ملحق أ

التحليل والتصميم الكائني
Object–Oriented Analysis and Design

بعد دراسة هذا الفصل ستكون قادراً على:

- تعريف المصطلحات الأساسية التالية: الارتباط association، مخطط الصنف class diagram، حدث event، كائن object، صنف الكائن object class، عملية operation، مخطط التسلسل sequence diagram، حالة state، تحول الحالة state transition، لغة النمذجة الموحدة unified modeling language، حالة الاستخدام use case.
- وصف المفاهيم concepts والمبادئ principles التي تقع ضمن الطريقة الكائنية.
- تطوير نموذج model بسيط للمتطلبات requirements باستخدام مخططات استخدام – حالة use – case diagrams.
- تطوير نموذج كائن بسيط لمجال المشكلة مستخدما مخططات الصنف.
- تطوير نموذج بسيط للمتطلبات مستخدما مخططات الحالة والتسلسل.

منهجية النمذجة الكائنية
The Object-Oriented Modeling Approach

لقد تعلمت، بداية من الفصل الأول وحتى العاشر، معلومات عن النهج التقليدي الخاص بتحليل وتصميم النظم. كما اكتسبت معلومات عن كيفية استخدام مخططات تدفق البيانات Data - Flow Diagrams ومخططات علاقات الكينونة Entity – Relationship diagrams لنموذج النظام. ثمة بيئات تكون في حاجة إلى الطرق الكائنية وليس الطرق التقليدية. يغطي هذا الملحق الأساليب techniques والمخططات البيانية diagrams graphical التي يستخدمها محللي النظم في التحليل والتصميم الكائني. كما الشأن مع أساليب النمذجة التقليدية، فإن المستلمات deliverables من أنشطة المشروع باستخدام النمذجة الكائنية هي عبارة عن مخططات تدفق البيانات وعلاقات الكينونة وشرح للمستودعات. ثمة خاصية رئيسية لهذه المخططات في النمذجة الكائنية ألا وهي مدى ارتباطها الشديد ببعضها البعض . تقدم طريقة النمذجة الكائنية عدد من الفوائد، تشمل:

١ – القدرة على معالجة مجالات مشاكل أكثر تحديا.

٢ – الاتصالات المتطورة بين المستخدمين والمحللين والمصممين والمبرمجين.

٣ – القدرة على إعادة استخدام نتائج التحليل والتصميم والبرمجة.

٤ – الاتساق المتزايد بين النماذج التي تم تطويرها أثناء التحليل والتصميم والبرمجة الكائنية.

دورة حياة تطوير النظام الكائني تتضمن تمثيل متطور بشكل تدريجي لمكون النظام (الذي سوف نطلق عليه اسم كائن) من خلال مراحل التحليل والتصميم والتطبيق. في المراحل المبكرة للتطوير، يكون بناء النظام تجريديا، حيث يركز على الصفات الخارجية لنظام التطبيق مثل هيكلة البيانات data structures والتوقيت وتسلسل عمليات المعالجة، وكيف يتفاعل المستخدمين مع النظام. مع تطور النموذج، يصبح النموذج مفصلا بشكل أكثر، محولا التركيز إلى كيف سيتم بناء النظام وكيف سوف يعمل. في مرحلة التحليل، يتم تطوير نموذج تطبيق حقيقي مبينا خصائصه الهامة. يقوم بتجريد المفاهيم من مجال التطبيق ويشرح ما يجب على النظام المقصود عمله وليس كيفية عمله. يقوم النموذج بتحديد السلوك الوظيفي للنظام، بعيدا عن الاعتبارات الخاصة بالبيئة التي سوف يتم تنجيز النظام بها في النهاية.

في مرحلة التصميم، يتم تكييف وصقل نموذج التحليل ذي التوجه التطبيقي حتى يناسب بيئة التنجيز المقصودة. يتبع ذلك مرحلة التنجيز، حيث يتم تنجيز التصميم باستخدام لغة برمجة و/ أو نظام إدارة قاعدة بيانات. يطلق على الأساليب ومجموعة الرموز التي يتم دمجها في لغة قياسية شيئية لغة النمذجة الموحدة (modeling language)، أو UML.

لغة النمذجة الموحدة (u m l)
مجموعة رموز تسمح لواضع النموذج بأن يحدد ويصور ويبني أدوات نظم البرمجيات، بالإضافة إلى نماذج المشاريع

تشمل الأساليب ومجموعة الرموز المستخدمة في لغة النمذجة الموحدة ما يلي:

- حالات استخدام، تمثل المتطلبات الوظيفية أو «ماذا» الخاصة بالنظام.
- مخططات الصنف، التي تبين الهيكل الثابت للبيانات والعمليات التي تعمل على البيانات.
- مخططات الحالة، التي تمثل النماذج الديناميكية الخاصة بكيفية تغيير الأشياء لحالاتها عند الاستجابة للأحداث.
- مخططات التسلسل، التي تمثل النماذج الديناميكية الخاصة بالتفاعل بين الأشياء.

تسمح لغة النمذجة الموحدة (UML) لوضع النموذج مجموعة رموز و تسمح لوضع النموذج بأن يحدد ويصور ويبني أدوات نظم البرمجيات، بالإضافة إلى نماذج المشاريع. فهي تبنى على وتوحد دلالات الرموز في الطرق الرئيسية الكائنية وتم تبنيها باعتبارها معيار صناعي.

تعد رموز لغة النمذجة الموحدة (UML) مفيدة في التحليل الكائني المصور ونماذج التصميم. فهي لا تسمح لك فقط بتحديد بتحديد متطلبات النظام واغتنام قرارات التصميم، لكنها أيضا تعزز من التواصل بين الأفراد الأساسيين المشاركين في جهود التطوير. يمكن للمطور أن يستخدم تحليلا ما لتصميم نموذج باستخدام رموز لغة النمذجة الموحدة للتواصل مع الخبراء في المجال والمستخدمين وحاملي الأسهم الآخرين.

لتمثيل نظام معقد بفاعلية، يحتاج النموذج الذي تم تطويره إلى أن يتوفر لديه مجموعة صغيرة من الآراء المستقلة الخاصة بالنظام. تسمح لك لغة النمذجة الموحدة (UML) بأن تمثل الآراء المتعددة للنظام باستخدام تشكيلة من المخططات البيانية، مثل مخطط استخدام – حالة use – case diagram، ومخطط الصنف class diagram، ومخطط الحالة state diagram، ومخطط التسلسل sequence diagram، ومخطط التعاون collaboration diagram. يقوم النموذج الأساسي بدمج هذه الرؤى بحيث يمكن تحليل وتصميم وتطبيق النظام بشكل كامل ومتسق.

نرى أولا كيف يمكن تطوير نموذج استخدام – حالة أثناء مرحلة تحليل المتطلبات. بعد ذلك، نريك كيفية تشكيل الهيكل الثابت للنظام باستخدام مخططات الصنف والكائن. ثم تتعلم كيفية السيطرة على الأوجه الديناميكية باستخدام مخططات الحالة والتسلسل. وأخيرا، نقدم لك شرحا موجزا عن مخططات المكونات، التي يتم توليدها أثناء مرحلة التصميم والتطبيق.

نمذجة حالة استخدام
Use Case Modeling

يتم تطبيق نمذجة استخدام – حالة لتحليل المتطلبات الوظيفية لنظام ما. يتم إنجاز نمذجة استخدام – حالة في المراحل الأولى من تطوير النظام (أثناء مرحلة التحليل) لمساعدة المطورين على فهم المتطلبات الوظيفية للنظام دون القلق بشأن كيفية تنجيز هذه المتطلبات. تتميز العملية بالتكرار في طبيعتها؛ لذلك يحتاج المطورون إلى إشراك المستخدمين في المناقشات طوال عملية تطوير النموذج وأخيرا يصلون إلى اتفاقا بشأن ماهية المتطلبات.

ممثل
Actor
وجود كينونة خارجية تتفاعل مع النظام (مشابه لوجود الكينونة الخارجية في وضع تخطيطات تدفق البيانات).

حالة استخدام
Use case
تسلسل كامل من الأجراءات المرتبطة ببعضها البعض يبدأها ممثل؛ فهي طريقة محددة لاستخدام النظام.

يتكون نموذج حالة استخدام من ممثلين وحالات استخدام. **الممثل** هو وجود كينونة خارجية تتفاعل مع النظام (مشابه لوجود الكينونة الخارجية في وضع تخطيطات تدفق البيانات). ما يتبادل المعلومات مع النظام إما أن يكون شخصا أو شيئا. حالة استخدام يمثل تسلسل من الإجراءات المرتبطة ببعضها البعض يبدأها ممثل؛ فهي طريقة محددة لاستخدام النظام. يمثل الممثل دورا يمكن أن يلعبه المستخدم. يجب أن يدل اسم الممثل على هذا الدور. يساعدك الممثلين في تحديد حالات الاستخدام التي ينفذونها.

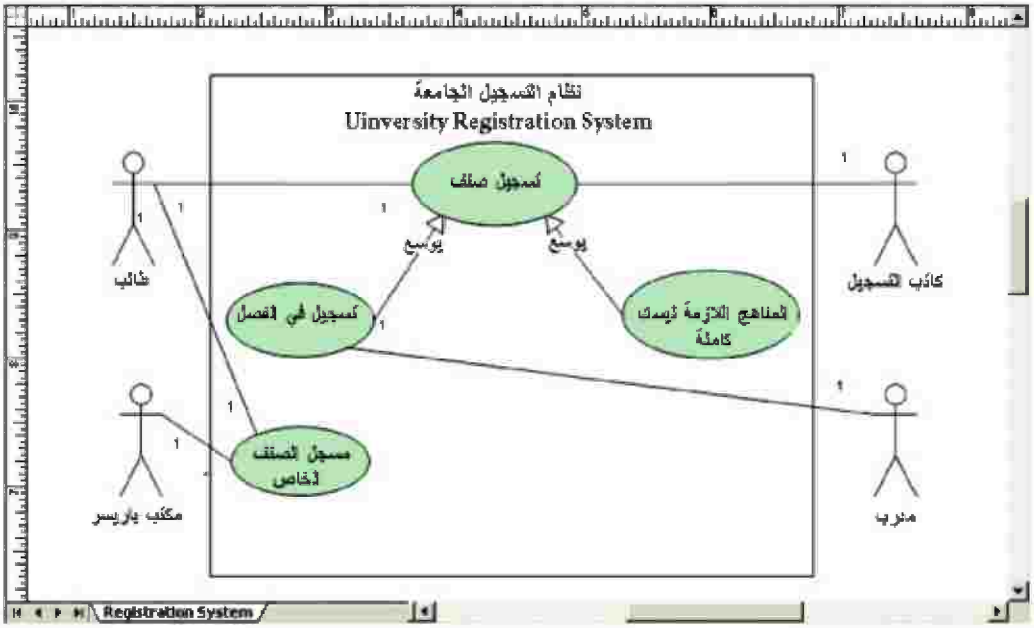
أثناء متطلبات مرحلة التحليل، سيجلس المحلل مع المستخدمين المقصودين للنظام ويقوم بتحليل كامل عن ما هي الوظائف التي يرغبون فيها من النظام. هذه الوظائف ممثلة كحالات استخدام. على سبيل المثال، نظام التسجيل في الجامعة به حالة استخدام للتسجيل في الصف وأخرى لدفع مصروفات الطالب. هذه الحالات حينئذ تدل على التفاعلات النموذجية للنظام مع مستخدميه.

في UML يصور نموذج وحدة المعالجة بشكل تخطيطي كما هو في الشكل أ–١. إن مخطط حالة الاستخدام use – case diagram هذا هو من أجل نظام التسجيل في الجامعة المبين علي هيئة صندوق. خارج الصندوق هناك أربعة عمال – طالب، موظف التسجيل، معلم، مكتب بارسر – الذي يتفاعل مع النظام (المبين علي هيئة خطوط ملامسة للعمال). الممثل موضح كرمز لرجل عصوي ومذكور اسمه أسفل المخطط. وداخل الصندوق، هناك أربع حالات استخدام – تسجيل في الصف الدراسي، التسجيل للفصول الخاصة، المناهج اللازمة غير الكاملة وفاتورة الطالب المبينة كقطاع بيضاوي بداخله أسمائهم. حالات الاستخدام هذه يقوم بها العمال خارج النظام.

حالة الاستخدام هي عادة ما يبدأها الممثل. على سبيل المثال، يصدر فاتورة الطالب مكتب بارسر. حالة الاستخدام من الممكن أن تتفاعل مع العمال بخلاف الشخص الذي يقوم بإصدارها أو بدايتها. حالة استخدام فاتورة الطالب، على الرغم من إصدارها من قبل مكتب بارسر، فإنها تتفاعل مع الطلاب بمراسلاتهم من خلال البريد الإلكتروني عن فواتير التعليم الخاص. هناك حالة استخدام أخرى، تسجيل خلال الفصل، قام بتنفيذها اثنين من العمال ن الطالب وموظف التسجيل. حالة الاستخدام هذه تؤدي سلسلة من الإجراءات المرتبطة الهادفة إلى تسجيل الطالب في الفصل.

إن الأعداد في كل نهاية من خطوط التفاعل توضح عدد الأمثلة في حالة الاستخدام التي بها يرتبط الممثل، على سبيل المثال، يسبب مكتب بارسر العديد من الأمثلة التي تحدث عن حاله الاستخدام في فاتورة الطالب. كل واحدة لكل طالب بالضبط.

شكل أ-١ تخطيط حالة لاستخدام نظام التسجيل الجامعي باستخدام واجهة ميكروسوفت فيزو.



حالة الاستخدام تدل علي توظيف كامل. يجب عليك أن لا تمثل إجراء فردي الذي هو جزء من الوظيفة الكلية كحالة استخدام. على سبيل المثال، على الرغم من أن الخضوع لنموذج التسجيل ودفع المصروفات هما إجراءين يقوم بهما المستخدم (الطالب) في نظام التسجيل في الجامعة، لم نريهم مثل ما في حالات الاستخدام، لأنهم لا يخصصوا منهج كامل للأحداث؛ كل من هذه الإجراءات يتم تنفيذه فقط كجزء من الوظيفة الكاملة أو حالة الاستخدام. يمكنك التفكير في «نموذج التسجيل الجزئي» كأحد أفعال حالة استخدام في تسجيل الفصل و«دفع المصروفات» كإجراء من إجراءات حالة الاستخدام لفاتورة الطالب.

