الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

كـلـيـــة هندسة الحاسوب والمعلوماتية والاتصالات
Faculty of Computer & Informatics
and Communications Engineering

تتمة الفصل الثاني

إدارة زمن المشروع

د.م. حسان أحمد

Hassan.Ahmad@spu.edu.sy , istamo48@mail.ru

5. تطوير الجدول الزمني

5.1 طريقة الجدولة باستخدام الأعمدة الطولية (مخططات غانت) (Bar chart)

- تستخدم أعمدة طولية لتمثيل النشاطات اللازمة للتنفيذ،
- تستخدم الأحرف الأبجدية للدلالة على الأعمال أو النشاطات،
- تستخدم الرموز الرقمية للدلالة على الأطوال الزمنية التقديرية لهذه النشاطات.
- تعد خرائط الأعمدة من التقنيات المستحدثة في جدولة البرامج والمشاريع.
  - قدرة هذا النوع من الخرائط على تصور العلاقة الزمنية بين النشاطات المختلفة، من حيث درجة تحكمها وتأثيرها في المواعيد المقررة لاستكمال البرامج والمشاريع.
  - دورها في دعم وترشيد القرارات المتخذة للحكم في مسائل تتعلق بتخطيط وتوجيه الأداء لتنفيذ المشاريع

ونقدم فيما يلي مثالاً في تطوير قرص مدمج لدمشق كأقدم عاصمة.

