# إختبار البرمجيات

- ثلاثة مجموعات وسائل لتوفير الإعتمادية:
1. تجنب العيوب.
2. نزع العيوب.
3. تحمل العيوب.

حيث تنفيذ البرمجيات في ظروف مراقبة على بيانات واردة ثم التأمل في ردود فعل البرمجية و تحليلها.

- يلعب اختبار البرمجيات دورا هاما في الدورة الحياتية.
- يلعب دورا هاما في سياسية المؤسسة المتعلقة بالجودة.
- يحتل مكانة هامة في ميزانية إنتاج البرمجيات.

- أنواع الاختبار:
- أربعة أنواع عامة:

1. اختبار الوحدات: هل وحدة ما تقوم بوظيفته حسب ما جاء في المواصفة؟
2. اختبار الإدماج: هل أطراف النظام تتعامل معا لتؤدي وظيفته؟
3. اختبار القبول / التسليم / الشهادة .هل النظام جاهز للتسليم؟ هل المستخدم مستعد لقبوله؟
4. اختبار الضمانة: هل النظام يلبي مواصفات الضمانية؟

- سبعة أبعاد لتعيين أنواع الاختبار:

1. الهدف.
2. المفترض.
3. المعيار.
4. الحكيم.
5. البيانات اللازمة.
6. إختيار البيانات.
7. المتدخلون / الأطراف المعنية.

- مراحل الاختبار:

1. تعيين بيانات الإختبار.
2. تعيين و تحهيز الحكيم.
3. تنفيذ الإختبار، تسجيل نتائجه.
4. تحليل نتائج الاختبار، اتخاذ الإجراءات المطلوبة.

- تعيين بيانات الإختبار:

1. الأساليب المبنية على البرامج
2. الأساليب المبنية على المواصفات.

- الأساليب المبنية على المواصفات:

1. تغطية جميع الخدمات.
2. تغطية حدود ميدان المواصفة.
3. تغطية مناطق مجاورة لميدان المواصفة.
4. اختيار بيانات متناقضة.
5. اختيار بيانات تمسح مجال كل متغيرة هامة من الميدان.
6. بيانات عشوائية.

❖ **الأساليب المبنية على البرامج:**

1. **الأساليب الهيكلية.**
2. **الأساليب الوظيفية.**

❖ **نكون بيانات الاختبار انطلاقا من:**

1. **نص /هيكل البرنامج.**
2. **وضيفة البرنامج.**
3. **مواصفات البرنامج.**

❖ **الأساليب الهيكلية:**

1. **تنفيذ جميع تعليمات البرنامج.**
2. **تنفيذ جميع شروط البرنامج**
3. **تنفيذ جميع مسارات البرنامج.**
4. **تنفيذ جميع مسارات البرنامج، مع تحديد عدد تنفيذ المدارات.**

❖ **تحديد و إنجاز الحكيم:**

**ضبط الحكيم يتغير حسب نوع الاختبار و حسب ظروفه:**

1. **اختبار الوحدات، كشف العيوب**
2. **اختبار الوحدات، قبول البرنامج**
3. **اختبار الإدماج**
4. **اختبار القبول**
5. **اختبار الضمانية**
6. **اختبار السلامة**

❖ **تحليل نتائج الاختبار:**

1. **تحليل الملف التي سُجلت فيه نتائج الإختبار.**
2. **القيام بالتصليحات اللازمة عند الحاجة.**

# إدارة تكوين البرمجيات

**تعرف إدارة التكوين بأنها فن التنسيق بين جهود تطوير البرمجيات لتقليل الإرباك**

**فائدة إدارة التكوين :زيادة الإنتاجية إلى حدها الأقصى وتقليل الأخطاء إلى حدها الأدنى**

- **يتم تطوير إدارة تكوين البرمجيات:**
  1. **تعيين هوية التغيير**
  2. **التحكم بالتغيير**
  3. **التحقق من أن التغيير قد نفذ بشكل صحيح**
  4. **إعلام الآخرين المهتمين بحدوث هذا التغيير**
- **الفرق بين الصيانة وادارة تكوين البرمجيات:**

**الصيانة :هي مجموعة من نشاطات هندسة البرمجيات التي تحدث بعد تسليم المنتج للزبون**
**( إدارة تكوين البرمجية) : هي مجموعة من نشاطات المتابعة والتحكم التي تبدأ عندما يبدأ المشروع البرمجي وتنتهي فقط عند التوقف عن استعمال البرمجية**

- **تابع إدارة تكوين البرمجياتSCM**

**يمكن تقسيم مخرجات عملية البرمجة إلى ثلاث فئات:**

1. **برامج كمبيوتر**
2. **وثائق تصف البرامج**
3. **بيانات**

- **يمكننا ايجاز مصادر التغيير في أربع مصادر أساسية هي:**
  1. **ظروف عمل جديدة أو سوق جديدة**
  2. **احتياجات جديدة للزبون**
  3. **إعادة التنظيم أو تخفيض حجم الأعمال**
  4. **تقييدات عائدة للموازنة او الجدول الزمني تسبب إعادة تعريف النظام أو المنتج**
- **تعريف الركيزة baseline هي مفهوم إدارة تكوين البرمجيات والتي تساعدنا على ضبط التغيير دون إعاقة كبيرة للتغييرات المبررة**
- **تعريف الركيزة وفق IEEE هي مواصفات أو منتج تمت مراجعته والموافقة عليه رسميا وتلعب بعد ذلك دور الأساس للتطويرات التالية**
- **الركيزة في سياق هندسة البرمجيات :هي معلم على طريق تطوير البرمجية لا يمكن تنفيذ تغيير على الركيزة إلا بعد اجراء رسمي لتقييم التغيير**
- **بنود تكوين البرمجيات:**

**بند تكوين البرمجية SCI هو وثيقة أو مجموعة كاملة من الاختبارات أو مكون مسمى من برنامج مثل دالةFunction**

- **يجب أن توضع كلا من الأدوات البرمجية وبنود تكوين البرمجيات تحت المراقبة وذلك لأن الأدوات تعتبر ركيزة**
- **لكل كائن من كائنات التكوين اسم وسمات ويربط بكائنات أخرى عن طريق علاقات**
- **السهم المنحني بين الكائنات يشير إلى علاقة مركبة**

- **عملية إدارة تكوين البرمجيات:**

**إدارة تكوين البرمجيات :هي عنصر هام لضمان جودة البرمجية**
**مهام إدارة تكوين البرمجيات ومسؤولياتها تتلخص في:**

1. **تعيين الهوية**
2. **التحكم بالإصدارات**
3. **التحكم بالتغييرات**
4. **تدقيق التكوين**
5. **الإعلام به**