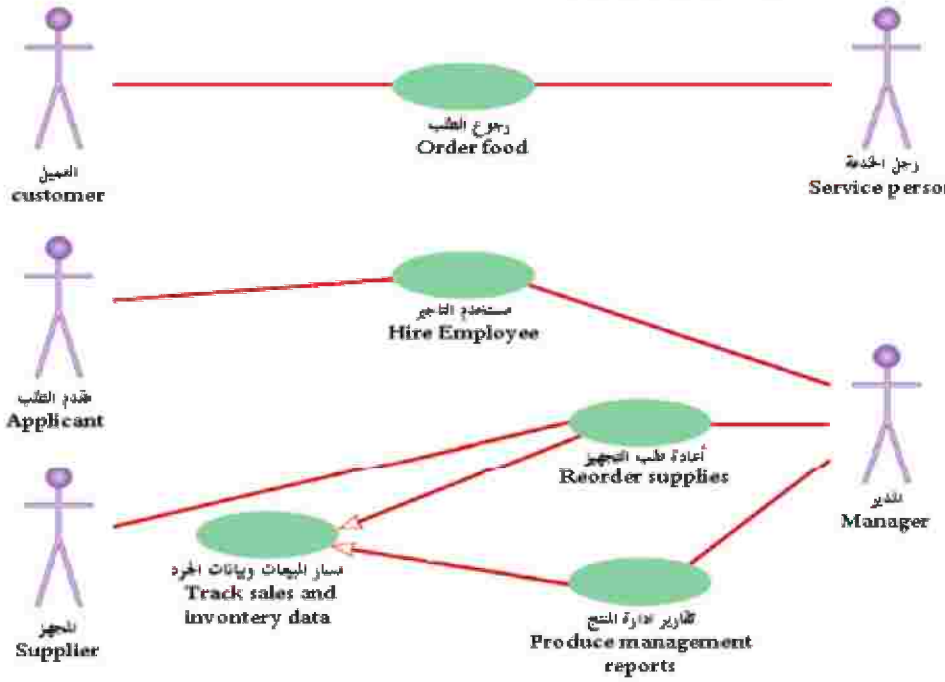
حالة الاستخدام ستشارك في العلاقات مع حالات الاستخدام الأخرى. العلاقة ممتدة، كما هو مبين في مايكروسوفت فيزو كخط مع مثلث مفرغ يشير تجاه حالة استخدام ممتدة ويتم تصنيفها مع رمز الامتدادات الممتدة كحالة استخدام عن طريق إضافة سلوكيات أو تصرفات أو إجراءات جديدة. في الشكل أ-١، على سبيل المثال، التسجيل لحالات الاستخدام الخاصة في الفصل تمتد لحاله الاستخدام للتسجيل في الفصل عن طريق التقاط الإجراءات الإضافية التي تحتاج إلى أن يتم أدائها في تسجيل الطالب في فصل خاص. يتطلب التسجيل للفصول الخاصة إذن سابق من المعلم، بالإضافة إلى خطوات أخرى يتم تنفيذها للتسجيل المنتظم. ربما تفكر في التسجيل في الفصل كمنهج أساسي، الذي نقوم به دائما بشكل مستقل عن سواء قمنا بهذا التوسع أو لم نقم – التسجيل للفصول الخاصة كمنهج بديل يتم تنفيذه فقط تحت ظروف خاصة.

مثال آخر لعلاقة التوسع هو أن بين المناهج المطلوبة أو اللازمة غير المكتملة و حالات الاستخدام للتسجيل في الفصل. التوسعات السابقة هي الأخيرة في المواقف التي لا يأخذ فيها الطالب المسجل للفصل المناهج اللازمة أو

المطلوبة. (لاحظ أن مايكروسوفت فيزو ٢٠٠٣) لا تستخدم آخر معايير الملاحظة لعلاقة الامتداد أو التوسع. المعيار الحالي من UML به خط محطم ورأس سهم منتظم مشيران إلى الاتجاه المعاكس كما هو موضح في الشكل (أ- ١).

يبين الشكل أ-٢ مخطط حاله استخدام لـ هوسر برجر، الممثل عميل بدأ في حالة استخدام لأمر طعام؛ الممثل الآخر المتضمن في الموضوع هو شخص الخدمة. سيدل سيناريو خاص علي العميل الذي يضع أمر مع خدمة الشخص.

شكل أ-٢ مخطط حالة الاستخدام لنظام هوسر برجر.



حتى الآن لقد رأيت نوع واحد من العلاقة والامتدادات بين حالات الاستخدام. نوع آخر من العلاقة الشامل في الأمر، الذي ينشأ عندما ترجع حالة الاستخدام الأولى إلى حاله استخدام أخرى. العلاقة االشاملة هي أيضا مبينة بشكل تخطيطي كخط محطم برأس منتظمة تشير تجاه حالة الاستخدام المستخدمة؛ يتم تصنيف الخط برمز «يشمل». في الشكل أ-٢ على سبيل المثال، العلاقة الشاملة بين الإمدادات بإعادة الطلب للمبيعات وحالات استخدام بيانات القائمة المفصلة التي تدل على أن الاستخدامات السابقة أحدث عهدا عند تنفيذها. ببساطة يضعها، عندما يقوم المدير بالإمداد بإعادة الطلبات، يتم اقتفاء أثر المبيعات وبيانات القائمة المفصلة. يتم أيضا اقتفاء أثر نفس البيانات عندما تنتج تقارير الإدارة، لذلك هناك علاقة شاملة أخرى بين تقارير إدارة المنتجات والمبيعات المتعقبة وحالات استخدام بيانات القائمة المفصلة. والمبيعات المتعقبة وحالات استخدام بيانات القائمة المفصلة هي حالة استخدام معممة، لتدل على السلوك المشترك بين حالات الاستخدام المتخصصة، وإمدادات إعادة الطلبات، وتقارير إدارة المنتجات. عندما يتم القيام بإمدادات إعادة الطلبات، وتقارير إدارة المنتجات، ويتم استخدام تقارير إدارة المنتجات والمبيعات المتعقبة الكاملة.

نمذجة الكائن
object modeling

مخططات الصنف (class diagram) في طريقه التصميم الكائنية. نحن ننمذج العالم ككائنات. **الكائن (an object)** هو الكينونة التي له دور في التعريف في نطاق التطبيق وحالته وسلوكه وتجانسه. الكائن هو مبدأ أو فكرة تجريدية أو الكائن الذي يصنع الحس في التطبيقات النطاقية. الكائن من الممكن أن يكون ملموس أو مرئي المدخل (مثل الشخص، المكان، أو شي) ويمكن أن يكون مبدأ أو حادثة (مثل الاقسام، تاديه، زواج، تسجيل،....) أو يكون نتاج صناعي من معالجات التصميم (مثل واجه المستخدم، المتحكم، جدولة،....). لديه حاله وسلوك يظهر بين العمليات التي تستطيع أن تختبر أو تدرس حالته. **حالة (the state)** الشرط الذي يجمع خصائص الكائن (صفات، علاقات) وقيم هذه الخصائص تحتوي على سلوكه **(behavior)** الذي يمثل كيف فعل ورده فعل الكائن. حالة الكائن تحدد عن طريق قيم صفاته وعلاقته مع الأشياء الأخرى. سلوك الكائن يعتمد على حالته والعمليات المكملة. العمليات هي عمل واضح الذي اشتغل فيه الكائن واحد تلو الآخر بشكل منظم لكي يحصل على استجابه. لو اخذنا بعين الاعتبار مثال الطالب، ماري جونز تمثل الكائن. وحالة هذا الكائن تشخص عن طريق صفاته، الاسم، تاريخ الميلاد، السنه، عنوان وهاتف، وقيم هذه الصفات تكون في الوقت الحاضر. على سبل المثال الاسم هو ماري جونز والسنه هي الثانوية، وهكذا. وسلوك الكائن هو يتضح خلال العمليات كحساب المعدل الذي يستخدم لحساب معدل الطالب الحالي. ماري جونز شي بسبب جمعها حالتها وسلوكها معا.

كل الأشياء لديها تعرف خاص بها: والذي لا يمكن أن يكون شيئين لديهم نفس التعريف. على سبيل المثال لو كان هناك طالبان متماثلان بالاسم وتارخ الميلاد فهذان الطالبان حقيقتا شيئين مختلفين. كذلك لو كان المثالان متطابقان في جميع قيم الصفات، الأشياء تستطيع تمييز التماثل. وفي نفس الوقت، الاشياء تبقي التماثل الخاص بها على مدى الحياة. على سبيل المثال لو تزوجت ماري جونز وتغير اسمها، عنوانها وهاتفها، سوف تبقى تمثل بنفس الكائن.

تستطيع أن توضح **صنف الكائن (object class)** (وضع الكائن التي تشترك في الهيكل والسلوك) بمخطط بياني في **مخطط الصنف (class diagram)** كما في المخطط ٣-أ (أ). مخطط الصنف يظهر الهيكل الثابت للتصميم الكائنية: أصناف الكائن، الهيكل الداخلي له، والعلاقات فيها التي تشترك معها. في UML، الصنف يمثل بمستطيل مقسم لثلاثة أقسام بخطوط أفقيه. اسم الصنف يظهر القسم الأعلى، وقائمة الصفات في الجزء الأوسط، وفي الجزء الأخير تكون العلاقات. المخطط يظهر صنفان، الطلاب والمقررات، ويظهر صفاتهم

الكائن
Object
هو الكينونة التي لها دور في تعريف نطاق التطبيق وحالته وسلوكه وتجانسه.

الحالة
State
الشرط الذي يجمع خصائص الكائن (صفات، علاقات) وقيم هذه الخصائص.

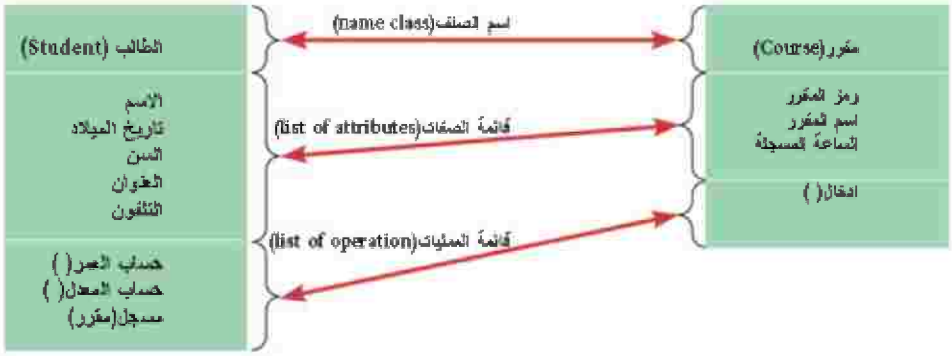
السلوك
Behavior
هو الطريقة التي تمثل كيفية فعل وردات فعل الكائن.

مخطط الصنف
Class diagram
هو المخطط يظهر الهيكل الثابت للتصميم الكائنية: أصناف الكائن، الهيكل الداخلي له، والعلاقات التي تشترك معها.

السلوك
Behavior
هو الطريقة التي تمثل كيفية فعل وردات فعل الكائن.

وعملياتهم. الأشياء تنتمي إلى نفس الصنف وتشترك معها في نفس العلاقات مع أشياء أخرى. على سبيل المثال كل الطلبة يسجلون في مقررات وبناءا على ذلك صنف الطالب يستطيع أن يشترك في علاقة تدعى تسجيل في مع صنف أخر يدعى مقرر.

شكل أ-٣ (أ) مخططات الصنف والشيء - مخطط الصنف يظهر صنفين.



شكل ٣-أ (ب) مخططات الصنف والشيء - مخطط الشيء بمثالين.

مخطط المثال
Object diagram
هو عبارة عن رسم لكائنات متوافقة مع شكل الفئة وتسمى شكل كائن.

مخطط الكائن، أيضا تعرف ب **مخطط المثال (instance diagram)**، وهو مخطط للمثال الذي ينسجم مع مخطط الصنف المعطى. في المخطط ٣-أ (ب) يظهر لدينا مخطط الكائن مع مثالين، واحد لكل صنفين الذين يظهران في المخطط ٣-أ (أ) مخطط الكائن الثابت هو مثال لمخطط الصنف، على أنه صورة مفصلة لنظام في نقطة معينة من الوقت. في مخطط الكائن، الكائن يمثل كمستطيل مقسم لجزأين. أسماء الكائن وصنفه ويضع تحتهم خط ويظهر في الجزء الأعلى من مستخدما الصيغة التالية: اسم الكائن: اسم الصنف. وصفات الأشياء وقيمها تظهر في الجزء الثاني من المستطيل. على سبيل المثال، لدينا الكائن يدعى ماري جونز، والمنتمي لصنف الطالب. وقيم الاسم، تاريخ الميلاد، والسنة صفات أيضا ظاهرة.

العمليات
Operation
الوظيفة أو الخدمه التي تشترط كل أمثلة الصنف.

العمليات (operation) مثل حساب المعدل لطالب (انظر المخطط ٣-أ (أ)) هي عملية أو خدمة التي تشترط كل أمثلة الصنف. هي فقط من بين العمليات التي تستطيع

الأشياء الوصول إليها وتغير المعلومات المخزنة في الكائن. لذلك العمليات تشترط واجهة خارجية لصنف، الواجهة تعرض لنظهره الخارجية لصنف بدون عرض الهيكل الداخل أو كيف تنفذ العمليات. تقنية إخفاء تفاصيل التنفيذ الداخلي للشي من العرض الخارجي تعرف **بالإحتوائية (encapsulation)** أو إخفاء المعلومات (information hiding).

الإحتوائية
Encapsulation
هي تقنية لاخفاء تفاصيل التنفيذ الداخلي للشي من الخارج.