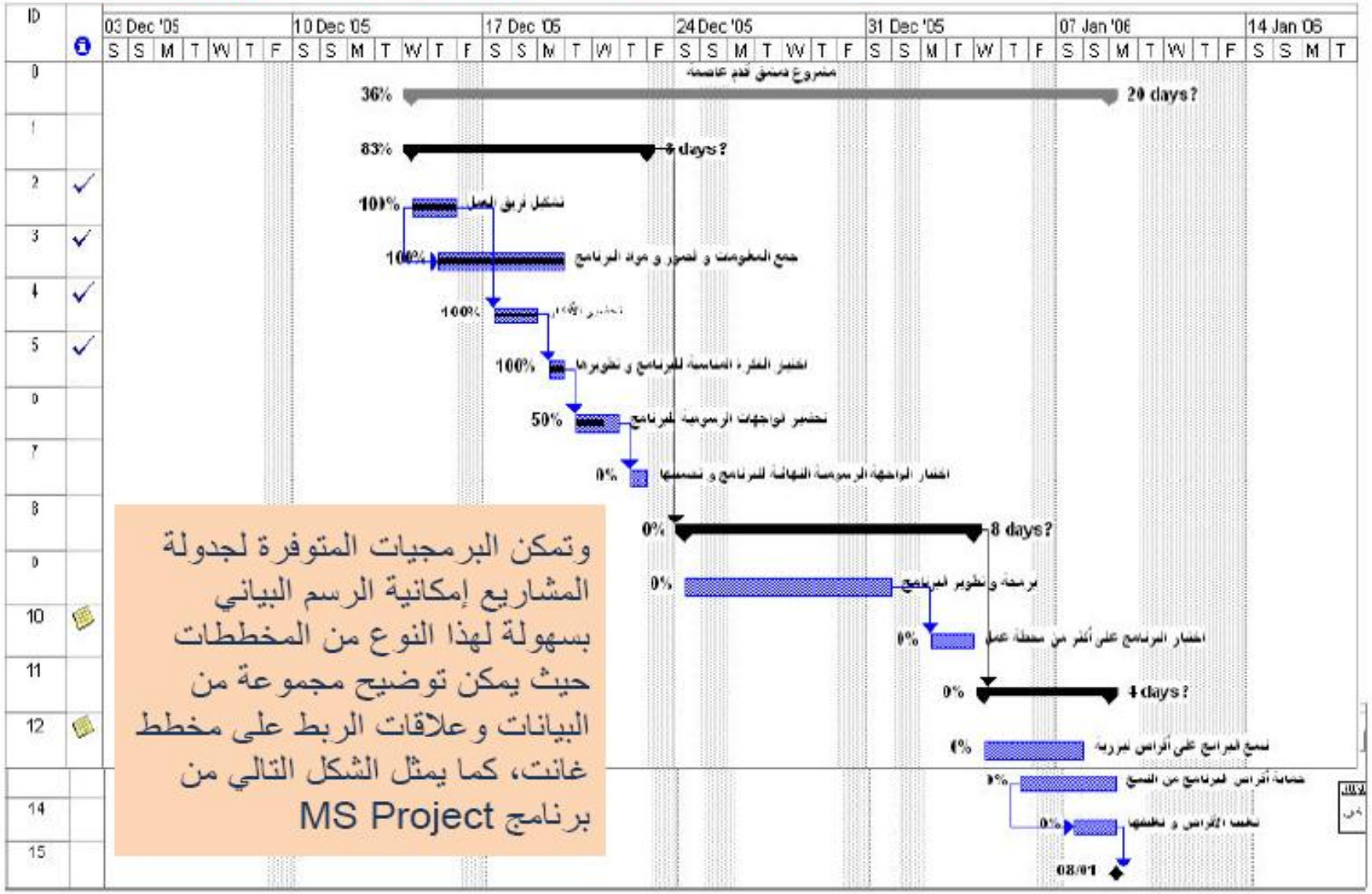
ID	Task Name	Duration	Start	Finish
0	مشروع أقدم عاصمة	20 days?	Wed 14/12/05	Sun 08/01/06
1	مرحلة التحضير للمشروع	8 days?	Wed 14/12/05	Thu 22/12/05
2	تشكيل فريق العمل	2 days?	Wed 14/12/05	Thu 15/12/05
3	جمع المعلومات والصور ومواد البرنامج	4 days?	Thu 15/12/05	Mon 19/12/05
4	تحضير الأفكار	2 days?	Sat 17/12/05	Sun 18/12/05
5	اختيار الفكرة المناسبة للبرنامج وتطويرها	1 day?	Mon 19/12/05	Mon 19/12/05
6	تحضير الواجهات الرسومية للبرنامج	2 days?	Tue 20/12/05	Wed 21/12/05
7	اختيار الواجهة الرسومية النهائية للبرنامج وتحسينها	1 day?	Thu 22/12/05	Thu 22/12/05
8	مرحلة تطوير البرنامج	8 days?	Sat 24/12/05	Tue 03/01/06
9	برمجة وتطوير البرنامج	6 days?	Sat 24/12/05	Sat 31/12/05
10	اختبار البرنامج على أكثر من محطة عمل	2 days?	Mon 02/01/06	Tue 03/01/06
11	مرحلة التحزيم والتعليب والتسليم	4 days?	Wed 04/01/06	Sun 08/01/06
12	نسخ البرنامج على أقراص ليزرية	3 days?	Wed 04/01/06	Sat 07/01/06