❖ **تعيين هوية الكائنات في تكوين البرمجيات:**

**يجب أن يسمى كل بند على حدي لضبطه وإدارته**
**نستطيع تعيين هوية نوعين من الكائنات:**

1. **الكائنات الأساسية :هو وحدة نصية ينشئها مهندس البرمجية خلال احدى مراحل هندسة البرمجية**
2. **الكائنات الإجمالية: مجموعة كائنات أساسية وإجمالية أخرى وهو قائمة من المؤشرات التي تحدد كائنات أساسية**

❖ **سمات الكائنات البرمجية:**

- **اسم :هو سلسلة أحرف تعين هويته بشكل متميز**
- **وصف: هو قائمة بنود بيانات تميز:**

1. **نوع البند الذي يمثله الكائن**
2. **هوية المشروع ومعلومات التغيير والإصدار**

- **قائمة موارد: هي كيانات تقدم أو تعالج وتعنون أو تطلب من قبل كائن**

➢ **نأخذ بعين الاعتبار العلاقات الموجودة بين الكائنات المسماة**
➢ **يمكن تمثيل العلاقات بين الكائنات باستخدام لغة الوصل البيني للوحدات النمطية MIL**

❖ **التحكم بالإصدارات:**

**يضم التحكم بالإصدارات إجراءات وأدوات لإدارة إصدارات مختلفة لكائنات التكوين**
**لتوضيح العلاقة بين المكونات والبدائل والاصدارات يمكن أن نمثلها بمجمع كائناتObject pool**

❖ **التحكم بالتغييرات:**

**تؤدي التغييرات غير المتحكم بها إلى الفوضى**
**يجمع التحكم بالتغييرات بين الإجراءات البشرية والأدوات المؤتمتة لتوفير آلية للتحكم بالتغييرات**

❖ **تدقيق التكوين:**

**لكي نضمن من أن التغيير قد نفذ بطريقة مناسبة علينا استخدام كلا من:**
**1.المراجعات التقنية الرسمية 2 .تدقيق تكوين البرمجية**
**تدقيق تكوين البرمجية هو المكمل للمراجعات التقنية الرسمية فهو يقوم بتقدير كائن التكوين من حيث الخصائص التي لا تدرس خلال المراجعة**

❖ **تقرير الحالة:**

**يجيب على الأسئلة التالية:**
**ماذا حدث؟ من فعل ذلك؟ متى حدث ذلك؟ ماذا سيتأثر أيضا؟**
**يمكن وضع الخرج الصادر عن تقرير حالة التكوين CRS في قاعدة بيانات على الخط**
**يولد التقرير بانتظام والهدف منه ابقاء الإدارة والممارسين على علم بالتغييرات الهامة**
**تقرير حالة التكوين هو عامل هام لنجاح المشروع لأنه يساعد في تجنب مشاكل تعارض التغيير**

## التصميم المعماري

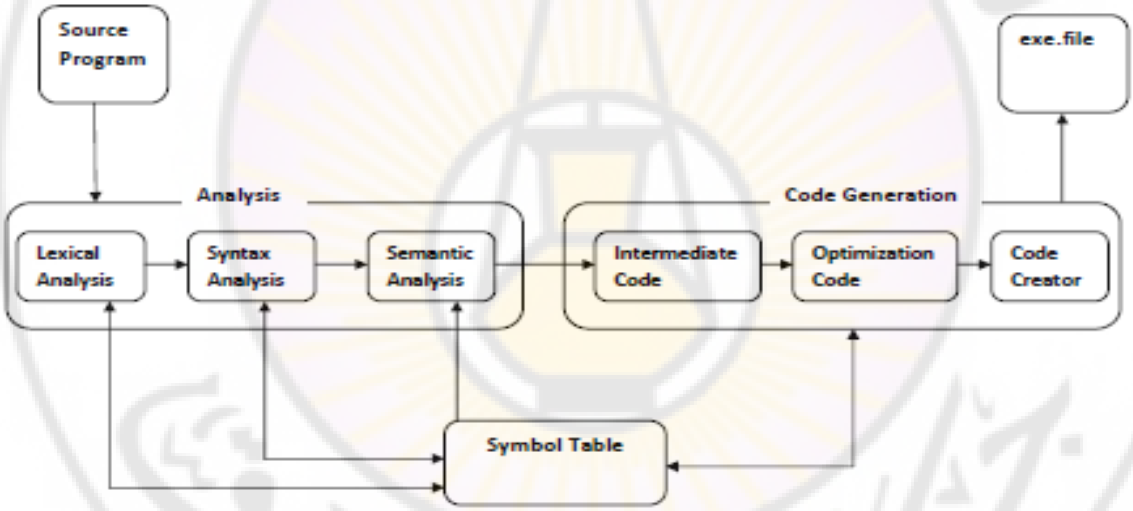
هو اسلوب بناء النظام وكيف سيتم التحكم بالانظمة الفرعية وما هي طريقة التواصل بين تلك الانظمة وطريقة تمرير ومعالجة البيانات فيما بينها، وبتم ذلك وفق العناوين الثالثة الرئيسية التالية:

1. طرق هيكلة الأنظمة.
2. طرق التحكم بالانظمة.
3. طرق الاتصال.

❖ طرق هيكلة الانظمة: طريقة بناء النظام بحيث نستغل موارد النظام بشكل أكبر وامكانية معالجة الاخطاء، ونحدد الطريقة من خلال رسم صندوقي يصف طريقة بناءنا للنظام.

➢ ولدينا ثالث طرق لبناء الانظمة هي كالتالي:

1. Repository Model: يوجد شيئ مشترك بين كل الدوال او اجراءات النظام كأن يكون جداول او ان يكون اجراء معين فيه شرط معين مثل اجراء الترقيم التلقائي للتقارير مثال لهذه الطريقة هي المترجمات.




1- Source Program :وهو البرنامج المكتوب من المستخدم بغض النظر عن اللغة المكتوب بها.