تمثيل الارتباط
Representing Associations

الارتباط هي علاقة بين أصناف الكائن. كما في نمذجة (E-R model) درجه الارتباط بين العلاقة قد تكون واحد (unary) أحادية، اثنين (binary) ثنائية، ثلاثة (ternary) ثلاثية، أو أعلى (n-aryl)، كما هو موضح في المخطط ٤-أ. الارتباط تمخطط بخط عريض بين الأصناف المشتركة. في نهاية الارتباط حيث تصل إلى الصنف تدعى **دور الارتباط (association role)**. الدور قد يحدد بشكل واضح مع علامة قرب نهاية الارتباط (انظر دور المدير في الشكل أ-٤(أ). اسم الدور يشير إلى الملعوب بواسطة صنف المتعلق بالنهاية قرب الاسم الظاهر. على سبيل المثال دور المدير من جهة أداره العلاقة يشير ضمنيا إلى أن الموظف يمكن أن يلعب دور المدير. كل دور له **تعدد (multiplicity)**، ويشير إلى العديد من الأشياء في علاقة الارتباط المسموحة. في مخطط الصنف مواصفات التعددية تظهر في سلسلة نصية تمثل فترة الإعداد الصحيحة كما في الصيغة التالية: الحد الأدنى....الحد الأقصى. الفترة تؤخذ على أساس أنها مغلقة، الذي يعني أن المدى يشمل كلا من الحدود الدنيا والحدود القصوى. بالإضافة إلى القيم الصحيحة، الحد الأقصى من التعددية يكون برمز النجمة (*)، والذي يدل على حد أقصى لا نهائي. إذا حددت إشارة القيمة الصحيحة فإن ذلك يعني أن المدى يحتوي فقط على تلك القيمة.

الارتباط
Association
العلاقة بين أصناف الكائن.

دور الارتباط
Association role
نهايه الارتباط عند الاتصال بالصنف

التعددية
Multiplicity
تبين عدد الأشياء المشتركة في العلاقة المعطاة

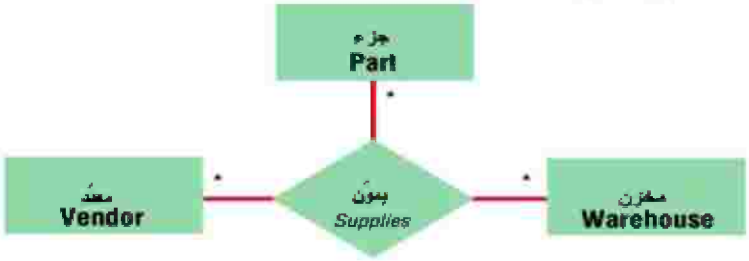
التعدد الشائع عمليا هو ٠..١، ٠..*، و١..١. التعددية ٠..١ تشير إلى الأدنى هو ٠ والأقصى هو ١ (الاختياري واحد)، بينما * (أو المكافئ، ٠..*) يمثل المدى من صفر إلى مالانهاية (الاختياري متعدد). الاشارة ١ يمثل ١..١، دلالة على أنه لابد أن يشترك شي واحد في العلاقة (إلزامي).

الشكل أ-٤(أ) يوضح ثلاث علاقات ثنائية: مخصص (واحد إلى واحد)، ويحتوي (واحد إلى متعدد) ويسجل (متعدد إلى متعدد). المثلث العريض بجانب اسم الارتباط يظهر الاتجاه في قراءه الارتباط.

شكل أ-٤(أ) أمثلة لعلاقات الترابط لدرجات مختلفة - أحادي.

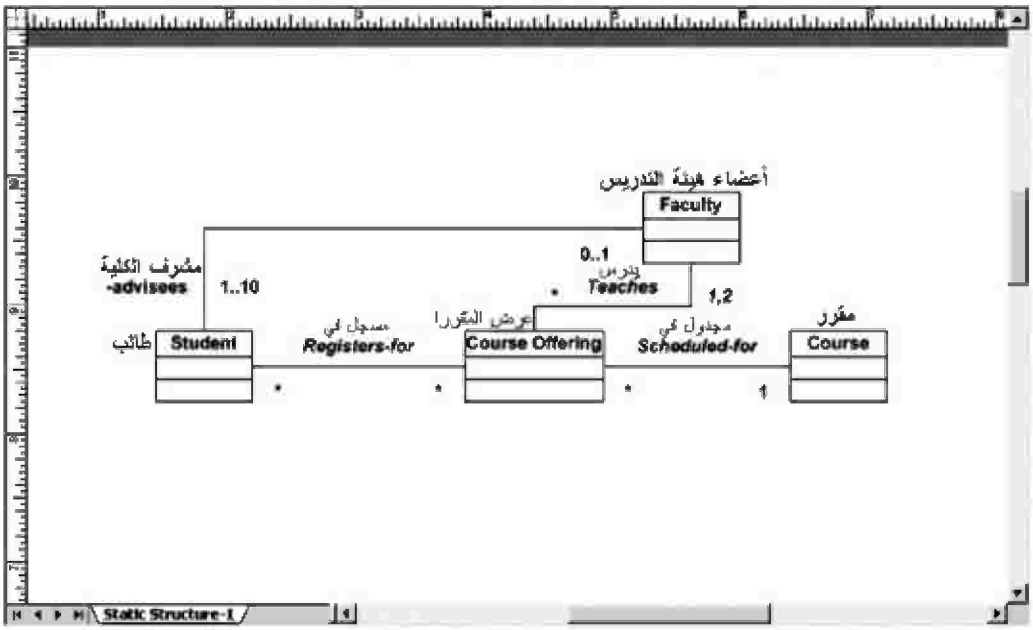
شكل أ-٤ (ب) أمثلة لعلاقات الترابط لدرجات مختلفة - ذاتي.

شكل أ-٤ (ج) أمثلة لعلاقات الترابط لدرجات مختلفة - ثلاثية.



على سبيل المثال، الارتباط المتضمن يقرأ من خط الإنتاج إلى المنتج. الشكل أ-٤ (ج) يوضح العلاقة الثلاثية بين الممول، البائع، القطعة والمستودع. كما في مخطط E-R، نمثل العلاقة الثلاثية بالشكل المعين ووضع اسم العلاقه داخله. مخطط الصنف في الشكل أ-٥ (أ) (معروف كاجدول هيكل ثابت في ميكروسوفت فيزو) يظهر ارتباط ثنائي بين الطالب والكلية، بين المقرر و المقرر المعروض، الطالب والمقرر المعروض، وبين المقرر المعروض والكلية. المخطط يوضح أن الطالب يجب ان يكون لديه مرشد، حيث عضو الكلية يجب أن يكون مرشد أكثر من عشر طلاب، أيضا، على الرغم من أن المقرر يجب أن يكون معروض، يعطى المقرر المعروض المجدول لمقرر واحد فقط. لاحظ عضو الكلية قد يلعب دورين: مرشد وموجة. الشكل أ-٥ (ب) يعرض مثال اخر لمخطط الصنف، الذي هو للطلب الزبون، مستخدما معيار UML.

شكل أ-٥ أمثلة لعلاقات الارتباط الثنائية مثال الجامعة (جدول هيكل ثابت في ميكروسوفت فيزو).



تمثيل التعميم
Representing Generalization

في الطرق الكائنية التوجه تستطيع أن تجرد الصفات الشائعة (صفات وعمليات) بين الأصناف المتعددة بينما العلاقات التي تشترك في أكثر من صنف عام تسمى (generalization التعميم)، والأصناف التي تعمم تسمى الصنف الجزئي والأصناف التي تعمم في الداخل تسمى الصنف الكلي.

تمعن المثال الموضح في الشكل أ-٦(أ) هناك ثلاث أنواع من الموظفين وهم موظفي الساعات وموظفي الرواتب والمستشارين. الصفات التي تشترك بواسطة كل الموظفين هي رقم الموظف واسمه وعنوانه وتاريخ الاستئجار والعلامة الملصقة تخزن في الصنف الكلي للموظفين، بينما المعالم الغريبة بالذات نوع الموظف تخزن في الصنف الكلي ا لمماثل. على سبيل المثال (معدل الساعات وحساب الأجر من موظفي الساعات) طرق التعميم موضحة في خط مصمم من الصنف الجزئي إلى الصنف الكلي مع سهم أجوف في النهايه منه. بالإشارة نحو الصنف الكلي. تستطيع أن توضح مجموعة من طرق التعميمات لأجل الصنف الجزئي كشجرة متعددة الفروع تربط الصنف الجزئي الفردي مع القسم المشرك مع سهم رأسي أجوف للإشارة نحو الصنف الكلي في الشكل مثلا لدينا مقارنة طرق التعميم من المتعالج الخارجي إلى المريض ومن المتعالج المقيم إلى المريض في داخل القسم المشترك مع سهم رأسي للإشارة نحو المريض. تستطيع أن تشير إلى قاعدة التعميم بواسطة تحديد التفرقه بجانب الطريق. التفرقة توضح ملكية كائنية الصنف. ذلك أن تكون مجردة عن طريق اشتراك علاقة التعميم. على سبيل المثال في الشكل أ-٦(أ) نفرق صنف الموظف على أساس نوع التوظيف (ساعات وذو مرتب ومستشار).

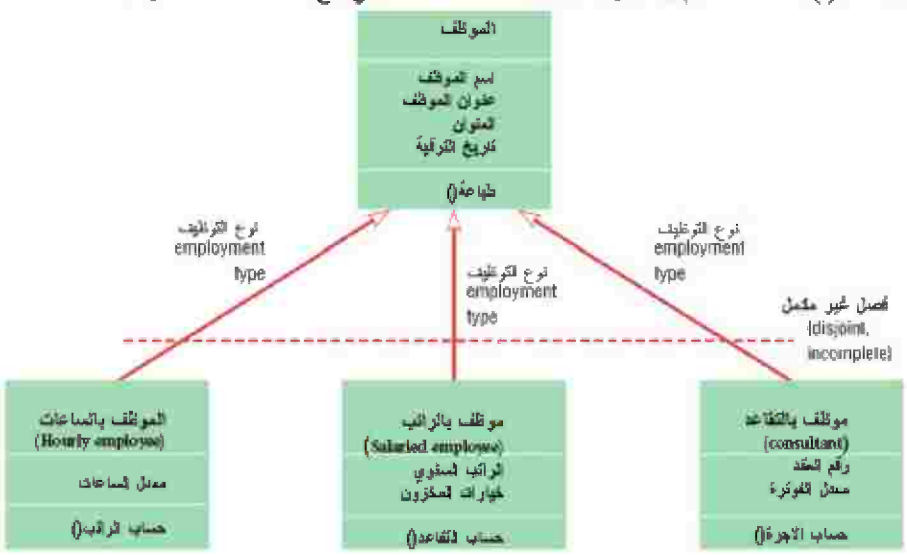
الصنف الجزئي يرث كل المعالم من الصنف الكلي على سبيل المثال بالإضافة أن المعالم الخاصة (معدل الساعات وحساب الأجر السنوي) الصنف الجزئي لموظف الساعات يرث اسم الموظف ورقمه وعنوانه وتاريخ الاستئجار والعلامة الملصقة من الموظف مثل موظفي الساعات سوف تخزن قيم من أجل صفات الموظف وموظف الساعات وعند الطلب سوف تطابق الملصق الخاص وعمليات حساب الأجر السنوي.

الوراثة واحدة من الميزات الرئيسية لاستخدام طرق الكائنيت التوجه. إنها تسمح باعادة كتابة الكود لذلك لا تحتاج مبرمج لكتابة الكود بل هي مكتوبة مسبقا في الصنف، المبرمج يكتب فقط الكود الجديد المنقاه من الصنف الكلي لايجاد الصنف. مقترحوا طرق الكائنيت التوجه يدعون أن نتائج اعادة استخدام الكود في معدل الإنتاج يكسب عدة أوامر في غاية الأهمية الشكل أ-٦(ب) صنف المريض المكتوب بالحروف المائلة يشير إلى **الصنف التجريدي abtract class** هو صنف ليس حالة مباشرة وإنما ينحدر من حالات مباشرة. الأصناف التي لديها حالات مباشرة هي (المتعالج الخارجي - المتعالج المقيم) تدعى **concreate class**. صنف المريض التجريدي يشترك في علاقة تدعى المعالجة من قبل الطبيب تدل إلى أن كل المرضى ومرضى الخارج والمرضى المقيمين مثل المعالجين بواسطة الطبيب بالإضافة إلى أن علاقة الوراثة لصنف المريض المقيم لديها علاقة خاصة وحيدة تسمى مخصص للسرير تشير أن المرضى المقيمون يمكن أن يتخصصوا السراير. القيود السيمانطيقيه بين الصنف الجزئي تستطيع أن تعبر عنها باستخدام (الاكمال وغير الاكمال والقرار و overlapping). الاكمال يعني في كل الحالات يجب أن تكون الحالة الصنف الجزئي إما عدم الاكمال يعني يمكن أن تكون الحالة من الصنف الكلي وإما القرار لا يمكن لأي حالة أن تكون أكثر من الصنف الجزئي في نفس الوقت و overlaping. تسمح المشاركة المتزامنة في عدة الصنف الجزئي.

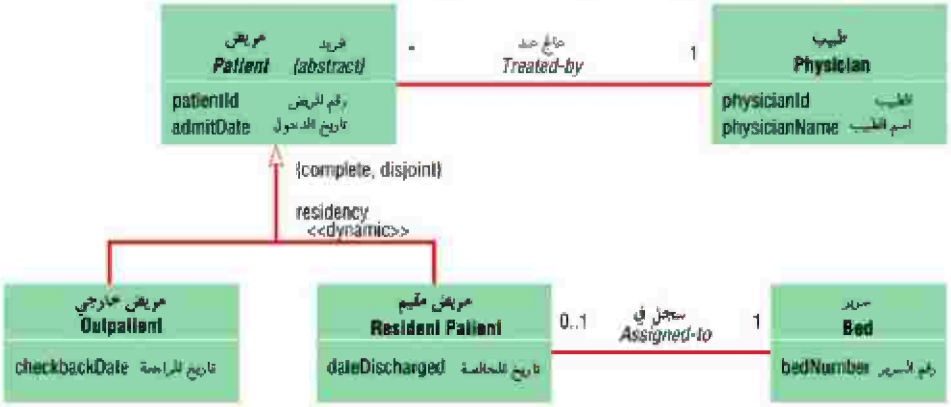
الصنف التجريدي
Abstract class
هو صنف ليس حالة مباشرة ولكن ينحدر من حالات أخرى.

الصنف المركب
Concrete class
هو صنف حالة مباشرة

شكل أ-٦(أ) أمثلة التعميم والتوريث وتقيد الصنف الأساسي مع ثلاثة أصناف فرعية.



شكل أ-٦(ب) أمثلة التعميم وتقييد تجريد صنف المريض مع صنفين فرعيين.



تمثيل التجميع

Representing Aggregation

التجميع
Aggregation
عبارة عن جزء من العلاقة بين أجزاء الكائن.