مثال في تطوير قرص مدمج لدمشق كأقدم عاصمة. وقد كانت حصيلة البيانات والمعلومات المطلوبة لجدولة هذا المشروع كما هو مبين.

بعد ذلك يتم جدولة هذه الأنشطة وفقًا لنمط تتابعها الزمني وذلك باستخدام فكرة التخريط البياني لمواعيد البدء والانتهاء من كل مرحلة عمل في المشروع

مخططات غانت

خريطة الأعمدة الطولية لمشروع تطوير قرص مدمج عن دمشق



وتمكن البرمجيات المتوفرة لجدولة المشاريع إمكانية الرسم البياني بسهولة لهذا النوع من المخططات حيث يمكن توضيح مجموعة من البيانات وعلاقات الربط على مخطط غانت، كما يمثل الشكل التالي من برنامج MS Project

مخطط غانت وفق برنامج MS Project

ID	Task Name	Duration
1	**كوسباس**	27 wks
2	تقديم العرض	0 days
3	اعتماد نقط المشروع	2 wks
4	تقديم العرض و الجدول الزمني و اعتماده	1 wk
5	اجتماع بدء المشروع	0 wks
6	**الطور الأول**	15 wks
7	دراسة الأقسام الرئيسية في الشركة	3 wks
8	مناقشة حصيلة دراسة الأقسام و الخطط الموضوعة	2 wks
9	تحديد المتطلبات اللازمة لتطبيق الأنظمة	0 wks
10	تخصيص البرامج	6 wks
11	تحضير البيانات اللازمة و ادخالها	4 wks
12	**التدريب على الأنظمة**	4 wks
13	ادارة المواد	4 wks
14	ادارة المستودعات و هندسة الانتاج	4 wks
15	ادارة الانتاج	4 wks
16	تدابير المشتريات و ادارة الموردين	4 wks
17	المبيعات و عناية الزبائن	4 wks
18	العمل الفعلي على البرامج	3 wks
19	انتهاء الطور الأول	0 wks
20	**الطور الثاني**	3 wks
21	المالية	3 wks
22	**الطور الثالث**	3 wks
23	ذاتية العاملين	3 wks
24	الرواتب و الأجور	3 wks
25	**الطور الرابع**	3 wks
26	نظام الخدمات و الصيانة	3 wks

6. تطبيقات الجدولة

1) **تحليل الموارد المادية والبشرية للمشروع: يحتاج القائمون لإنجاز أي مشروع (صناعي، إنشائي، أو تجاري) في الوقت المحدد على موارد، وعادة تكون الموارد والتي هي العناصر الضرورية من القوى العاملة (الإداريين، العمالة ماهرة، والغير ماهرة.. الخ) أو الموارد المادية (أجهزة، والمواد الأولية الضرورية للعملية الإنتاجية).**

2) **مخططات الموظفين: وهي مخططات تمثل عدد الموظفين المطلوبين خلال فترة تنفيذ المشروع.**

3) **مخططات استخدام المعدات: وهي مخططات تمثل عدد الآليات المستخدمة ونوعها خلال فترة تنفيذ المشروع.**

4) **مخططات استهلاك المواد: وهي مخططات تعبر عن كمية الموارد المستخدمة من كل مادة.**

ولتنظيم وتسهيل عمل الإدارة في متابعة الأعمال تعرض جميع الموارد المستخدمة في المشروع في جدول يسمى **جدول تحليل الموارد.**

- الموارد المستخدمة في كل نشاط
- إجمالي الموارد المستخدمة في تنفيذ المشروع
- الأنشطة التي تم إنجازها بالموارد المتوفرة

7. قياسات البرمجيات

- **القياسات المباشرة للبرمجيات:**
  - الكلفة والجهد المطبقين
  - أسطر الترميز
  - سرعة التنفيذ
  - حجم الذاكرة اللازمة للتنفيذ
  - الأعطال المعلن عنها خلال فترة معينة.
- **القياسات غير المباشرة للبرمجيات:**
  - الوظيفية
  - درجة التعقيد
  - الجودة
  - قابلية الصيانة
  - الموثوقية
  - الفعالية

تصنّف القياسات إلى قياسات تعتمد على حجم البرمجية وأخرى تعتمد على الوظيفة التي تقوم بها هذه البرمجية.

7.1 القياسات الحجمية التوجه للبرمجيات

تُستنتج المقاييس الحجمية التوجه اعتمادًا على حجم البرمجية الناتجة. يبين الجدول التالي المقاييس الحجمية التوجه في مشاريع تطوير البرمجيات في منظمة ما:

المشروع	LOC	الجهد	$(000)	الصفحات/ الوثيقة	الأخطاء	العيوب	العنصر البشري
ألفا	12100	24	168	365	134	29	3
بيتا	27200	62	440	1224	321	86	5
غاما	20200	34	314	1050	256	64	6
...	...	...	...	...	...	...	...

على سبيل المثال، في المشروع **ألفا**، جرى تطوير 12100 سطراً من الترميز بجهد 24 شخص/شهر، وبكلفة 168000 $ يجب الانتباه إلى أن الجهد والكلفة المسجلين في الجدول السابق يمّثلان جميع مراحل تطوير البرمجية (التحليل والتصميم والترميز والاختبار) وليس فقط الترميز. تشير المعلومات الأخرى عن المشروع **ألفا** إلى أنَّه جرى تطوير 365 صفحة من الوثائق، وتسجيل 134 خطأً قبل إصدار البرمجية، ومصادفة 29 عيبًا بعد تسليم البرمجية للزبون خلال العام الأول من التشغيل. وقد عمل ثلاثة أشخاص في تطوير برمجيات المشروع **ألفا**.