2-Analysis مرحلة التحليل وفيها:

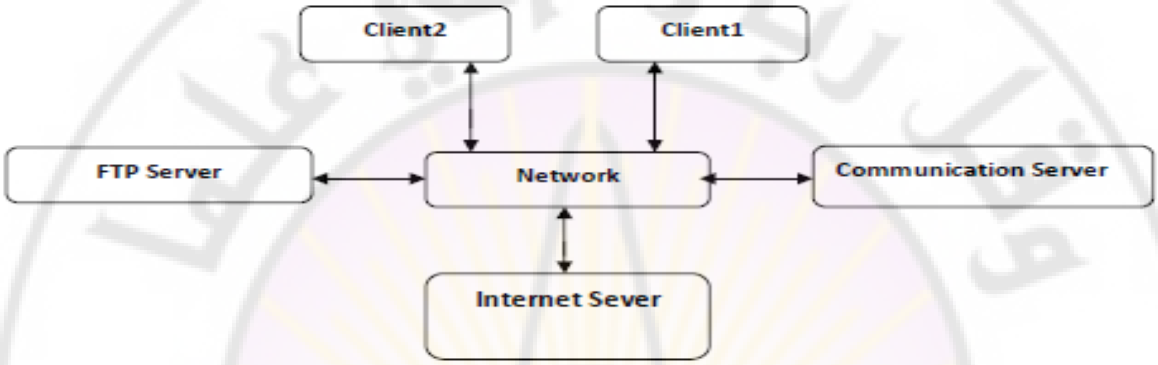
- Lexical Analysis : تحليل البرنامج المكتوب من قبلالمستخدم وتحديد الكلمات المحجوزة للغة مثل if و for وغيره.
- Syntax Analysis: مقارنة مفردات البرنامج مع الكلمات المكتوبة في الجدول Symbol Tableمن حيث الاخطاء المحتملة عند كتابة الكود.
- Semantic Analysis :تحليل المعاني أي ماذا تعني كلمة If او ماذا تعني كلمةfor وكل ذلك يتم بالمقارنة مع الجدول الوسيط Symbol Table .

3- Code Generation :مرحلة بناء الكود أي ان يتم في هذه المرحلة انشاء الكود البرمجي الخاص بالالة وغيره وفيها:

- Intermediate Code:تحويل واعادة بناء الكود الذي قام بكتابة المبرمج الى كود وسيط وهو كود ما بين لغة الالة وكود بلغة الانسان.

- **Optimization Code:تحسين الشفرة أي الغاء ماليس كان يتم الغاء التعليقات اوالفواصل المنقوطة.**
- **Code Creator:بناء الكود البرمجي من الشفرة العادية الى شفرة بلغة فرعية من لغة الأوامر.**

2. **Model Server/Client : وفي هذه الطريقة نقوم بإنشاء برنامج Client أي مستفيد ومرتبط ببرنامج Serve r وفق شبكة Server خادم لتلك الأجهزة المستفيدة وكذلك يقوم بعملية التحكم والمراقبة لا نعتمد على بيانات مشتركة المثال تقنية الانترنت**

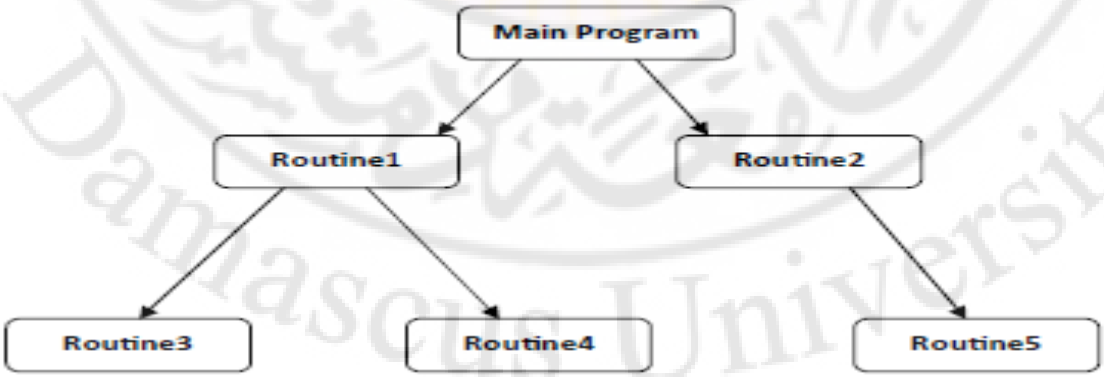



3. **Abstract Machine Model : تعتمد هذه الطريقة على بناء الانظمة على شكل طبقات وكل طبقة من الطبقات تعتمد على التي تحتها فعلى سبيل المثال شاشات الMDI .**

❖ **طرق التحكم بالانظمة: نحدد كيف سنقوم بالتحكم بالنظام أي ماهي استراتيجية السيطرة على الانظمة الفرعية او الاجراءات اوغيره.**

➢ **ولدينا نوع من التحكم: التحكم المركزي: أي ان يتم التحكم بالنظام ككل من خلال برنامج رئيسي او دالة رئيسية ومن انواعها:**

1. **Manager: وفي هذه الطريقة يتم التحكم بالانظمة بشكل مركزي بحيث يقوم البرنامج الرئيسي بالتحكم بكل البرامج.**




2. **Real time: في هذه الطريقة فان التحكم يكون على اساس الاحداث فعند انطلاق الحدث يتم تنفيذ عمل معين.**

3. **Procedural:** أي يتم تنفيذ اجراءات بمجرد انطلاق الحدث ولا تعتمد على الوقت الحقيقي بل عند الضغط.

❖ **طرق الاتصال:** استراتيجية الاتصال بين مكونات النظام ولها طريقتين هما:

1. (OOP): أي تتم عملية الاتصال على اساس البرمجة الكائنية او الشيئية.
2. تدفق البيانات Data Flow Model .

# تخطيط المشاريع البرمجية

المدير هو المسئول عن تحديد المشكلة واختيار النموذج المناسب لحل المشكلة وعمل الخطة التمهيدية للبدء بالعمل

- ماذا نحتاج لنستطيع القيام بتقدير جيد وفعال؟

1. الخبرة
2. توفر المعلومات التاريخية الجيدة
3. تقبل المجازفة
4. العلم بأن المخاطرة موجودة

- كيف نقيس المخاطرة وماهي العوامل المؤثرة على المخاطرة والشك في دقة التقديرات؟

بقياس درجة الشك في التقديرات والكلفة والجدول الزمني وتزيد المخاطرة بعدم وجود أفق مشروع واضح أو متطلبات متغيرة والعوامل:

1. تعقيد المشروع :يبقى نسبياً
2. حجم المشروع :بازدياد الحجم تزداد قوة الارتباط بين العناصر
3. درجة الشك البنيوي :ونعني بها مستوى صرامة المتطلبات
4. المعلومات التاريخية :بتوفر المعلومات التاريخية نستطيع تجنب المشاكل

- ما لهدف من التخطيط لأي مشروع؟

توفير الأساس الذي يسمح للمدير بإجراء تقديرات معقولة بالنسبة للموارد والكلفة والزمن

- الموارد:

1. الأشخاص
2. المكونات البرمجية القابلة لإعادة الاستخدام
3. الأدوات العتادية والبرمجية

- مميزات الموارد:

1. وصف المورد
2. مدة الحاجة إلية
3. لحظة الحاجة إليه
4. توفر المورد

- ببيئة هندسة البرمجيات : المحيط الذي يوفر الدعم للمشروع البرمجي ويضم البرمجيات والعتاد
- العتاد : هو المنصة اللازمة لاستخدام الأدوات البرمجية
- خيارات الحصول على تقدير موثوق للجهد والكلفة اللازمين لإنجاز البرمجية:

1. تأجيل عملية التقدير لمرحلة متأخرة
2. الارتكاز على مشاريع أخرى مشابهه سبق أن أنجزت
3. استخدام تقنيات التفكيك
4. استخدام نموذج تجريبي أوأكثر

- التقدير المعتمد على المشكلة: يمكن استخدام بيانات LOC و FP خلال عملية التقدير بطريقتين:

1. متغير التقدير: ويستخدم لتحديد حجم كل عنصر من عناصر البرمجية
2. مترية الركيزة:متراكمة من مشاريع سابقة ويمكن استخدامها مع متغيرات التقدير لتطوير توقعات الكلفة والجهد

- خطوات التقدير المعتمد على المشكلة:

1. تحديد أفق البرمجية
2. تفكيك البرمجية إلى وظائف المشكلة لتقدير كل منها على حدى
3. يقدر LOC و FP لكل وظيفة
4. تطبق مترية الركيزة للإنتاج loc/pm و fp/pm على متغير التقدير المناسب
5. تشتق الكلفة والجهد للوظيفة
6. ضم تقديرات الوظيفة للحصول على تقدير اجمالي يشمل كامل المشروع

- أنواع التقدير:

1. تقدير معتمد على أسطر الشيفرةLOC
2. تقدير معتمد على نقاط الوظيفةFP

3. التقدير المعتمد على عملية البرمجة

❖ تختلف LOC عن FP من حيث مستوى التفصيل المطلوب للتفكيك والهدف من التجزئة:

➢ تقديرLOC :

1. التفكيك أمر جوهري
2. نحتاج لعدد كبير من مستويات التفصيل
3. كلما كانت درجة التجزئة عالية يزيد احتمال الحصول على تقديرات دقيقة

➢ تقديرFP :

1. لا يركز على الوظيفة
2. التقدير يكون لمميزات نطاق المعلومات
3. نستخدم تقديرات المميزات للحصول على قيمة FP لتوليد التقدير

❖ التقدير المعتمد على عملية البرمجة:

هو أكثر تقنيات التقدير انتشارا وهو التقدير الذي نقوم من خلاله بتفكيك عملية البرمجة إلى مجموعة صغيرة نسبيا من النشاطات أو المهام وبعدها يقدر الجهد اللازم لإنجاز كل منها

❖ البدائل المتوفرة لاتخاذ قرار التطوير أو الشراء هي:

1. شراء البرمجيات الجاهزة أو الحصول على رخصة استخدامها
2. اقتناء بعض المكونات البرمجية ذات الخبرة الشاملة أو الخبرة الجزئية
3. تفصيل البرمجية الخاصة بنا لدى متعاقد خارجي

❖ شروط اتخاذ قرار التطوير أو الشراء:

1. هل سيكون وقت تسليم المنتج أبكر من حالة التطوير الداخلي؟
2. هل ستكون كلفة اقتناء البرمجيات وتعديلها أرخص من التطوير الداخلي؟
3. هل ستكون كلفة الدعم الفني الخارجي والصيانة أرخص من الدعم الفني الداخلي؟

❖ أدوات التقدير المؤتمتة:

ومن أنواعها تقنيات التفكيك ونماذج التقدير التجريبية

➢ الهدف منها :السماح للمخطط بتقدير الكلفة والجهد والقيام بتحليل للمتغيرات الهامة للمشروع

➢ خصائص أدوات التقدير المؤتمتة:

1. تقدير كمي
2. وصف كيفي
3. بعض الأوصاف لبيئة التطوير

# جدولة المشرع

❖ **الأدوات الرئيسية الثلاثة في إدارة المشروع:**

**خطة المشروع وجدولة المشروع و موازنة المشروع**

❖ **تجزئة هيكل العمل: تجزئة البرنامج إلى مشاريع والمشروع إلى مهمات والمهمة إلى حزم عمل وحزم العمل إلى وحدات عمل ووحدة العمل إلى أنشطة**

❖ **تعريف مبسط لجدولة المشروع :هي عملية تحويل خطة المشروع إلى جدول زمني لتشغيل المشروع ابتداء من لحظة مباشرة العمل في المشروع مرور ا بجميع الأنشطة المتتابعة والمتداخلة والأحداث والمحطات الرئيسية ووصولا إلى لحظة انتهاء العمل في المشروع**

❖ **منافع جدولة المشروع ومنها:**

1. **إطارا منسقا للتخطيط وتوجيه ومراقبة المشروع.**
2. **تبين حالة الاعتمادية والتداخل لكافة الأنشطة.**
3. **تشير إلى الوقت الذي يحتاج فيه المشروع إلى تواجد بعض الخبرات والمهارات الخاصة**
4. **تساعد في توفير خطوط اتصال أوضح وأقصر بين الأقسام**
5. **تساعد في تحديد التاريخ المتوقع لإنهاء المشروع**
6. **تساعد في تحديد الأنشطة**
7. **تساعد في تحديد الأنشطة الراكدة**
8. **تساعد في تحديد تواريخ بداية ونهاية الأنشطة وعلاقة الأنشطة بالأنشطة الأخرى.**
9. **تساعد في تخفيف الخلافات الشخصية وتقليل من الصراعات على الموارد.**

❖ **مراحل جدولة المشروع:**

1. **مرحلة التخطيط :وتتضمن تحليل أنشطة المشروع إلى وحدات بحيث تكون كل وحدة مكونة من مجموعة من الأنشطة المتشابهة في العمل والحجم**
2. **مرحلة جدولة الأنشطة :تكون من تحديد الوقت اللازم لإنجاز كل نشاط من أنشطة المشروع ثم تقدير التكاليف اللازمة لإنجاز كل نشاط من هذه الأنشطة.**
3. **مرحلة الرقابة :وفي هذه المرحلة يتم التحقيق فيما إذا كان العمل قد تم تنفيذه وفق ما خطط له أم أنه قد حدثت انحرافات.**

❖ **طرق جدولة المشروع:**

1. **<u>خرائط جانت:</u>**

**هي إحدى أقدم الطرق المستخدمة في جدولة الأنشطة وقد تم تطويرها على يد هنري جانت عام 1917 م وهي طريقة بسيطة، سهلة الإعداد، سهلة القرأة وفعالة خاصة في تحديد مدى التقدم وتنفيذ الأنشطة ومراقبة الزمن وتتكون خرائط جانت من محورين أحدهما أفقي والآخر عامودي**

2. **<u>البرمجة الشبكية:</u>**

**تعرف الشبكة على أنها تمثيل بياني لأنشطة المشروع بطريقة تبين التسلسل والتتابع المنطقي لأنشطة المشروع والأوقات اللازمة لتنفيذ هذه الأنشطة من لحظة بداية المشروع وحتى نهايته**