التجميع (aggregation) عبارة عن جزء من العلاقة بين أجزاء الكائن ومجموعة أشياء أخرى وهي أقوى شكل للعلاقة المشتركة. وتمثل بمعين أجوف في نهاية المجموعة.

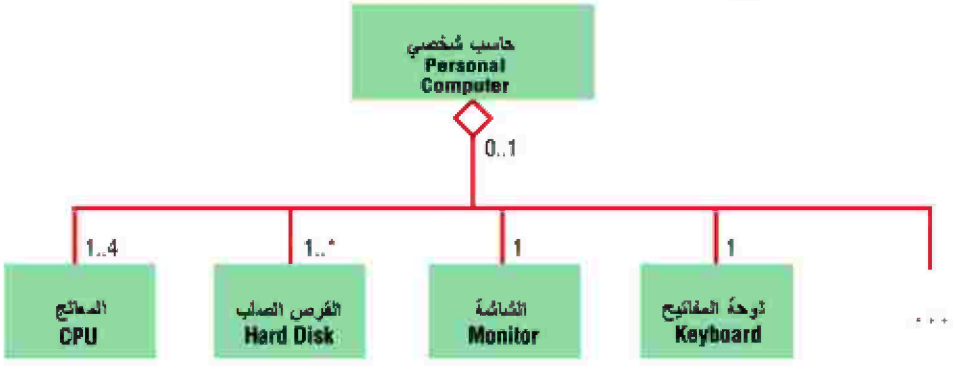
على سبيل المثال يوضح الشكل أ-٧ حاسب شخصي كمجموعة معالج (CPU) والقرص الصلب والشاشة لوحة المفاتيح وأشياء أخرى.

لاحظ التجميع يتطلب مجموعة من حالات الكائن الواضحة، واحد منها يحتوي أو يتكون من الأخرين.

على سبيل المثال شيء الحاسب الشخصي متعلق شيء المعالج، واحد من هذه الأشياء يقارن التعميمات المتعلقة بشيئيت الصنف.

الكائن هو بنفس الوقت حالة من هذه الأصناف (مثل الطالب الخريج) والصنف الكلي(مثل الطالب).

شكل أ-٧ مثال التجمع.



النمذجة الديناميكة: مخططات الحالة
Dynamic Modeling:State Diagrams

في هذا القسم نوضح كيف تشكل المظهر الديناميي من المخطط المنظوري للانتقال الرسومي. في UML التنقلات الرسومية توضح استخدام التخطيط الرسومي. التخطيط الرسومي يصور الانتقالات الرسومية المختلفة أو تبديل الكائن يمكن أن يواجه أثناء دورة حياته سوية مع الأحداث المسببة لهذه التنقلات. الحالة شرط أثناء حياة الكائن الذي يستوفي بعض الشروط التي تؤدي بعض العمل أو ينتظر بعض الحدث التغييرات المخططية عندما يستلم الكائن بعض الحدث؛ هي أن الكائن ينقاد للمرور بلانتقال الرسومي الحالة الكائنية تعتمد على قيم الصفات وربطها مع الأشياء الأخرى. An event (الحدث) بعض الأحيان يأخذ مكان في النقطة المؤكدة للوقت أنها ملاحظة جديرة أن تسبب انتقال رسومي مثل (أمر أماكن الزبون، تسجل الطلاب في قسم، شخص يطابق لوحده، أجر الشركة لموظف جديد) الغرض الرئيسي للعرض هو أن الحدث يعتبر آني. في الحقيقه يمكن أن يأخذ بعض من الوقت بقايا الكائن في الحالة المشتركة لبعض الوقت قبل التنقلات إلى حالات أخرى على سبيل المثال كائن الموظف يمكن أن يكون حالة من الوقت الجزئي (تحديد صفات التوظيف) لأشهر قليلة قبل التنقلات لحالة الوقت الكلي المستندة على توصية من المدير. في UML الحالة توضح كمستطيل مدور الزوايا في الشكل أ-٨ على سبيل المثال لدينا توضيح لحالات مختلفة من كائن الطالب مثل الاستعلام والمطابقة والتحسن. التخطيطات الرسومية توضح كيفية التنقل من الحالات الأولية إلى الحالات الأخرى عند تأكيد الحدث تحصل أو عندما الشروط تحدث، عندما كائن طالب ينشأ هذه حالة بدائية، الحالات البدائية مثل الاستعلامات والتنقلات، عندما يكون الطالب في حالة الاستعلام يعد طلب لحالة التطبيق الكائن ينقله لحالة التطبيق، التنقل موضح كسهم مصمم من الاستعلام إلى التطبيق ملصق مع الاسم وتقديم الطلب. التنقل يمكن أن يكون ملصق بحروف متضمنة اسم الحدث وبرامترات الحدث وشرط الحدث التنقل ليس لديه خاصية توضيح لكل العناصر، هو فقط يوضح التنقلات الخاصة بالموضوع على سبيل المثال نربط التنقل من حالة الاستعلام إلى حالة التطبيق مع سهولة اسم الحدث لكن التنقل من حالة التطبيق إلى حالة التحسن نوضح اسم الحدث والشرط المقيد.

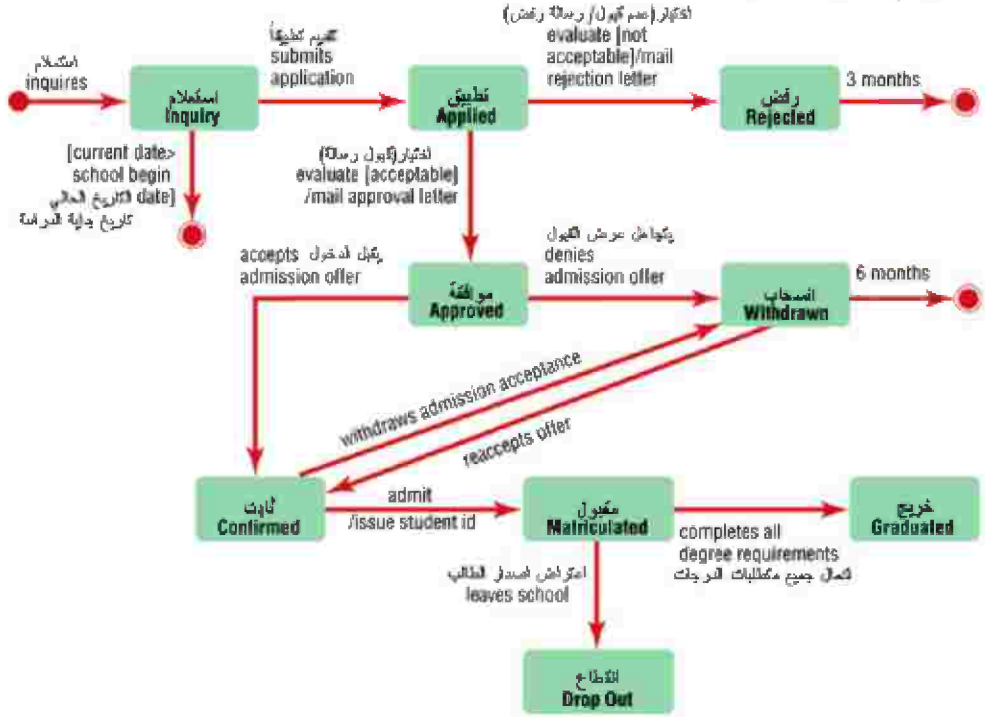
حدث أخذ رسالة موافقة بريدية يعني أن الطلب يحدد القبول إذا كان مكتب القبول عمل تقيم للمتقدم إذا قبل ترسل رسالة موافقة بريدية وإلا يرسل رسالة رفض. الشرط المقيد يوضح بمربع أقواس والحدث يوضح بشعار ((/)). بقيت الأحداث للثلاثة أشهر

حاله التنقل
State transition
هي تغيير في صفات الكائن أو ربط الأشياء بأشياء أخرى.

الحدث
Event
بعض الحيان يأخذ مكان في النقطة المؤكدة للوقت أنها ملاحظة جديرة أن تسبب انتقال رسومي.

قبل وصول الحالة الأخيرة في التخطيط سوف نوضح حدوث الوقت المنقضي للثلاثة أشهر. الحالة الأخيرة توضح كدائرة محاطة بدوائر أخرى بعد التنقل للحالة الأخيرة شيء الطالب يقترب من الخروج. لاحظ شيء الطالب يمكن أن ينتقل إلى الحالة الأخرى من الحالتين الأخريتين (الاستعلام – الانسحاب). من أجل الاستعلام نحن لا نستطيع أن نحدد (اسم أو فعل)، لاكن لدينا توضيح الشرط المقيد. هذه الأحداث تشير إلى أن شيء الطالب يتوقف ليخرج بعد أول يوم من الدراسة ما لم يكون الفصل الدراسي والكائن تحرك في الوقت المعين من حالة الاستعلام إلى الحالات الأخرى. التخطيط الرسومي الموضح في الشكل أ–٨ يلتقط كل الحالات الممكنة لكائن الطالب وحالة التنقلات و الحدث أو الشرط الذي يسبب التنقلات. من أجل نوع الطالب أنها تلتقط زيارة الطالب من خلال الكلية عندما يعبر عن رغبته في الكلية حتى التخرج.

شكل أ–٨ حالة الرسم الشبيهة للطلاب.



النمذجة الديناميكية: المخططات التسلسلية
Dynamic Modeling: Sequence Diagrams

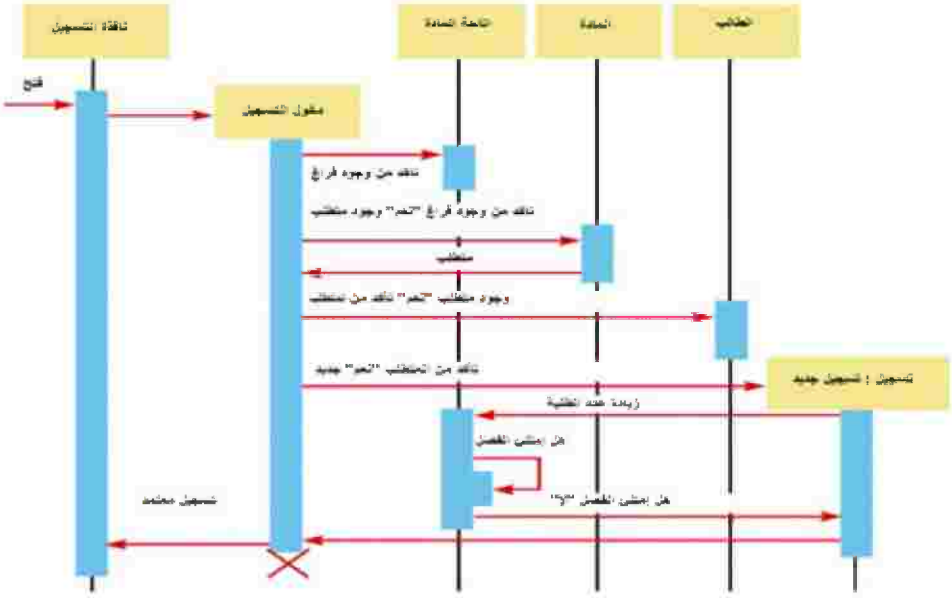
في هذا الفصل نبين كيفية تصميم بعض الحالات التي عرفناها سابقا في مرحلة التصميم. حالة التصميم العملية تظهر كيف تنفذ هذة الحالات بمجموعة من الكائنات الرابطة. في لغة النمذجة الموحدة (UML)، يستخدم مخطط تفاعلي لإظهار النموذج للترابطات بين الكائنات في حالة عملية معينة. يوجد نوعين من الرسوم التفاعلية: الرسوم التسلسلية والرسوم المتشاركة. نظهر لك كيف نصمم حالات التصميم باستخدام الرسوم المتسلسلة.

مخطط تسلسلي
Sequence diagram
وصف للترابطات بين الكائنات خلال فترة معينة من الزمن

المخطط المتسلسل Sequence Diagram يصف الترابطات بين الكائنات خلال فترة من الزمن. ولأن نموذج الترابطات يتغير من حالة إلى أخرى، كل مخطط تسلسلي يقوم فقط بإظهار الترابطات المتعلقة بكل حالة على حده. يقوم بإظهارالكائنات المشاركة بخطوط حياتها والترابطات بين هذه الكائنات – مرتبة بالتسلسل الزمني – بالرسائل التي تتبادلها في ما بينها.

الشكل أ–٩ يظهر مخطط تسلسلي لسيناريو تسجيل طالب لمادة تتطلب مادة أو أكثر سبق دراستها. المحور العمودي من المخطط يمثل الوقت، والمحور الأفقي يمثل الكائنات المترابطة المختلفة. يتزايد الوقت كلما نزلنا مع المحور العمودي. يحتوي الشكل على ستة كائنات، بدئا من نافذة للتسجيل في اليسار، إلى التسجيل الجديد في اليمين تسجيل جديد. يظهر كل كائن كخط عمودي يسمى خط الحياة، يمثل خط الحياة وجود كائن في فترة معينة من الزمن. رمز الكائن – مستطيل مع خط تحت اسم الكائن – يوضع في مقدمة كل خط حياة.

شكل أ–٩ رسم تسلسلي لتسجيل مادة مع متطلبات.



التفعيل
Activation
الفترة الزمنية التي يؤدي فيها الكائن وظيفة.

