرياضية **Mathematical Models** أو حالات الاستخدام **Usage Scenarios** أو حتى نموذج أولي...

- **Requirements validation:**

يتم في هذه المرحلة مراجعة الـ **Specification** الخاص بالمتطلبات من قبل فريق من الـ**Reviewers** (يتألف من مجموعة من المهندسين, الزبائن أوالـ**stakeholders**) ليضمنوا من أن جميع المتطلبات قد تم توصيفها وحل جميع التناقدات والغموض فيها.

- **Requirements management:**

مجموعة من النشاطات التي تساعد فريق التطوير في التحكم و متابعة عملية **Requirement Engineering** في أي مرحلة من المراحل السابقة بطريقة صحيحة.

وهنا تكون قد انتهت محاضرتنا لنكمل في المحاضرة القادمة مراحل Requirements Engineering Process بشكل أكثر تفصيلاً (حسب السلايدات)...

-النهاية-