❖ **عناصر الشبكة:**

**1.النشاط :وهو أحد وظائف المشروع والذي يتطلب إكماله كمية محددة من الوقت <u>والموارد وتتمتع أنشطة المشروع بالخصائص التالية:</u>**

- **التتابع**
- **التفرد**

- **التعقيد**
- **الترابط**
- **الاعتمادية**

❖ **طرق رسم النشاط:**

- **النشاط على السهم**
- **النشاط على القطب**

**2.الحدث :وهي لحظة البدء بنشاط معين أو لحظة الانتهاء منها، والحدث هو نتيجة نشاط أو أكثر**
**3.المسار :وهو عبارة عن سلسلة من الأنشطة المتتابعة التي تربط بين نقطة البدء بالمشروع ونقطة إتمامه.**
**4.المسار الحرج :وهو سلسلة من الأنشطة الحرجة المتتابعة التي تربط بين نقطة بدء المشروع ونقطة نهايته، وهو أطول المسارات**
**5.النشاط الحرج :وهو النشاط الذي يترتب على تأخيره تأخر المشروع ككل.**
**6.النشاط الوهمي :وهو نشاط ليس له وجود ويستخدم فقط لتسهيل رسم الشبكة وبيان العلاقة بين الأحداث**

❖ **البرمجة الشبكية باستخدام أسلوب بيرت:**

**يحديد ثلاثة أوقات محتملة لإنهاء كل نشاط من أنشطة المشروع وهي:**

**A. الوقت المتفائل ويرمز له في الشبكة بالرمز a وهو أقصر وقت ممكن لتنفيذ النشاط**
**B. الوقت المتشائم ويرمز له b وهو أطول وقت ممكن لتنفيذ النشاط**
**C. الوقت الأكثر احتمالاً ويرمز له على الشبكة m وهو الوقت الأكثر احتمالاً أن يتم تنفيذ النشاط به.**

❖ **تسريع المشروع:**

**وهي العملية التي يتم من خلالها تسريع وقت إنهاء المشروع مع الاستعداد لتحمل التكاليف الإضافية المترتبة على هذا التسريع.**
**وعند القيام بتنفيذ عملية التسريع من المفيد الانتباه إلى المرتكزات التالية:**

1. **إن عملية تسريع المشروع ليست اعتباطية**
2. **إن عملية تسريع المشروع ليست مزاجية**
3. **إن عملية التسريع تبدأ بالأساس على المسار الحرج لأنه المسار الأطول**

# ضمان جودة البرمجيات

❖ ما هو ضمان جودة البرمجيات؟

هو نشاط مظلة يطبق من خلال عملية البرمجة ولا يجب تأجيله إلى ما بعد توليد الشيفرة.

➢ ويتضمن:

1. منهج إدارة الجودة
2. التكنولوجيات الفعالة لهندسة البرمجيات
3. المراجعات التقنية الرسمية
4. استراتيجيات الاختبار المتعددة المستويات
5. ضبط توثيق البرمجيات
6. عملية برمجة لضمان التوافق مع مترية تطوير البرمجيات
7. آليات أخذ القياس وكتابة التقارير

❖ ما هي الصفات التي يمكن قياسها في خصائص البرمجية؟

1. التعقيد المساراتي
2. التلاحم
3. عدد نقاط الوظيفة
4. عدد أسطر الشيفرة

❖ أنواع الجودة:

- جودة التصميم:

مجموعة المميزات التي يحددها المصممون لشيء
ويتضمن: متطلبات، مواصفات، تصميم النظام

- جودة التوافق المترية:

هي درجة اتباع مواصفات التصميم وكلما ازدادت درجة التوافق ارتفع مستوى جودة التوافق
ويتضمن: التركيز على الانجاز

❖ ما الذي نقصده بالتحكم في الجودة؟

هو سلسلة عمليات التفتيش والمراجعات والاختبارات المستخدمة من خلال دورة التطوير للتحقق
ويتضمن التحكم بالجودة :حلقة أراء وتعليقات عن عملية البرمجة

- المفهوم الأساسي في التحكم بالجودة هو أن تكون منتجات العمل معرفة بمواصفات قابله للقياس

❖ تعريف جودة البرمجيات:

هو التوافق بين المتطلبات الوظائفية والأداء المعرفين بوضوح مع مترية التطوير الموثقة بوضوح أيضا

➢ يركز تعريف جودة البرمجيات على 3 نقاط رئيسية:

1. المتطلبات البرمجية
2. المترية المحددة
3. المتطلبات الضمنية

❖ ما هي نشاطات مجموعةSQA ؟

1. وضع خطة SQA
2. المشاركة في وضع مواصفات عملية البرمجة
3. مراجعة نشاطات هندسة البرمجيات
4. تدقيق منتجات العمل المتعلقة بالبرمجية للتأكد من توافقها
5. التحقق من توثيق ومعالجة الانحرافات في العمل البرمجي
6. تسجيل أي عدم توافق وإعداد تقرير بشأنه للإدارة العليا

7. تنسيق التحكم وإدارة التغيير
8. المساعدة في تجميع وتحليل متربة البرمجية

❖ مراجعات البرمجية:

الهدف منها :هي آلية لضمان نقاء عملية هندسة البرمجيات إذ تساعد في كشف الأخطاء وإزالتها

❖ تضخم العيوب وإزالتها:

لتوضيح نشوء الأخطاء وكشفها خلال مراحل التصميم الأولى والتصميم التفصيلي والتشفير

❖ أهداف FTR :

1. كشف الأخطاء الوظائفية والمنطقية الناجمة عن الانجاز
2. التحقق من قابلية تطبيق البرمجية للاحتياجات المطلوبة منها
3. التأكد من توافق المترية المعرفة سلفا
4. توفير الانتظام في عملية التطوير.
5. جعل المشاريع أكثر قابلية للإدارة.