مجموعة من المقاييس الحجمية التوجه للمشروع

- ✓ عدد الأخطاء في كل ألف سطر من الترميز
- ✓ عدد العيوب في كل ألف سطر من الترميز
- ✓ كلفة كل سطر من الترميز
- ✓ عدد صفحات الوثائق لكل ألف سطر من الترميز

يمكن إضافة إلى ذلك حساب مقاييس أخرى مثيرة للاهتمام:

- ✓ عدد الأخطاء لكل شخص/شهر
- ✓ عدد سطور الترميز لكل شخص/شهر
- ✓ كلفة كل صفحة وثائق

- ✓ لا تعد المقاييس الحجمية التوجه عمومًا أفضل طريقة لقياس البرمجيات.
- ✓ يدور معظم الجدل حول استخدام عدد أسطر الترميز (LOC) قياسًا أساسيًا.
- ✓ العديد من نماذج التقدير البرمجية الحالية تستخدم (LOC) أو (KLOC) دخلاً رئيسيًا.
- ✓ قياسات (LOC) أو (KLOC) تابعة للغة البرمجة المستخدمة في تطوير البرمجية.

7.2 المقاييس الوظيفة التوجه للمشروع

✓ **تعتمد المقاييس الوظيفية التوجه على قياس الوظيفة التي تقوم بها البرمجية الناتجة عن المشروع .**

✓ **ولما كان من غير الممكن قياس"الوظيفية" قياسًا مباشرًا، فلا بد من استنتاجها بأسلوب غير مباشر باستخدام قياسات مباشرة أخرى.**

1) **نقطة الوظيفة: جرى اقتراح أحد المقاييس الوظيفية التوجُّه، والذي يسمى نقطة الوظيفة، ويجري استنتاج نقاط الوظيفة باستخدام علاقة تجريبية تستند إلى قياسات مباشرة قابلة للعد لنطاق المعلومات البرمجية، وإلى تقييم تعقيد البرمجيات.**

نبين حساب نقاط الوظيفة بإكمال الجدول التالي.

نقاط الوظيفة

- **عدد مدخلات المستخدمين:** يحصى كل دخل للمستخدم يوفِّر معطيات تطبيقية التوجُّه مميزة للبرمجية، يجب التمييز بين المدخلات والاستفسارات التي تحصى على حدا.
- **عدد مخرجات المستخدمين:** يحصى كل خرج للمستخدم يوفِّر معطيات تطبيقية التوجُّه مميزة للمستخدم .ضمن هذه السياق، يشير الخرج إلى التقارير، الشاشات، رسائل الأخطاء وغير ذلك.
- **عدد استفسارات المستخدمين:** يحصى كل استفسار مميز، ويعرف الاستفسار بأنه دخل آني يؤدي إلى توليد استجابة فورية من البرمجية على شكل خرج آني.
- **عدد الملفات:** يُحصى كل ملف رئيسي منطقي، أي تجميع منطقي لمجموعات من المعطيات التي قد تكون جزءًا واحدًا من قاعدة معطيات واسعة أو ملف مستقل.
- **عدد الواجهات الخارجية:** تُحصى جميع الواجهات التي يمكن للآلة قراءتها (كملفات المعطيات على الأقراص) والتي تستخدم لنقل المعلومات من نظام لآخر.

				عامل التثقيل Weighting Factor		
موسطات أخذ القياسات	العدد count		معقّد Complex	متوسط Average	بسيط Simple	
عدد مدخلات المستخدمين		×	6	4	3	=
عدد مخرجات المستخدمين		×	7	5	4	=
عدد استفسارات المستخدمين		×	6	4	3	=
عدد الملفات		×	15	10	7	=
عدد الواجهات الخارجية		×	10	7	5	=
المجموع						

2) حساب نقاط الوظيفة

لحساب نقاط الوظيفة نستخدم العلاقة التالية: $$FP = count\text{-}total \times [0.65 + 0.01 \times SUM(Fi)]$$

حيث: count-total - التعداد الكلي / مجموع جميع المداخل التي حصلنا عليها من الجدول السابق.