يوضع مستطيل رفيع على خط حياة الكائن ويمثل تفعيل الكائن. **التفعيل activation** يظهرالفترة الزمنية التي يؤدي فيها الكائن وظيفة، مباشرة أو من خلال وظيفة تابعة. قمة المستطيل التي تقع عند علامة الرسالة القادمة، تشير الى بدء التفعيل، والقاع يشير إلى الإتمام.تتواصل الكائنات مع بعضها البعض بإرسال الرسائل، تظهر الرسالة كسهم من المرسل إلى المستقبل، مثال: رسال تأكد من وجود فراغ تمثل كسهم من دخول التسجيل إلى

إتاحة المادة. قد تكون الرسائل من أنواع مختلفة. كل نوع يميز في المخطط برأس سهم معين. **الرسالة المتزامنة Synchronous Message** التي تظهر كسهم محكم هي الرسالة التي ينتظر المنادي للكائن المستقبل ليتم تنفيذ الوظيفة المناداة قبل أن يواصل هو التنفيذ. مثال على الرسالة التزامنية رسالة تأكد من وجود فراغ. عندما يرسل كائن تسجيل هذة الرسالة إلى كائن إتاحة المادة، يتأكد كائن الإتاحة من وجود وجود فراغ بتنفيذ عملية تأكد من وجود فراغ (نفس اسم الرسالة). بعدما يتم تنفيذ هذة العملية، يعود التحكم إلى عملية التسجيل لإعطاء قيمة «نعم» أو «لا». الرسالة التزامنية دائما يرتبط معها رسالة رد . قد ترجع الرسالة للمنادي ببعض القيم أو ببساطة تبلغه بأن العملية التي قد طلبها قد تمت بنجاح. لم نظهر ما يرجع من رسالة تأكد من وجود فراغ؛ لأنه ضمني. لقد أظهرنا صراحة المرتجع من رسالة وجود متطلب من دخول التسجيل إلى المادة. ذيل الرسالة المرتجعة مصطف مع أساس مستطيل التفعيل لعملية وجود متطلب. **الرسالة البسيطة Simple Message** ببساطة تنقل التحكم من المرسل إلى المستقبل بدون أن تظهر تفاصيل الإتصال. يظهر رأس السهم للرسالة البسيطة في المخطط كعلامة عرض. كما رأينا، فإن الرسلة التزامنية ترجع رسالة بسيطة. رسالة فتح في الشكل أ-٩ هي أيضا رسالة بسيطة، ببساطة تقوم بنقل التحكم إلى *نافذة التسجيل*.

الرسالة التزامنية
Synchronous message
نوع من الرسائل حيث ينتظر المنادي للكائن المستقبل ليتم تنفيذ الوظيفة المناداة قبل أن يواصل هو التنفيذ.

الرسالة البسيطة
Simple message
تنقل التحكم من المرسل إلى المستقبل بدون أن تظهر تفاصيل الاتصال.

تصميم حالة الاستخدام مع المخطط التسلسلي
Designing a Use Case with a Sequence Diagram

لنر كيف نصمم حالة عملية. سوف نخطط مخطط تسلسلي لمثال instance عن حالة عملية لتسجيل المادة، في هذة الحالة تتطلب المادة مواد أخرى. هنا وصف لهذا السيناريو:

١- يقوم المسجل بفتح نافذة التسجيل ويدخل معلومات التسجيل (الطالب والفصل).

٢- يتأكد إذا كان الفصل مفتوح.

٣- إذا كان الفصل مفتوح يتأكد إذا كان للمادة أية مواد متطلبة.

٤- إذا كان للمادة متطلبات، يتأكد إذا كان الطالب قد درس هذه المواد.

٥- إذا كان الطالب قد درس هذه المواد، إذن سجل الطالب في هذه المادة، وأضف واحد لحجم الفصل.

٦- تأكد إذا كان الفصل قد إمتلئ، إذا لم يمتلئ، لا تفعل شيئ.

٧- إعرض التسجيل المعتمد في نافذة التسجيل.

في ردة فعل لرسالة «فتح» من المسجل (الممثل الخارجي)، تظهر نافذة التسجيل على الشاشة، ثم تدخل بيانات التسجيل. هذا ينتج كائن لدخول التسجيل، الذي يرسل رسالة تأكد من وجود فراغ للكائن المعطى (الذي يمثل الفصل الذي يريد الطالب التسجيل فيه). هناك احتمالان لما قد يرجع: نعم أو لا. في هذا السيناريو، نفترض أن الفصل مفتوح . لذلك وضعنا شرط حامي Guard Condition، تأكدمن انه مفتوح = «نعم»، في رسالة وجود متطلب. الشرط الحامي يتأكد من ان الرسالة سوف ترسل فقط عندما يكون الفصل مفتوح. القيمة المرجعة هي قائمة بالمتطلبات، القمية المرجعة تظهر صراحة في المخطط. لهذا السيناريو، حقيقة وجود متطلبات تلتقط بالشرط الحامي، وجود متطلبات = «نعم». إذا تحقق هذا الشرط يرسل كائن دخول التسجيل رسالة تأكد من وجود متطلبات، تحتوي على المتطلب، لكائن الطالب لتتأكد من أن الطالب قد درس هذة المواد. إذا كان الطالب قد درس هذة المتطلبات، يقوم كائن دخول التسجيل بإنتاج كائن جديد يطلق عليه تسجيل جديد، الذي يمثل التسجيل الجديد. بعد ذلك، يقوم التسجيل الجديد بارسال رسالة تسمى زد عدد الطلبة لإتاحة المادة لزيادة حجم الفصل بواحد، تقوم رسالة زد عدد الطلبة بمناداة هل امتلئ الفصل، عملية أخرى في نفس الكائن. بافتراض أن الفصل غيرممتلئ، تقوم عملية هل إمتلئ الفصل باعادة التحكم للعملية المنادية بقيمة «لا». بعد ذلك عملية زد عدد الطلبة تكتمل وتتنازل عن التحكم للعملية المنادية في تسجيل جديد.

أخيراً، في وصل للرسالة القادمة من التسجيل الجديد، يقوم كائن التسجيل بتدمير نفسه (يظهر التدمير بعلامة X) ويرسل تأكيد عن التسجيل لنافذة التسجيل. لاحظ أن دخول التسجيل ليس كائن ثابت، أي أنه يتكون أثناء التسجيل ليتحكم في تتابع التداخلات ويحذف عندما ينتهي التسجيل. أثناء وجوده، يقوم باستدعاء عدة عمليات في كائنات أخرى بإرسال هذة الرسائل تتابعيا: تأكد من وجود فراغ، وجود متطلبات، تأكد من وجدود متطلبات، وجديد .بخلاف كائن دخول التسجيل، ينشئ «تسجيل جديد» أيضا خلال الفترة الزمنية الملتقطة في المخطط. الرسائل التى تنشئ هذه الكائنات تمثل بأسهم تؤشر مباشرة على رموز هذة الكائنات. كمثال، السهم الذي يمثل الرسالة جديد متصل برمز الكائن للتسجيل الجديد. خط الحياة لمثل هذا الكائن يبدأ عندما تستقبل الرسالة التي تكونه (يختفي خط الحياة المتقطع خلف مستطيل التفعيل).

التحرك إلى التصميم
Moving to Design

عندما تتحرك إلى التصميم، تبدأ بالمجموعة المتواجدة من نماذج التصميم Design Models وتستمر في إضافة تفاصيل فنية. كمثال، قد تضيف عدة واجهات لمخطط التسجيل لنمذجة النوافذ التي ستنفذها لاحقا باستخدام أداة تطويرواجهة المستخدم الصورية GUI مثل Visual Basic أو Visual C++. سوف تعرِف جميع العمليات بالتفصيل، توضِح الإجراءات والتوقيعات والقيم الراجعة بشكل كامل. إذا قررت استخدام قاعدة بيانات علاقية DBMS ستحتاج ادخال وتنظيم الكائنات والعلاقات في جداول ومفاتيح رئيسية ومفاتيح أجنبية. النماذج المحدثة في مرحلة

التصميم ستكون مفصلة أكثر بكثير من نماذج التحليل Analysis Models. يظهر الشكل أ–١٠ عمارة لثلاث طبقات، تتكون من حزمة واجهة المستخدم، حزمة الكائنات العملية، وحزمة قاعدة البيانات. تمثل هذه الحزم أنظمة فرعية مختلفة لنظام معلوماتي. الخطوط المتقطعة تمثل الترابطات بين هذه الحزم. كمثال، حزمة واجهة المستخدم تعتمد على حزمة الكائنات العملية، تشارك الحزم في علاقة عميل– مزود. إذا قمت بتغييرات في بعض الكائنات العملية، الواجهات (الشاشات) قد تتغير نتيجة لذلك. تتكون الحزمة من مجموعة من الطبقات. الطبقات التي داخل الحزم متلاصقة. بمعنى أنها مقترنة بشدة. الحزم نفسها يجب أن تكون مقترنة بسلاسة، لكي لا تتأثر حزمة بالتغيير في حزمة أخرى بشكل كبير. في العمارة التي في الشكل أ–١٠.

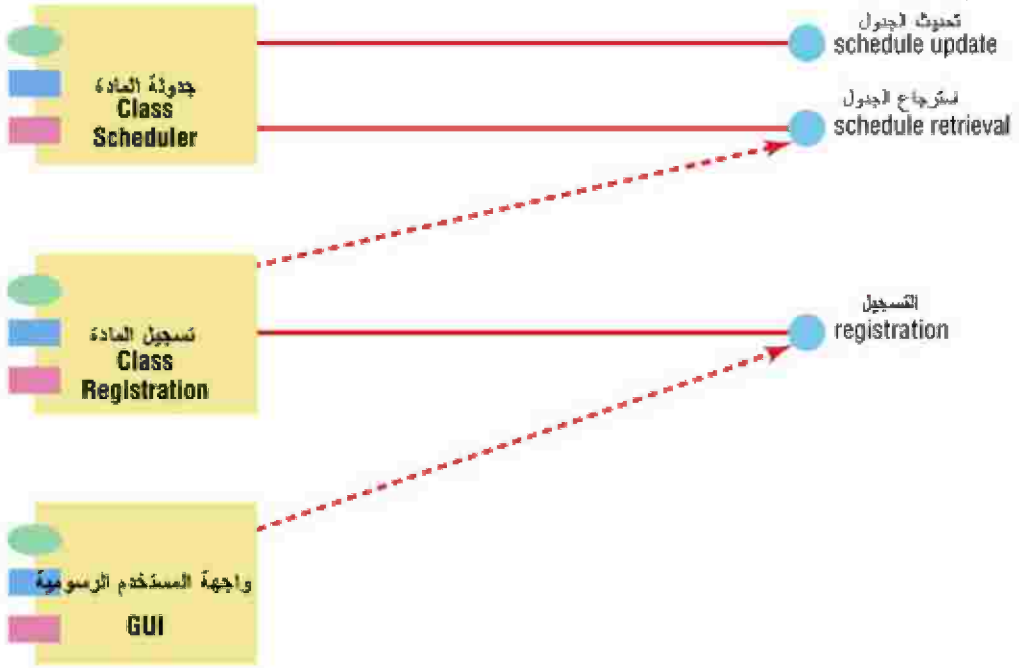
شكل أ–١٠ مثال على حزم وترابطات لغة التصميم العملية UML.

حزمة واجهة المستخدم تحتوي على جميع النوافذ، وتحتوي حزمة الكائنات العملية على نطاقات المشاكل التي قمت بتعريفها في مرحلة التحليل، تحتوي حزمة قاعدة البيانات على طبقة دائمة لتخزين البيانات واسترجاعها. في نظام التسجيل في الجامعة الذي ناقشناه سابقا، حزمة واجهة المستخدم قد تحتوي على مكتبات فصول Microsoft Windows لتطوير مختلف أنواع النوافذ. حزمة الكائنات العملية تحتوي على جميع فصول النطاقات، مثل: الطالب، المادة، إتاحة المادة، التسجيل، وغير ذلك . إذا كنت تستخدم خادم SQL ، الفصول في حزمة قاعدة البيانات ستحتوي على عمليات لتخزين البيانات، استرجاعها، تحديثها (كل ذلك باستخدام أوامر SQL). خلال التصميم، سوف تصقل نماذج التصميم الأخرى. كمثال، قد تحتاج إلى إظهار الترابطات بين كائن نافذة جديدة تقوم بطرحها خلال التصميم والكائنات الموجودة الأخرى في مخطط تتابعي. أيضا، عندما تختار لغة البرمجة للعمليات المعروضة في المخطط

التتابعي، يجب عليك أن توفر نفس الأسماء التي سوف تستخدم في البرنامج، مع أسماء جميع المعطيات. بالإضافة إلى الرسومات التي رأيت إلى الآن، يوثّق نوعات من الرسومات – رسومات المكونات و رسومات النشر – خلال مرحلة التصميم. مخطط المكونات Component Diagram يظهر مكونات البرامج و المجتزئات Modules وترابطاتها. كمثال، تستطيع أن تمخطط مخطط مكونات لإظهار إجتزاءات مصدرالكود Code، الكود الرقمي Binary Code، والكود القابل للتنفيذ Executable Code وترابطاتها. الشكل أ–١١ يظهر مخطط مكونات لنظام التسجيل في الجامعة. في هذا الشكل، تعرف ثلاث من مكونات البرامج: مجدول الفصول Class Scheduler، تسجيل الفصول، وواجهة المستخدم الصورية. الدوائر الصغيرة في المخطط تمثل الواجهات. واجهة التسجيل كمثال تستخدم لتسجيل طالب في فصل، وواجهة تحديث مجدول الفصول تستخدم في تحديث المجدول. نوع آخر من الرسوم، مخطط النشر Deployment Diagram (غير موضح)، يظهركيف تنتشر مكونات البرامج، العمليات، والكائنات في العمارة المادية للنظام. تقوم بإظهار أشكال المكونات المادية (الحواسيب، أجهزة الإتصال، إلخ..) وكيف توزع البرمجيات (المكونات، الكائنات، إلخ..) في الوحدات. كمثال، مخطط النشر لنظام التسجيل الجامعي يظهر شكل العقد في عمارة عميل/ خادم وانتشار كائنات التسجيل في خادم Windows NT ومكونات الواجهات الصورية في جهاز العميل.

عندما تنتهي مرحلة التصميم، تتحرك إلى مرحلة التنفيذ عند كتابة الكود للنظام. إذا كنت تستخدم لغة برمجة شيئية، ترجمة نماذج التصميم إلى الكود يجب أن تكون سهلة نسبيا. يتم اختبارالنظام بعد برمجته، يتم تطوير النظام بعد مروره بعدة تكرارات، مع كل تكرار يتم التزويد بنسخة أفضل من النظام. النماذج التي طورتها خلال التحليل، التصميم، التنفيذ صالحة للملاحة Navigable في كلا الاتجاهين.

شكل أ-١١ رسم مكونات تسجيل المادة.



مراجعة النقاط الأساسية
Key Points Review

١- عرف المصطلحات الأساسية الآتية: الإتحاد، مخطط الصنف، حدث، كائن، الصنف الكائني، العملية، مخطط التسلسل، حالة، تحول الحالة، لغة النمذجة الموحدة، حالة الاستخدام.