❖ إرشادات المراجعة:

1. مراجعة المنتَج وليس المنتِج
2. وضع جدول أعمال والتقيد به
3. الحد من المناظرات
4. عرض جميع المشاكل بصراحة
5. توثيق الملاحظات كتابيا
6. حصر المشاركة بعدد قليل من الأشخاص
7. إعداد لائحة تدقيق لكل منتج عمل
8. تخصيص الموارد والجدول الزمني لإجراءات FTR
9. إجراء التدريب المناسب للمراجعين
10. مراجعة المراجعات المنصرمة

❖ قياسات الموثوقية:

- موثوقية العتاد تبني توقعاتها على الأعطال الناجمة عن عوامل البلى أكثر من أعطال التصميم
- في البرمجيات جميع الأعطال يمكن ردها إلى مشاكل في التصميم والانجاز
- أبسط طريقة لقياس موثوقية نظام كمبيوتري هي اعتماد الزمن الوسطي بين الأعطالMTBF
- MTBF= MTTF + MTTR

❖ قياسات التوفر:

- توفر البرمجية هي احتمال أن يعمل النظام في لحظة معينة من الزمن بشكل مطابق للمتطلبات الموضوعة
- AV=[ MTTF / (MTTF + MTTR)] * 100%
- القياس MTBF للموثوقية ذو حساسية متساوية ل MTTR و MTTFأما قياس التوفر فهو ذو حساسية أكبر تجاه MTTR

❖ كيف نستطيع تحقيق الأمان وتحليل المخاطر للبرمجية؟

1. من خلال تنفيذ عملية نمذجة وتحليل
2. بعد حصر المخاطر المتوقعة يمكننا استخدام تقنيات التحليل لتحديد درجة خطورتها واحتمال وهذه التقنيات هي:

- تحليل شجرة العيوب: تعتمد على إنشاء مخطط بياني يتألف من تركيبات متسلسلة من الأحداث بحيث يمكن لهذه التركيبات أن تقود إلى نتيجة عشوائية أو إلى حالة معينة للنظام
- منطق الزمن الحقيقي: تعتمد نموذجا يقوم على تحديد الأحداث مع الأفعال الموافقة لها
- النماذج التي تعتمد على شبكاتPetri : تستخدم في تحديد أكثر العيوب عشوائية

❖ الفرق بين الموثوقية والأمان؟

- الموثوقية: تستخدم التحليل الإحصائي لتحديد احتمال حدوث الأعطال البرمجية وليس من الضروري أن يكون العطل مفاجئ
- الأمان: يتفحص كيفية نشوء الأعطال المفاجئة

❖ **خطة ضمان جودة البرمجيات وفق:IEEE**

1. **الغاية من الخطة**
2. **الإدارة**
3. **العوامل القياسية والممارسات والاصطلاحات**
4. **الاختبار**
5. **توثيق المشاكل وإجراءات التصحيح**
6. **ضبط الشيفرة**
7. **ضبط الموردين**
8. **التدريب**

❖ **مقاييس الجودة فيISO 9000 :**

- **نظام ضمان الجودة: هو البنية التنظيمية والمسؤوليات والإجراءات وعمليات البرمجة والموارد اللازمة لإنجاز إدارة الجودة**
- **ISO 9000 :تشرح عناصر ضمان الجودة بتعابير عمومية ولذا فهي قابلة للتطبيق في جميع مجالات الأعمال**
- **مقياس الجودةISO 9001 : هو مقياس ضمان الجودة المطبق في هندسة البرمجيات يتضمن هذا المقياس 20 مطلب يجب توفرها لتحقيقه**
- **ISO 9000 :هو أحد المقاييس المطورة لتفيد في استخدامها لعملية البرمجة.**

## عمليات هندسة المتطلبات

❖ إدارة المتطلبات:

مجموعة من الأنشطة التي تساعد فريق المشروع على تعريف والتحكم في متابعة المتطلبات والتغيرات كلما تقدم المشروع حيث:

1. يتم تعريف المتطلبات اولا وتسمى بالنوع
2. يتم تطوير جدول التعقب مثل الملامح والمصدر
3. ويتم تحديثها في أي وقت يتم فيه تعديل المتطلبات.

❖ نموذج التحليل:

الغرض هو توفير توصيفات للمعلومات المطلوبة والوظائف والمجالات السلوكية للنظم المبنية على الكمبيوتر.

➢ عناصر نموذج التحليل:

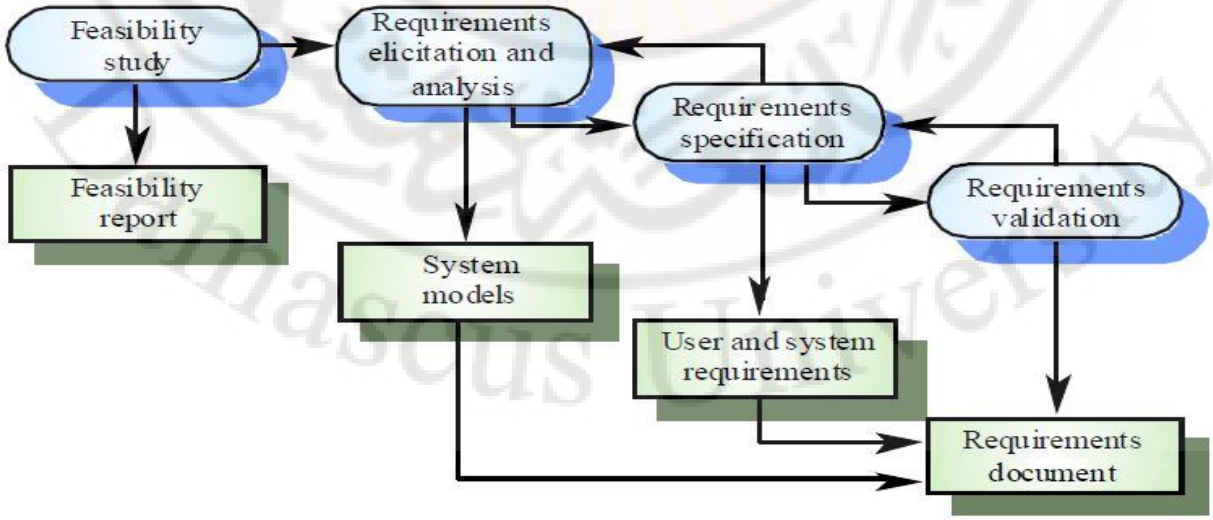
1. العناصر المبنية على السيناريو :وتصف النظام من وجهة نظر المستخدم.
2. العناصر المبنية على الفئة :العلاقات بين الكائنات التي تدار بواسطة الفعل وسماتها.
3. العناصر السلوكية :تصور النظام وتصرف الفئة كحالات وانتقالات بين الحالات.
4. العناصر التدفقية المنحى :تبين كيفية انسياب المعلومات من خلال النظام

❖ عمليات هندسة المتطلبات:

هي العمليات التي تستخدم لاكتشاف وتحليل والتحقق من متطلبات النظام.