Fi - قيم تعديل درجة التعقيد وتستند إلى الإجابات على الأسئلة التالية:

1. هل يتطلب النظام إجراء نسخ احتياطي دوري واستعادة موثوقين؟
2. هل هناك حاجة إلى اتصالات المعطيات؟
3. هل تتطلب الوظائف معالجة موزعة؟
4. هل تشكل فعالية الأداء أمرًا أساسيًا؟
5. هل ستعمل البرمجية ضمن بيئة تشغيل موجودة وتستخدم بكثافة؟
6. هل تتطلب البرمجية إدخالا آنيًا للمعطيات؟
7. هل يحدث الملف الرئيسي تحديثًا مباشرًا؟
8. هل المدخلات أو المخرجات أو الملفات أو الاستفسارات معقَّدة؟
9. هل المعالجة الداخلية معقَّدة؟
10. هل تم تصميم الترميز بحيث يمكن إعادة استخدامه؟
11. هل يتضمن التصميم عمليتي التحويل والتجهيز/ التثبيت
12. هل تم تصميم البرمجية من أجل تجهيز متعدد في مؤسسات مختلفة؟

نقاط الوظيفة

تُستخدم نقاط الوظيفة فور حسابها على وجهٍ مشابه لعدد أسطر الترميز (LOC) لتنظيم قياسات أخرى للبرمجية، مثل:

- **عدد الأخطاء في كل نقطة وظيفة**
- **عدد العيوب في كل نقطة وظيفة**
- **كلفة كل نقطة وظيفة**
- **عدد صفحات الوثائق لكل نقطة وظيفة**
- **عدد نقاط الوظيفة لكل شخص/شهر**

7.3 الطرق الأساسية لتقدير الكلفة في مشاريع تطوير البرمجيات

- ✓ تشكل نفقات الأشخاص المشاركين في مشاريع تطوير البرمجيات النسبة الأكبر من نفقات هذه المشاريع.
- ✓ لذلك يتطلب تقدير الكلفة أن يجري تقدير الجهد الذي يقوم به الشخص بشكلٍ أساسي. وهناك عدة طرق تستخدم لهذه الغاية:

1) تقدير حجم البرنامج: وهي طريقة لتقدير الجهد اعتماداً على التميز المصدري للبرنامج، تعرف باسم نموذج CoCoMo (Constructive Cost Model) يجري تقدير الجهد اعتمادًا على المواصفات الداخلية فقط، وبالتالي تكون دقة التقدير منخفضة في المراحل الأولى من تطوير البرمجية، وهي غير مناسبة في هذه المراحل.
2) تقدير نقطة الوظيفة: يجري في هذه الطريقة تقدير الجهد اعتماداً على تعقيد البرمجية باستخدام الوحدة "نقطة الوظيفة" بالإضافة إلى استخدام مواصفات خارجية للبرنامج.

3) **طريقة التشابه: يجري في هذه الطريقة تقدير الجهد والكلفة اعتمادًا على النتائج السابقة لمشاريع مشابهة للمشروع الحالي .قد يكون من الصعب أحيانًا تحديد فيما إذا كانت مشاريع قديمة مشابهة للمشروع الحالي وكذلك تقدير الفرق بينها.**

4) **طريقة التجميع أو المراكمة: يجري التقدير في هذه الحالة من خلال التحّقق من العمليات وتجميع الجهد المطلوب لها. من الضروري التحقق من جميع العمليات. تعتمد دقة التقدير على دقة العمليات التي يجري التحقق منها، لذلك من الضروري أن يجري تدقيق هذه العمليات في مرحلة مبكرة من إجرائية التطوير.**

7.4 أساسيات نموذج CoCoMo

- نموذج تقدير برمجي جديد للكلفة والجدولة الزمنية، وهو مناسب للنماذج الجديدة في تطوير البرمجيات، مثل:
  - برمجيات الأعمال،
  - البرمجيات الغرضية التوجه،
  - نماذج التطوير التطورية أو الحلزونية (Spiral /Evolutionary)
- قياس حجم البرمجيات:

يستخدم نموذج CoCoMo 2.0 ثلاثة مقاييس مختلفة لقياس حجم المشروع البرمجي:

- نقاط الغرض (Object Points)
- نقاط الوظيفة غير المضبوطة ((Unadjusted Function Points)
- أسطر الترميز المصدري (Source Line of Code, SLC)

نمذجة الجهد

- **الجهد المُفتَرض أو المتصور (Nominal Effort): يُعطى الجهد المَفترض من أجل حجم مشروع ما بالعلاقة التالية:**

$$PM\ nominal = A \times (Size)^{B}$$

  - ويعبر عن هذا الجهد باستخدام الوحدة شخص/شهر (Person/Month, PM)
  - يعرف نموذج CoCoMo عاملا أسيًا من أجل التوفير النسبي أو الخسارة النسبية للامتداد الذي نصادفه عندما يزداد حجم المشروع البرمجي. يجري تمثيل هذا العامل من خلال الأس (B)
  - يستخدم الثابت لتمثيل التأثيرات الخطية على الجهد في المشاريع ذات الحجم المتزايد، ويؤخذ (A=2.94)
- **عوامل الامتداد الأسية (Exponent Scale Factor):**
  - يجري حساب العامل (B) باستخدام المعادلة التالية: $B = 0.91 + 0.01 * SUM_{i=1\ to\ 5}(Wi)$
  - يبين الجدول التالي مستويات التقدير (Rating Levels) الخاصة بعوامل الامتداد الأسية في نموذج CoCoMo 2.0

جدول مستويات التقدير الخاصة بعوامل الامتداد الأسية في نموذج CoCoMo 2.0.

عوامل الامتداد (W_i)	منخفض جداً (5)	منخفض (4)	مفترض (3)	مرتفع (2)	مرتفع جداً (1)	مرتفع جداً جداً (0)

يجري جمع التقديرات الرقمية للمشروع من أجل كل العوامل (*Wi*)، وتستخدم لتحديد عامل الامتداد (B).

مثال1: إذا كان لدينا مشروع برمجي ذو حجم (100 KLOC) وتقدير مرتفع جدًا جدًا (0) من أجل جميع العوامل. سيكون لدينا:

- $W_i=0$؛
- $B=0.91+0.01*0=0.91$؛
- $E=PM=2.94*100^{0.91}=2.94*66=194$ PM (الجهد النسبي)

بفرض لدينا 8 أشخاص يعملون في المشروع، تكون المدة اللازمة للتنفيذ هي:

Duration = 194/8 = 24.25 Months

مثال 2: إذا كان لدينا مشروع برمجي ذو تقدير منخفض جدًا (5) من أجل جميع العوامل. سيكون لدينا:

- $W_i=0+1+2+3+4+5=15$؛
- $B=0.91+0.01*15=1.06$؛
- $E=PM=2.94*100^{1.06}=2.94*209=614$ PM (الجهد النسبي)

بفرض لدينا 8 أشخاص يعملون في المشروع، تكون المدة اللازمة للتنفيذ هي:

Duration = 614/8= 76.75 Months

- **التوفير والخسارة الناتجة عن الامتداد**
  - إذا كان (B <1.0) سيبدي المشروع توفيرًا نتيجة الامتداد. إذا تضاعف حجم المشروع، فإن الجهد اللازم سيكون أقل من ضعف الجهد السابق. تزداد إنتاجية المشروع بازدياد حجم المشروع؛
  - إذا كان (B =1.0) سيكون هناك توازنًا بين التوفير والخسارة الناتجة عن الامتداد. يستخدم هذا النموذج الخطي من أجل تقدير كلفة المشاريع الصغيرة.
  - إذا كان (B >1.0) سيبدي المشروع خسارًة نتيجة الامتداد. هذا يعود بشكل رئيسي إلى سببين أساسيين:
    - نمو عبء التواصل بين الأشخاص
    - ونمو عبء تكامل الأنظمة الضخمة

سيكون للمشاريع الأضخم عدد اكبر من الأشخاص، وبالتالي سيكون هناك مسارات تواصل أكثر بين الأشخاص.

نهاية المحاضرة الرابعة

&

نهاية الفصل الثاني