الاتحاد هو علاقة بين الأصناف الكائنية. ومخطط الصنف يحدد التركيب الثابت للنموذج الكائني والمكون من الأصناف الكائنية وتركيبها الداخلي بالإضافة إلى العلاقات التي تشارك فيها. الحدث هو الكائن الذي يحدث في فترة معينة من الزمن وهو حدث جدير بالملاحظة حيث أنه يفجر عملية انتقال الحالة. والكائن هو الكيان الموجود الذي له دور محدد جدا في مجال التطبيق ويتميز بالحالة والسلوك والهوية، ويعد مفهوم الصنف الكائني هو مجموعة من الكائنات التي لها تركيب وسلوك مشترك، أما العملية فهي الوظيفة أو الخدمة بين الكائنات أثناء فترة زمنية معينة، وتشمل الحالة مجموعة خصائص الكائن (الصفات والعلاقات) والقيم التي تحملها هذه الخصائص، تغير عملية انتقال الحالة صفات الكائن أو الروابط التي تربط الهدف بالأهداف الأخرى، تعتبر لغة تصميم النموذج الموحدة هي الفكرة التي تسمح لواضع النموذج بالفحص والتصوير والمقارنة بين النتائج الاصطناعية الخاصة بأنظمة البرامج كما في نماذج التجارة، حالة الاستخدام هي تسلسل كامل من الأفعال المرتبطة التي بدأت بواسطة الصانع وهي تمثل الطريقة المحددة لاستخدام النظام.

٢- صف المفاهيم والمبادئ التي تقع ضمن الطريقة الكائنية.

إن المفهوم الأساسي في الطريقة الكائنية هو أن نستطيع أن نصيغ العالم على أنه مجموعة من الأهداف (الكائنات) المرتبطة بحالتها وخصائصها وسلوكياتها المرتبطة بها، ومن خلال الاستخدام المختلف للكائن تتغير حالته. يمكن إخفاء تفاصيل التنفيذ الداخلية للكائن عن الرؤية الخارجية بواسطة تقنية الكبسلة (الإغلاق). ربما يكون نوع الكائن هي مجموعة أكبر أو مجموعة فرعية من الأنواع الأخرى للعناصر مشكلة في ذلك المبدأ العام المسيطر أو شبكة أنواع الكائن. وبهذه الطريقة ربما يرث الهدف السمات المميزة للأنواع الأخرى الأكبر التي ترتبط بها. ويمكن أن يكون الكائن أيضا جزءا من عنصر أخر إجمالي أكثر.

٣- قم بتطوير نموذج بسيط للمتطلبات باستخدام مخططات استخدام - حالة.

يتكون شكل حالة الاستخدام من مجموعة من الأجراءات المرتبطة التي بدأت بواسطة واضعي النموذج، تمثل حالة الاستخدام الوظيفة الكلية وليس القيام بفعل وحيد، ربما تمتد حالة الاستخدام إلى حالة أخرى عن طريق إضافة أجراءات وسلوكيات جديدة، ويمكن أن تستخدم حالة الاستخدام حالة استخدام أخرى عندما تستدعي إحداهما الأخرى.

٤- قم بتطوير نموذج كائن بسيط لمجال المشكلة مستخدما مخططات الصنف.

يصور مخطط الصنف التركيب الثابت لأنواع الكائن وتركيبها الخارجي والعلاقات التي تشترك فيها. ويتضمن تركيب الكائن كل من الاسم، الصفات والعمليات، لكل كائن طريقة واحدة لتعريفه بعيدا عن خصائصه، يمكن أن يكون نوع الكائن مجرد (أي بدون حالات مباشرة) أو واقعي (له حالات مباشرة). ربما يكون لأنواع الكائن اتحادات تشبه العلاقات في مفهوم علاقة الهوية بالتعددية والدرجة. ينتهي الاتحاد عندما يرتبط بالنوع الذي يمكن

تصنيفه مع دور الاتحاد.ويوضح مخطط الصنف المبدأ العام للعلاقات بين أنواع الكائن والأنواع الفرعية التي يمكن أن تكون كاملة أو غير كاملة ومفككة أو متداخلة. بالإضافة إلى ذلك، مخطط الصنف ربما يوضح تجميع الاتحاد بين أنواع الكائن.

٥- قم بتطوير نموذج بسيط للمتطلبات مستخدما مخططات الحالة والتسلسل.

تصور مخططات الحالة والتسلسل الأداء الديناميكي للنظام، يوضح مخطط الحالة جميع الحالات الممكنة للكائن والأحداث التي تفجر الكائن للانتقال من حالة إلى أخرى. تحدث عملية انتقال الحالة عن طريق التغيير في الخصائص المميزة للكائن أو التغيير في الروابط التي تربط الكائن بالكائنات الأخرى. يبدأ الكائن بالحالة الأولى وينتهي بالحالة الأخيرة وربما يكون للحالة شرط حماية حيث يتأكد من أن الخصائص المعينة للعنصر موجودة قبل حدوث عملية الانتقال، عندما تحدث عملية الانتقال يمكن لبعض الأفعال أن تحدث. يمثل مخطط التسلسل التفاعلات بين العناصر أثناء فترة محددة من الزمن. تمثل المحاور العمودية مخطط يمثل الزمن (تتجه إلى أسفل باتجاه المحور) والمحاور الأفقية المحور الذي يمثل هذه الكائنات مشاركة متعددة. كل كائن هو حبل السلامة والإنقاذ الذي يدل على وجود الكائن خلال فترة معينة. تتواصل هذه الكائنات مع بعضها البعض من خلال إرسال رسائل. من بين هذه الأنواع المختلفة من الرسائل هي متزامنة (لكل متصل عليه أن ينتظر من أجل كائن الاستقبال ليستكمل عملية الاتصال بالرقم قبل أن يستطيع المتصل استئناف التنفيذ) وبسيط (فمن أجله يتم نقل التحكم من الراسل إلى المرسل إليه).

المصطلحات الأساسية Key Terms Cheekpoints

وصل كل مصطلح من هذه المصطلحات الأساسية بتعريفه المناسب له:

١- مخطط يصور حالات الاستخدام و عمال النظام.

٢- شيء ما يحدث في نقطة معينة في وقت معين؛ أن ما يتم يستحق الملاحظة حيث يسبب نقل الحالة.

٣- فترة من الوقت يتم أثنائها أن يقوم الكائن بأداء العملية.

٤- مجموعة من الكائنات التي شاركت البناء المشترك والسلوك المشترك.

٥- الملاحظة التي تسمح لمصمم النموذج أن يخصص ويصور ويكون إنتاج صناعي من أنظمه البرمجيات وأيضا نماذج العمل.

٦- نوع من الرسائل الذي فيه يجب علي المتصل أن ينتظر الموضوع المستقبل لإنهاء تنفيذ عملية الاتصال قبل أن تستأنف عملية التنفيذ ذاتها.

٧- غاية الإتحاد التي يتم فيها الاتصال بالصنف.

٨- مخطط من الأمثلة المطابقة مع مخطط معطى للصنف و يسمى أيضا مخطط مثال.

٩- التغيرات في الصفات المميزة للكائن في روابط الكائن مع الكائنات الأخرى.

١٠- كيان ذا دور معرف جيدا في مجال التطبيق ولديه حاله، وسلوك، ذاتية.

١١- مخطط يبين مكونات البرمجيات أو المعايير والمقاييس و تبعياتها.

١٢- كينونة خارجية تتفاعل مع النظام (مشابهة للكينونة الخارجية في تخطيط تدفق البيانات).

١٣- مجموعة كاملة من الإجراءات المرتبطة ببعض التي أصدرها و الممثل، أنها تدل على طريقة معينة لاستخدام النظام.

١٤- «علاقة جزئيه» بين مكونات الكائن والكائن الإجمالي.
١٥- مؤشر عن عدد الكائنات المشاركة في العلاقة المعطاة.
١٦- الأسلوب المستخدم في إخفاء تفاصيل التنفيذ الداخلية للكائن من وجهات النظر الخارجية.
١٧- الوظيفة أو الخدمة المعطاة من قبل جميع الأمثلة بالصنف.
١٨- الحالة التي تشمل خصائص الكائن (الصفات المميزة والعلاقات).
١٩- الطريقة التي تدل علي كيفيه أن يعمل الكائن ويؤثر بشكل قوي.
٢٠- مخطط يبين البناء الثابت لنموذج الكائن: صنف الكائن، تركيبهم الداخلي والعلاقات المشاركين فيها.
٢١- العلاقة بين صنف الكائن.
٢٢- صنف بدون أمثلة مباشرة ولكن المنحدرين من هذا الأصل ربما يكون لهم أمثلة مباشرة.
٢٣- صنف به أمثلة مباشرة.
٢٤- تصور للتفاعلات بين الكائنات أثناء فترة معينة من الوقت.
٢٥- الرسالة التي تنقل السيطرة والتحكم من الراسل إلى المرسل إليه بدون وصف لتفاصيل الاتصال.

أسئلة مراجعة — Review Questions

١- وضح أوجه التشابه والإختلاف بين نماذج التحليل والتصميم الكائني ونماذج التحليل والتصميم المنظم.
٢- أذكر مثالاً يوضح حالة الإستخدام التجريدي. ويجب أن يتضمن المثال على الأقل حالتين لحالات الإستخدام وتوضيح علاقتهما بحالة الإستخدام التجريدي.
٣- وضح أهمية الدور التجميعي بإستخدام مخطط الصنف.
٤- أذكر مثال للمبدأ العام. يجب أن يحتوي المثال على الأقل على واحد من الأصناف العليا وثلاث من الأصناف الثانوية، وكحد أدنى على عمل واحد وصفة واحدة مميزة لكل صنف. وضح أيضاً العلامات المميزة والحالات الإضطرارية للأصناف الثانوية.
٥- أذكر مثال يوضح التجمّع. يجب أن يحتوي المثال على الأقل على مادة تجميعية واحدة وثلاث مواد تكوينية. وضح عدد العلاقات التجميعية.
٦- أذكر مثال يوضح الحالة الإنتقالية. يجب أن يوضح المثال كيفية تحوّل المادة إلى الحالة الإنتقالية تحت ظروف معينة.

مسائل وتمارين — Problems and Exercises

١- إن المخطط الخاص بحالات الإستخدام والموضح في الشكل أ-١ يوضح الدالّة الخاصة بترتيب أسماء التلاميذ ولكنه لا يوضح رسوم التعليم التي يدفعها التلاميذ. قُم بمراجعة المخطط لتوضيح ذلك. وأيضاً، وضح بعض السلوكيات الشائعة لحالات الإستعمال بإستخدام العلاقات «الضمنية» وتوضيحها على المخطط.
٢- بإفتراض أن الموظفين الممثلين في الجامعة لم يحصلوا على سندات مالية تثبت قيامهم بدفع المصروفات. وأن أقرانهم لم يحصلوا على وثيقة تنازل عن دفع المصروفات إلا أنهم يقومون بدفع ٢٥٪ من إجمالي المصروفات. وضح ذلك بإستخدام المخطط السابق.
٣- إخطط مخطط صنف يوضح التصنيفات والأعمال

والصفات المميزة والعلاقات للموقف التالي (إذا كان لديك أية إفتراضات أخرى، قم بتوضيحها وذكرها):
يوجد العديد من الكيميائيين الذين يقومون بعمل التجارب داخل المعمل. ويمكنهم إستخدام معدات وأدوات خاصة أثناء عمل التجارب. المعلومات الخاصة بالكيميائيين تتضمن الإسم ورقم الهاتف. أما المعلومات الخاصة بالتجربة تتضمن إسم التجربة وتاريخ البدء. امّا المعلومات الخاصة بالأدوات والمعدات تتضمن الرقم المسلسل والتكلفة. المؤسسة تطلب تسجيل التاريخ الخاص بتسليم الأدوات (اليوم الذي تمّ فيه تسليم الأدوات للكيميائيين). وإجمالي عدد الساعات (إجمالي عدد الساعات التي قام الكيميائي خلالها بإستخدام الأدوات الخاصة بالتجربة). والمؤسسة أيضاً تطلب تتبُّع وملاحظة إستخدام الكيميائي للأدوات. إنها تقوم بعمل ذلك من خلال تسجيل عدد الساعات التي قام الكيميائي خلالها بإستخدام الأدوات وذلك على جهاز الحاسب الآلي. يجب أن يقوم الكيميائي بعمل تجربة واحدة على الأقل وإستخدام عنصر واحد على الأقل من الأدوات. وليس من الضروري تحديد التجربة المفترض القيام بها أو حتى الأدوات المفترض إستخدامها.

٤–تتبنى المؤسسة تطوير نظام التسجيل والذي يقوم بحفظ المعلومات الخاصة بجميع (العربات) التي تمّ تسجيلها في ولاية محددة. وبالنسبة لكل (عربة) يتمّ تسجيلها في مكتب التسجيل، فإن النظام يقوم بحفظ إسم صاحب العربة والعنوان ورقم الهاتف، وتاريخ بداية التسجيل وتاريخ إنتهاؤه، والمعلومات الخاصة بلوحة الأرقام (المصدر، والسنة، والنوع، والرقم)، والملصق الخاص بالعربة (السنة، والنوع، والرقم)، ومصروفات التسجيل. بالإضافة إلى ذلك، فإنه يتم الإحتفاظ بالمعلومات الخاصة بالعربة ذاتها مثل: الرقم، تاريخ الصنع، النوع، الطراز، الحجم، عدد الركاب، ما إذا كانت تعمل بمحركات الديزل أم لا، اللون، التكلفة، المسافة بالميل. أمّا إذا كانت العربة مقطورة، فإنه ليس من الضروري معرفة المعلومات الخاصة بعدد الركاب أو ما إذا كانت تعمل بمحركات الديزل أم لا. أمّا بالنسبة للمقطورات التي تقطع مسافات كبيرة، فإنه من الضروري معرفة الأرقام والأطوال. يقوم النظام أيضاً بحفظ المعلومات الخاصة بسعة الحقائب التي يمكن أن تتحملها العربة، والحد الأقصى للحمولة، والقدرة الحصانية بالنسبة للدراجة البخارية. يقوم النظام بإرسال ملاحظات تنبيهية لأصحاب العربات الذين سوف تنتهي مدة التسجيل الخاصة بهم خلال شهرين. وعندما يقوم المالك بمراجعة طلب التسجيل، فإن النظام يقوم بتحديث بيانات التسجيل.

أ) قم بعمل نموذج كائني ثابت بإستخدام مخطط الصنف الذي يوضح جميع التصنيفات، والصفات، والوظائف، والعلاقات، وأعدادها. وضح القوائم الخاصة بكل وظيفة.