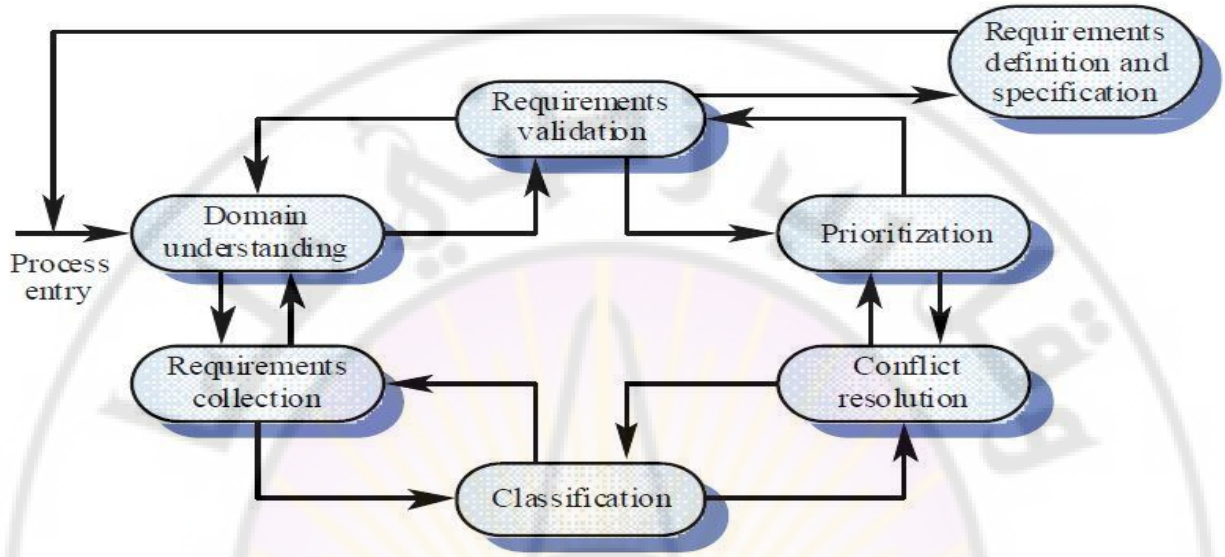
❖ الأنشطة العامة الشائعة في هذه العمليات:

1. استبيان أو أستباط واستخرج المتطلبات.
2. تحليل المتطلبات.
3. التثبت من المتطلبات.
4. إدارة المتطلبات.



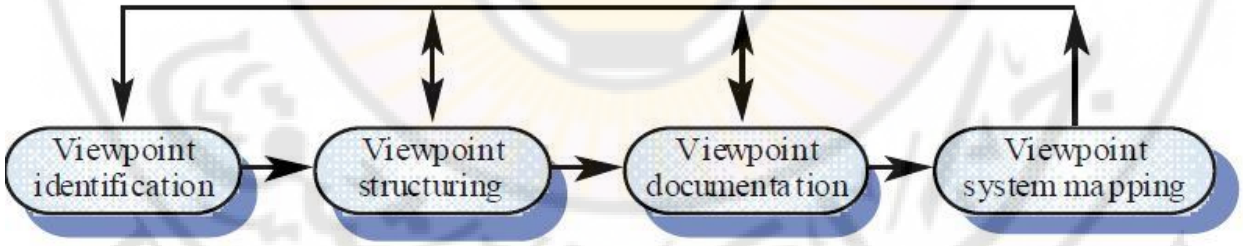
❖ **عملية تحليل المتطلبات:**

**تتم بفهم المجال وتحديد الأولويات وتحقق وتجميع المتطلبات وحل التعارض وتصنيف وتعريف وتوصيف المتطلبات.**




❖ **وجهات النظر الخارجية:**

**ووجهات النظر هي الطريقة طبيعية لتشير استبيان المتطلبات، ومن السهل نسبيا تحديد ما إذا كانت وجهة النظر تتحقق، وتستخدم وجهات النظر والخدمات في تشييد المتطلبات الغير وظيفية.**




❖ **مكونات الطريقة الموجهة بوجهة النظر:**

1. **تعريف وجهة النظر :اكتشاف وجهات النظر التي تستقبل خدمات النظام وتعريف هذه الخدمات.**
2. **وجهة نظر تخطيط النظام**
3. **النماذج القياسية للطريقة الموجهة بوجهة النظر.**

❖ **التحقق من المتطلبات:**

**يهتم بإظهار واثبات أن المتطلبات تقوم فعلا بتعريف النظام الذي يريده المستهلك.**

❖ **فحص واختبار المتطلبات:**

1. **التحقق :هل يوفر النظام الوظائف التي تلبى بأفضل أسلوب**
2. **التماسك أو المتانة :هل تتعارض المتطلبات مع بعضها.**
3. **الاكتمال :هل تم تضمين كل الوظائف المطلوبة للمستهلك.**

4. الواقعية :هل يمكن تنفيذ كل المتطلبات
5. قابلية التحقق :هل يمكن فحص واختبار المتطلبات.

❖ مراجعة المتطلبات:

مراجعات عامة يجب أن تنفذ خلال استنباط وصياغة تعريف المتطلبات، ويجب أن يقوم بهذه المراجعات كل من المستهلك وطاقم تنفيذ العقد

❖ فحوصات المراجعة:

1. قابلية التحقق منها.
2. قابلية الفهم.
3. قابلية التتبع.
4. قابلية التكيف.

❖ إدارة المتطلبات:

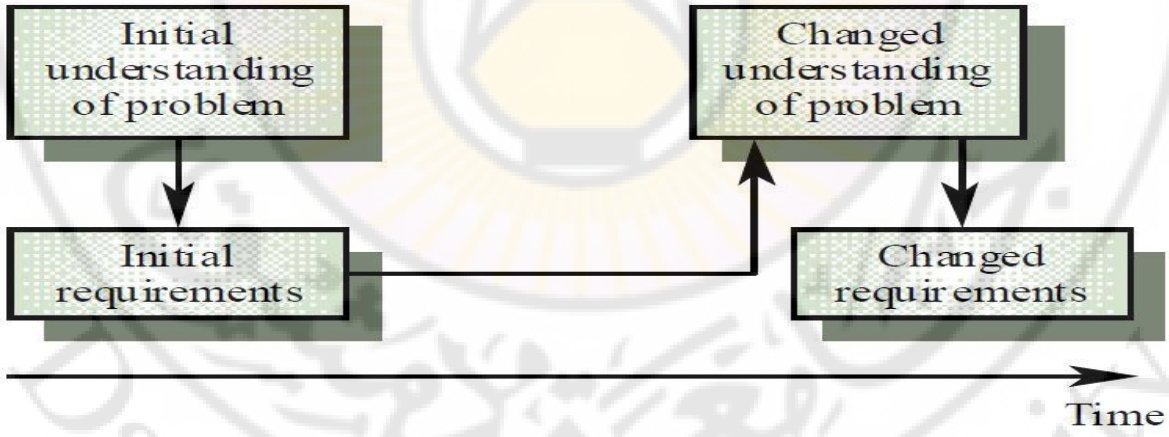
هي عملية إدارة تغيير المتطلبات خلال عملية هندسة المتطلبات وتطوير النظام.