ب) قم بعمل مخطط صنف ثابت للعربات، بدءاً من تاريخ صنع العربة حتى تاريخ وصولها إلى المكان المخصص للخردوات. ويمكنك إضافة بعض الإفتراضات والتي لها علاقة بالموضوع.

ج) إختار أي حدث أو حالة من المستويات العليا لمخطط الحالة ووضح البنية الهيكلية الخاصة به (الحالات الثانوية وتحوّلاتها الإنتقالية).

د) إن أحد هذه التطبيقات يتمثل في «الملاحظات الخاصة بتجديد طلب التسجيل»، والذي يتم عمله مرة كل يوم. أخطط مخطط تسلسل شامل موضحاً التفاعلات المحتملة الخاصة باستخدام الحالة.

ملحق

ب

الطرق الرشيقة
Agile Methodologies

بعد دراسة هذا الملحق ستكون قادراً على:

- تعريف المنهجيات الرشيقة.
- شرح متى ستستخدم المنهجيات الرشيقة ومتى ستستخدم الطريقة التي تعتمد الهندسة في تطوير النظم.
- تعريف البرمجة القصوى.
- مناقشة طريقة المنهجيات الرشيقة لاتجاه حاجات النظم، تحديد التصميم وتركيب الكود والاختبار.

الاتجاه نحو الطرق الرشيقة
The Trend to Agile Methodologies

مر تحليل وتصميم النظم أو تطوير النظم بثلاث مراحل منذ ولادته. المرحلة الأولى: كانت غير منظمة والتي كان فيه المطورون ينظر إليهم باعتبارهم عباقرة وفنانين. وتميزت بنقص التوثيق وكذلك نقص أدوات التطوير إضافة إلى الاعتماد الكامل على المصمم لاستمرار تشغيل النظام الذي اخترعه. وعندما تعقد الحاسوب وكذلك الحاجيات في تطبيقات الأعمال. أصبح كثير من رجال الأعمال يرون نظمهم باعتبارها استثمارات وبذلك انتهت مرحلة اعتبار المصمم فنان وحل مكانها مرحلة اعتبار المصمم مهندساً. رؤية نظم المعلومات باعتبارها عمل هندسي جلب الكثير من الترتيب والتنظيم لها في النظرية والتطبيق والتطوير، فقد أدخل فيها أسس من الهندسة مثل العوامل والتأصيل والتحليل لتطبق في عمليات تطوير النظم. كذلك أصبح التوثيق والاختبارات الصارمة والتقنيات والأدوات المصممة أجزاء معيارية في التحليل والتصميم. وهذه النظرة التي تعتمد الهندسة جمعت بين طرق التحليل والتصميم التركيبي للتطوير والتي استخرجنا منها الكثير من الأدوات والتقنيات المستخدمة في هذا الكتاب. وعلى الرغم من أن الطريقة الهندسية مازالت تستخدم بكثرة هذه الأيام إلا أن كثير من النقاد اعتبرها بطيئة ومحبطة وعديمة الفائدة في زمن يدار فيه الاقتصاد عبر شبكات الإنترنت. باجتماع طريقة الاتجاه الكائني object-oriented واقتصاد الإنترنت بدأ الطور الحديث السائد في تطوير النظم. معتمداً على الاتجاه نحو الهدف والحاجة في السرعة، فإن هذه الطريقة الحديثة قد ضحت بمعالم ومراحل الطريقة الهندسية المنفردة مفضلة عليها التعاون اللصيق بين المصممين والعملاء وذلك بربط مجموعة من دورات الحياة في عدد محدود وكذلك زيادة سرعة نشر البرامج. وعلى الرغم من أن هناك عدة مناهج فردية تعكس هذه الطريقة الحديثة إلا أنها بمجموعها تسمى المنهجيات الرشيقة.

سنتعرف في هذا الملحق على المنهجيات الرشيقة بصفة عامة وسنركز على أشهرها وهو البرمجة القصوى eXtreme Programming باعتباره مثالاً على هذه الطريقة. أولاً سنصف المنهجيات الرشيقة وكيف تختلف عن الطرق التقليدية التي تعتمد على تطوير نظم الهندسة والتي تتضح في كثير من أجزاء هذا الكتاب. وكذلك سنصف متى ستكون المنهجيات الرشيقة مناسبة ومتى ستصلح الطرق التقليدية. ومن ثم سنناقش البرمجة القصوى eXtreme Programming وأساسياتها. وسيشرح الملحق كيف أن اتجاهات المطالب وتخصيص التصميم وربط الكود والاختبار تعالج في المنهجيات الرشيقة.

الطرق الرشيقة
Agile Methodologies

يأتي تحت مظلة المنهجيات الرشيقة مجموعة كبيرة من الأفراد المنهجيون مثل منهجيو أسرة الكريستال Crystal family of methodologies ومجموعة تطوير البرامج المتكيفة Adaptive Software Development وتطوير خاصية السواقات Feature Driven Development والبرمجة القصوى eXtreme Programming. اجتمع مجموعة من مناصري هذه الطرق البديلة لتحليل وتصميم النظم في مدينة يوته في الولايات المتحدة الأمريكية في فبراير ٢٠٠١ لكي يتصلوا لإجماع للأسس الضمنية التي تشملها أساليبهم المختلفة. وتحول هذا الإجماع إلى وثيقة سميت «البيان الرشيق»

(الشكل ب – ١) وقد اشترك هؤلاء المنهجيون في ثلاث أسس مفتاحية: ١ – التركيز على التكيف أكثر من المنهجيات التنبؤية، ٢ – التركيز على الناس أكثر من الأدوار، ٣ – عملية التكيف الذاتي.

يتناقش فريق المناهج الرشيقة في أن طريقة تطوير البرامج المقتبسة من الهندسة بصفة عامة لا تتناسب جيداً مع حقيقة البرامج المتنامية، ففي فروع الهندسة مثل الهندسة المدنية فإن المتطلبات تميل إلى أن تكون معلومة وعندما العمل الإبداعي فإن شكل البناء يكون متوقعاً إضافة إلى البناء ربما قرر ٩٠٪ من المجهود الكلي للمشروع أما بالنسبة للبرنامج في الجانب الآخر فإن المتطلبات نادراً ما تفهم جيداً فبذلك تقرر حوالي ١٥ ٪ من المجهود للمشروع تاركاً للتصميم ليمثل ٥٠٪ من مجهود المشروع. لذلك نجد التقنيات المطبقة والتي تعمل جيداً في المشاريع الثابتة مثل تشييد جسر تميل ألا تعمل جيداً في تصميم المشاريع الثقيلة المائعة مثل كتابة برنامج ، هكذا يقول مناصرو المنهجيات الرشيقة .فالمطلوب إذن هي منهجيات تقبل التغيير وقادرة في نفس الوقت للتعامل مع نقص التوقع ونجد أن واحدة من هذه التقنية التي تتعامل مع نقص التوقع هي التطوير المكرر. فالتطوير المكرر يركز على الإنتاج المكرر من النسخ المعدلة للنظام الذي يملك خاصية مرحلية من النظام الكلي. لذلك نجد أن التطوير المكرر يوفر تغذية راجعة للزبون والمصمم في ذات الوقت.

بيان تطوير البرامج الرشيقة

اتفاق الفوضويين السبعة عشر *:

كشفنا أفضل الأساليب لتطوير البرامج بعملها ومساعدة الآخرين على ذلك. وقد وصلنا بهذا العمل لتنظيم:

- الأفراد والتفاعل مع العمليات والأدوات.
- عمل البرامج بتوثيق شامل.
- تعاون العملاء عن طريق الاتصال التفاوضي.
- الاستجابة للتغيرات حسب الخطة الآتية.

فنحن نثمن العناصر السالفة وكذلك العناصر المتبقية وقد اتبعنا في ذلك الأسس التالية:

- إرضاء الزبون فهو أولويتنا الأولى وذلك من خلال التوفير المبكر والمستمر للبرامج القيمة (ذات القيمة العالية).
- نرحب بالمتطلبات المتغيرة حتى في المراحل المتأخرة من التطوير فعتاد المعالجات الرشيقة يتغير لمزايا الزبون التنافسية.
- توفير البرامج الفعالة وبسرعة في خلال أسبوعين إلى شهرين مع تحبيذ المدة الأقصر.
- يعمل أصحاب العمل والمصممون يومياً في المشروع.
- تبنى المشاريع من خلال أفراد محفزون نقر لهم البيئة المناسبة والدعم الذي يحتاجونه ونثق في أنهم سينجزون العمل.
- ووجدنا أن أكثر الأساليب تأثيراً وكفاءة لإيصال المعلومات إلى ومن خلال فريق التطوير هي المناقشة وجهاً لوجه.
- البرنامج الذي يعمل هو المقياس الأساسي لمدى التقدم.
- ترقي المعالجات الرشيقة التطور المحتمل، يجب أن ينال الكفلاء والمصممون والمستخدمون تقدماً مضطرداً في زمن محدد.
- الاهتمام المستمر والجودة الفنية والتصميم يعزز الرشاقة.
- البساطة – فن تقليل كمية العمل الذي لم يؤدي – ضرورية و أساسية.
- أفضل متطلبات التشييد والتصميم تنبثق من فرق العمل ذاتية التنظيم .
- يتفكر هذا الفريق بين كل فترة زمنية في كيفية زيادة الكفاءة و يضبط عمله طبقاً لذلك.

شكل ب–١

البيان الرشيق (من فولر وهايسميث، ٢٠٠١. استخدم بإذن مسبق) ٢٠٠١، من المؤلفين. وهذا الإعلان يسمح بنسخه في أي شكل، ولكن فقط في شكله الكامل.

ثانياً فإننا نجد أن التركيز على الناس في المنهجيات الرشيقة يكون على الأفراد أكثر من الأدوار التي يقوم بها الناس. فالأدوار التي يؤديها الناس بالنسبة لمحلل النظم، المختبر والمدير ليست مهمة مثل الفرد الذي يملأ هذه الأدوار. يقول «فاولر» إن التركيز على الأسس الهندسية بتطبيقها على تطوير النظم نتج عنه رؤية الناس وحدات قابلة للتبادل بدلاً من نظرة الناس أفراد مبدعون لكل واحد منهم شيء مميز يأتي به لفريق التطوير.

ثالثاً نجد أن منهجيوا المنهجيات الرشيقة قد طوروا التكيف الذاتي لتطوير عمليات البرامج. فعندما يتقدم البرنامج فإن الإجراءات يجب أن تراجع وتطور. فيقوم فريق العمل بأداء هذا الأمر بمراجعة الإجراءات التي ترتبط بإكمال التكرار. لذلك فإن ذلك يتضمن أن المصممون تكيفوا فلا تتوقع وجود منهج واحد متناغم في مؤسسة محددة. وستجد بدلاً من ذلك اختلافات في المناهج تعكس كل واحد منها تجربة خاصة لذلك الفريق الذي استخدمها.

لا تصلح المنهجيات الرشيقة لكل المشاريع لذلك فإنه يوصي بها في المشاريع التي تتضمن:

- حاجات ديناميكية أو غير متوقعة.
- مصممون مسئولون ومحفزون.
- زبائن يفهمون ويتفاعلون.

نجد في الجانب الآخر أن العملية الهندسية الموجهة ضرورية إذا كان فريق التطوير يتجاوز المائة شخص وأن المشروع يعمل بسعر أو عقد محدد. لذلك فإن المنظمات تحتاج طرق مختلفة لتطوير نظم المعلومات من خلال مزايا النظام وفريق التطوير.

على الرغم من أنه ليس متفق عليه عالمياً فإن الفروق الأساسية في طرق التطوير قد وضعت في جدول ب-١ بواسطة «بوهيم» و«تورنر» (٢٠٠٤). هذه المفاتيح الخمس وبما ساعدت في تحديد أي الطرق التطويرية ستعمل جيداً في مشروع محدد.

وكما ترى في الجدول ب-١ فإن حجم مشروع تطوير النظم دليل مؤشر في أن المنهجيات الرشيقة مناسبة أم لا فالمنهجيات الرشيقة تتناسب بصورة أحسن مع المشاريع الصغيرة أكثر من الكبيرة التي تتناسب معها المناهج التقليدية وكذلك تتناسب هذه المنهجيات مع النظم الأمنية الحساسة مثل نظم التحكم في محطات الطاقة النووية ومنصات إطلاق الصواريخ. وسبب استخدام الطرق التي تعتمد الهندسة في النظم الأمنية الحساسة لأنها قد أثبتت جدارتها في مثل هذه النظم.

والفرق الآخر الذي يميز بين المنهجيات الرشيقة والمنهجيات التقليدية هو الديناميكية فالمنهجيات الرشيقة تعمل بكفاءة في البيئة الملتهبة والمائعة والتي نجد فيها أن أوضاع الأعمال مضطربة وسريعة التغير. ونجد أن المنهجيات التقليدية تتناسب – ونتيجة لنسبة التخطيط العالية الموجودة فيها – للعمل في البيئة المستقرة. وقد سلف أن قرأت أن المنهجيات الرشيقة تركز على أهمية الأفراد أكثر من الأدوار لذلك فهي لا تعمل جيداً إلا في وجود مجموعة من خبراء الرشاقة الممتازون. أخيراً نجد أن المنهجيات الرشيقة تزدهر وتنمو في المنظمات والمؤسسات التي يزداد فيها الاضطراب والتغير السريع. بينما تعمل الطرق التقليدية في المؤسسات التي ترسم فيها الأدوار بصورة محددة وواضحة والتغير فيها قليل.

نجد أن أفضل ما كتب عن المنهجيات الرشيقة يسمى البرمجة القصوى eXtreme Programming والذي طور بواسطة كنت بيك في أواخر التسعينات من القرن الماضي والذي يوضح فيه مجموعة من الفلسفات الأساسية لهذه الطريقة الجديدة لتطوير النظم فنحن نستخدمها كمثال للفكرة المحورية الأساسية لكثير من المنهجيات الحديثة وسنشرحها في تفاصيل أكثر فيما يلي.

جدول ب–١. العوامل الخمسة التي تميز المنهجيات الرشيقة من المنهجيات التقليدية لتطوير النظم.