➢ تغيير المتطلبات:

أولوية المتطلبات من تغييرات وجهات النظر المختلفة خلال عمليات التطوير.

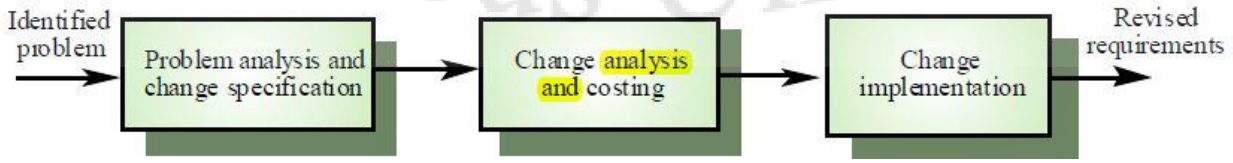
➢ ارتقاء المتطلبات:

عن طريق الفهم المبدئي للمشكلة ووضع متطلبات مبدئية




❖ إدارة تغيير المتطلبات:

تتضمن: المشكلة المعرفة – تحليل المشكلة وتغيير المواصفات– تحليل التغيير والتكلفة – تنفيذ التغيير –مراجعة المتطلبات.



❖ **المراحل الأساسية:**

1. **تحليل المشكلة :بمناقشة المتطلبات والتغييرات المقترحة.**
2. **تحليل التغيير والتكلفة :بمعرفة وتخمين وتقدير مدى تأثير التغيير على المتطلبات الأخرى.**
3. **تنفيذ التغيير :بتعديل مستند المتطلبات والمستندات الأخرى لتعكس التغيير الذي تم.**

# متطلبات البرمجيات

تبين مجموعة المتطلبات ما يجب أن يقوم به النظام وتقوم بتعريف القيود على تشغيل هذا النظام وتنفيذه، وتبين مجموعة المتطلبات الوظيفية الخدمات التي يوفرها النظام، وتبين مجموعة المتطلبات غير الوظيفية القيود التي يتم خلالها تطوير النظام.

- نماذج عمليات البرمجيات:

1. نموذج الشلال أو النموذج الإنحدارى
2. النماذج المتزايدة
3. النموذج التزايدي: يسلم البرمجيات فى قطع صغيرة قابلة للاستخدام تتراكب كل قطعة منها على القطع السابقة.
4. نموذج التطبيق والتطوير المفاجئ

- نماذج العمليات الإرتقائية:

1. النموذج الأولى : خطوة أولى جيدة عندما تكون للمستهلك حاجة ممكنة لكن بدون تفاصيل عملية
2. النموذج الحلزوني :يجمع بين طبيعة النموذج الأولى مع خصائص التحكم والعمل التقليدي للنموذج الخطى المتتالي.
3. نموذج التطوير المتزامن :يشبه النموذج الحلزوني وغالبا ما يستخدم عند تطوير تطبيقات العميل الخادم.

- دورة حياة تطوير البرمجيات:

1. تحديد وتعريف المتطلبات.
2. تصميم النظام.
3. كتابة البرنامج.
4. اختبار وحدات البرنامج واختبار النظام.
5. تسليم النظام.
6. صيانة النظام.

- مرحلة تجميع المتطلبات:

تتم هذه المرحلة على خطوتين هما:

1. دراسة السوق وتجميع متطلبات البرنامج.
2. تحليل المتطلبات وفهم العلاقات بينهم.

- خطوات تحديد المتطلبات:

الاجتماعات مع العميل للتعرف على المتطلبات.

1. تسجيل المتطلبات في وثائق أو قاعدة بيانات
2. إعادة تسجيل المتطلبات رياضيا
3. التثبت والتحقق من المتطلبات.

- هندسة المتطلبات:

هي عملية إنجاز الخدمات التي يطلبها المستهلك من النظام والقيود التي يعمل ويطور فيها النظام،

- أنواع المتطلبات:

1. متطلبات مستخدم :جمل بلغة طبيعية مع أشكال توضيحية للخدمات التي يوفرها النظام وقيود التشغيل مكتوبة للمستهلك.

2. متطلبات النظام :وثيقة هيكلية تبين الوصف التفصيلي لخدمات النظام، مكتوبة كعقد بين المقاول والمستهلك.
3. مواصفات البرمجيات :وصف تفصيلي للبرمجيات.

❖ المتطلبات الوظيفية وغير الوظيفية:

- المتطلبات الوظيفية :يجب توفير بيان وافادة وعرض الخدمات التي يوفرها النظام، وكيفية تصرف النظام في مواقف معينة.
- متطلبات المجال :المتطلبات التي تأتى من مجال التطبيق للنظام والخصائص المنعكسة من هذا المجال.

❖ التصنيفات الغير وظيفية:

1. متطلبات المنتج :وهي متطلبات مواصفات تصرف المنتج المسلم بطريقة معينة مثل سرعة التنفيذ والاعتمادية وغيرها.
2. المتطلبات التنظيمية :وهي متطلبات تعبر عن نتائج سياسات المنظمة والإجراءات مثل المعايير المستخدم، ومتطلبات التنفيذ وغيرها.
3. المتطلبات الخارجية :وهي متطلبات تنشأ من عوامل خارج النظام مثل متطلبات السلطة التشريعية وغيرها

❖ الهدف :

مقصد عام يتحقق للمستخدم مثل سهولة الاستخدام.
تساعد الأهداف المطور على بلوغ وتحقيق رغبات ومقاصد مستخدمي

❖ متطلبات المجال:

مستخرج من مجال تطبيق ووصف خصائص نظم والملامح التي تعكس المجال وقد تكون متطلبات وظيفية جديدة

❖ مشاكل متطلبات المجال:

1. قدرة الفهم
2. الوضوح

❖ اللغة المهيكلة للمواصفات:

مواصفات مبينة على شكل ثابت:
تعريف الوظيفة أو المكون، وصف المدخلات ومن أين يأتي، وصف المخرجات والى أين تذهب، الإشارة إلى أي مكونات أخرى مطلوبة، الشروط، والتأثيرات الجانبية إذا كانت موجودة.

❖ مواصفات واجهة المستخدم:

أنواعها:

1. الواجهات الإجرائية
2. هياكل البيانات التي سيتم تبادلها
3. تمثيل البيانات
4. التدوين الشكلي

❖ وثائق المتطلبات: هي إفادة رسمية بما هو مطلوب من مطور النظام، ويجب أن تتضمن كلا من تعريف المتطلبات ومواصفاتها

❖ هيكل توثيق المتطلبات:

وتتضمن مقدمة، مفردات، تعريف متطلبات المستخدم، بنية النظام، مواصفات متطلبات النظام، نماذج النظام، ارتقاء النظام، ملاحق، وفهرس.