العامل	المناهج الرشيقة	المناهج التي تعتمد الهندسة
الحجم	يناسب مع فرق العمل الصغيرة والمنتجات القليلة. الاعتماد على المعرفة ذات التمدد المحدود.	النظم تتطور لتؤدي أعمالها مع فرق الأعمال الكبيرة ذات المنتجات الكثيرة.
الدقة	لم يجرب في المنتجات الحساسة. هناك صعوبات محتملة في التصاميم البسيطة ونقص في التوثيق.	صممت هذه المناهج لتؤدي المنتجات عالية الحساسية. ويصعب ربطها بالمنتجات الأقل حساسية.
الديناميكية	التصميم البسيط مناسب لها وكذلك البيئة الديناميكية ولكنها مصدر لاحتمال الكلفة العالية في البيئة المستقرة	مناسبة للبيئات المستقرة ذات التخطيط المفصل وواجهات التصميم الكبيرة.
المستخدمون	يحتاج الوجود المستمر لمجموعة من الخبراء النادرين وهناك خطورة في استخدام غير الرشيقين من الموظفين.	يحتاج لمجموعة من الخبراء النادرين في مراحل تعريف المشروع ولكن يمكن تشغيله في المراحل المتأخرة بعدد أقل إلا إذا كانت البيئة عالية الديناميكية
الثقافة	تزدهر وتنمو في المكان الذي يشعر فيه الناس بارتياح كبير ومحفزين بإعطائهم درجات من الحرية (الازدهار في الاضطراب).	تزدهر وتنمو بالبيئات التي يشعر فيها الموظفين بارتياح كبير ومحفزين بمنحهم أدوار محددة وكذلك إجراءات وتطبيقات واضحة (الازدهار في النظام).

المصدر: Boehm & Turner, 2004 مستخدم بتصرف

البرمجة القصوى
EXtreme Programming

البرمجة القصوى eXtreme Programming تتميز بدوراتها القصيرة في التطوير، وطريقة تخطيطها المتزايدة، وتركيزها على التجارب الذاتية التي يكتبها المبرمجون والعملاء لرصد عملية التطوير، واعتمادها على النهج المتنامي للتطوير الذي يظل باقيا طيلة عمر الجهاز. إن واحدة من المرتكزات الأساسية للبرمجة القصوى هي استخدامها فرقا تتكون من شخصين، وذكرت هنا، مع بقاء العميل جانبا أثناء عملية التطوير. إن الأجزاء ذات الصلة بالبرمجة القصوى والتي تتعلق بمواصفات التصاميم هي: ١– كيفية اندماج التخطيط والتحليل والتصميم والإنشاء مع بعضها البعض في مرحلة واحدة من النشاط، ٢– أسلوبها المتفرد في اجتذاب وتقديم احتياجات النظام ومواصفات

التصميم. إن جميع مراحل عمر البرنامج تلتقي ببعضها البعض في سلسلة من النشاطات مرتكزة على العمليات الأساسية للكود والاختبار والاستماع والتصميم.

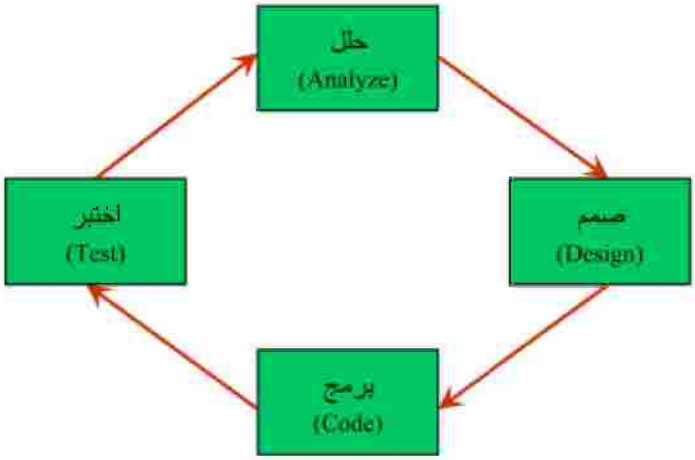
ففي هذا النهج، يكون الكود والاختبار إجراء منسجمة لنفس العملية. المبرمجين الذين يضعون الكود هم أيضا الذين يضعون الاختبارات. يكون التركيز في اختبار الأشياء التي تقبل الكسر أو الخطأ، ولكن ليس اختبار كل شيء. يختبر الترميز فورا بعد كتابته (وضعه). فالفلسفة الكاملة وراء البرمجه القصوى eXtreme Programming هي أن يندرج الكود في النظام الذي جرى تطويره له وتم اختباره بساعات قليلة بعد كتابته. يكتب الكود ويدمج في النظام ثم يختبر بعد ذلك. فإذا كانت جميع الاختبارات ناجحة فسيستمر التطوير. وإذا لم تنجح الاختبارات فسيعاد عمل الكود حتي يتحقق النجاح.

جزء آخر من البرمجة القصوى eXtreme Programming الذي يجعل عملية الكود واختباره تعملان بسلاسة، هو عملية البرمجة المزوجة. كب الكود والاختبار يؤديه شخصان يعملان مع بعض، يكتبان الكود والاختبار. كتابة البرمجة المزدوجة ليست عمل شخص واحد وأن لا يشارك الآخر. وإنما يعمل البرنامجان مع بعض في المشكلة التي ينويان حلها وفي تبادل المعلومات والرؤى والمهارات المشتركة. مقارنة بعمليات الترميز العام، فإن فوائد البرنامج المزدوج تشمل: ١- اتصال أكثر (وأفضل) بين المطورين، ٢- مستويات عليا من الإنتاج، ٣- ترميز ذو جودة-عالية، (٤) تدعيم الاعمال الاخرى في برنامج اكستريم مثل أسلوب الكود-و-الاختبار. وإن كان للبرمجة القصوى eXtreme Programming فوائده، مثله ومثل أي وسيلة أخرى لتطوير الأنظمة، إلا أنه ليس مناسبا لكل شخص أو يمكن تطبيقه لأي مشروع.

صميم عملية تطوير الأنظمة

The Heart of the Systems Development Process

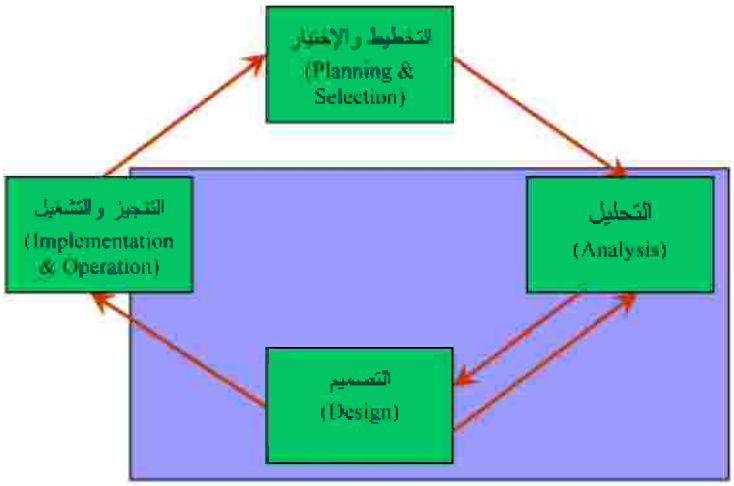
إن عمر تطوير الأنظمة المستخدم في هذا الكتاب يعطي طريقة ميسرة لتحقيق عملية تطوير الأنظمة وتساعد في ترتيب هذا الكتاب. لقد وضحت جليا المراحل المختلفة وحددت تماما علاقتها بكل منها وتسلسل المراحل من مرحلة إلى أخرى من البداية وحتى النهاية، في أسلوب محكم. في المنهجيات الرشيقة Agile Methodologies تجتمع العمليات الحالية مع النشاطات العامة اعتقدت أنها تدخل تحت التحليل والتصميم والتفسير، في عملية واحدة. بدلا من أن تكون متطلبات النظم التي تنتج في التحليل ومواصفات النظام التي عملت في التصميم والكود والاختبار عملت مع بداية التطبيق، فإن الأعمال الحالية تجمع جميع هذه النشاطات في عملية تحليل - تصميم - كود - اختبار واحدة (الشكل ب-٢). هذه النشاطات هي مركز تطوير الأنظمة، كما اقترحنا في الشكل (ب-٣) هذا الجمع بين النشاطات بدأ تطوير التطبيقات السريع (RAD) والذي قرأت عنه في الفصل الأول، والذي جاء في الكثير من المنهجيات الرشيقة Agile Methodologies.



شكل ب-٢

حلقة تحليل- تصميم- كود- اختبار

The Analysis-Design-Code-Test Loop



شكل ب-٣

صميم تطوير الأنظمة

The Heart of Systems Develpment

في الباقي من هذا الملحق، سوف نركز على مركز تطوير الأنظمة - تحليل وتصميم وتفسير من منظور المنهجيات الرشيقة Agile Methodologies إلا أن هنالك سهمان في الشكل ب-٣) بين التحليل والتصميم. وهذا يرمز إلى التكرار بين هذين النشاطين الذين يحتلان نواة الحركة التطويرية. في الأقسام التالي سوف تقرأ عن التكرار بين التحليل والتصميم ومكان تناسب تطبيق أوجه معينة فيه. القسم الأول عن تحديد المتطلبات. والقسم الثاني عن مواصفات التصميم، والقسم الأخير عن تطبيق الجمع بين الكود والاختبار. ولكما قرأت من قبل، هذه عبارة عن نكهات متعددة عن المنهجيات الرشيقة Agile Methodologies. وسوف نوضح كيف تعمل هذه الميثودولوجيا أولا مع عملية البرمجة القصوى، وتصميم رشيق المتمركز حول الاستخدام، والذي طور في البدء لاري كونستانتين وكيّف المنهجيات الرشيقة Agile Methodologies. من جيف باترون.

تحديد المتطلبات
Requirements Determination

قدمت ثلاث تقنيات لتحديد المتطلبات في هذا القسم. التقنية الأولى هي مشاركة المستخدم المستمرة في عملة التطوير، وهي تقنية تعمل جيدا خاصة بمصطلحات تطوير قليلة ورقيقة. النهج الثاني هو أسلوب يشبه تصميم تطبيقات مشترك (JAD) يسمى تصميم يرتكز-على-استخدام Agile. النهج الثالث هو لعبة التخطيط، والتي طورت كجزء من البرمجه القصوى.

مشاركة المستخدم المستمرة Continual User Involvemen: أحد الانتقادات للنهج العام المبني – على – الهندسة لتطوير الأنظمة هو أن المستخدمين يشاركون فقط في المراحل الأولى من التحليل. وبمجرد أن يتم تجميع المتطلبات منها، فلن يشترك المستخدمين في الأسلوب حتى يتم تركيب النظام ثم يطلب منهم الخروج منه. طبقا لذلك، ففي الوقت الذي يرى فيه المستخدم النظام مرة أخرى، فسيكون خلافا لما كانوا يتوقونه. وأيضا، بالنظر إلى كيفية تبدل مفهوم أعمالهم التجارية منذ انتهاء التحليل، فإن النظام من المحتمل جدا أن لا يتجاوب مع احتياجات المستخدم. وبشكل ما، إن هذه النظرة للمشاركة المستخدم المحدودة هو شكل مصغر للنهج العام للتطوير. ورغما عن ذلك فإن مشاركة المستخدم المحدودة صارت طبيعية جدا بأن ينظر إليها كمشكلة حقيقية وجادة في تطوير الأنظمة.

أحد الطرق لمشكلة مشاركة المستخدم المحدودة هو مشاركة المستخدمين باستمرار خلال عملية التحليل والتصميم كلها. مثل هذا النهج يعمل بأفضل ما يكون عندما يتبع التطور دورة تحليل – تصميم – كود – اختبار المفضلة من قبل كيّف المنهجيات الرشيقة Agile Methodologies للتطوير (الشكل ب-١) ، حيث يستطيع المستخدم أن يقدم معلومات عن الاحتياجات/ المتطلبات ومن ثم المشاهدة والتقييم بالشكل الذي تم به تصميم والكود واختبار تلك الاحتياجات. قد تستمر عملية التكرار هذه في عدة دورات حتى يتم تطوير أغلب التشغيلات الأساسية للنظام. المشاركة الفعالة للمستخدم في عملية التحليل والتصميم تعتبر جزءا أساسيا من نهج Agile ولكن أيضا جزء أساسي لتطبيق التطوير السريع (راجع الفصل الأول).

مشاركة المستخدم المستمرة هي شيء أساسي لنجاح تصميم بوينج واير ونظام تركيب الأسلاك wires بالنسبة للطائرة ٧٥٧ (بيدول، ٢٠٠٣). هذا النظام كان الغرض منه هو دعم المهندسين الذين كيفوا هيكل الطائرة للزبائن، ممهدين لهم تحليل أن كان جميع الأسلاك الـ٥٠٠ يمكن تركيبها في ٧٥٧. محاولة سابقة لبناء نظام مماثل استمر أكثر من ٣ سنوات، وأن النظام الناتج لم يستخدم أبدا. المحاولة الثانية التي ارتكزت على نظريات ايغايل، نتج عنها نظام دخل الإنتاج بعد ٦ أسابيع فقط. أحد مفاتيح النجاح هو منسق المستخدم الذي صرف نصف زمنه مع فريق تطوير صغير والنصف الآخر مع المستخدم في الجهة المقابلة. إضافة إلى الاتي ان دورة التحليل-التصميم والكود والاختبار، كان للفريق إصدارات أسبوعية للإنتاج. استخدم منسق المستخدم جميع الخطوات لهذه الطريقة. وبديهي، لكي ينجح مثل تحديد الاحتياجات هذا، فإن المستخدم الذي يعمل مع فريق التطوير يجب أن يكون متعلما جدا، ولكن عليه أو عليها أن تكون في موقع يمكنه من تنفيذ مسئولياته الطبيعية في الأعمال حتى يتم مشاركته بفاعلية في تطوير النظام.

تصميم رشيق متمركز حول الاستخدام Agile Usage-Centered Design: إن مشاركة المستخدم المستمرة في تطوير الأنظمة هي طريقة ممتازة لتثبت بأن الاحتياجات يمكن الحصول عليها بدقة تامة وتطبق على الفور في تصميم النظام. وعليه، فإن مثل هذا الاندماج المتواصل يعمل أفضل عندما يكون فريق التطوير صغيرا، كما هو الحال بالنسبة لمثال بوينج السابق. أيضا إنه ليس في كل الحالات ممكنا للإتاحة المتواصلة للمستخدمين أثناء التطوير المشروع. فلذلك فإن مطوري Agile قد أتوا بوسائل أخرى لإشراك فعال للمستخدمين في عملية تحديد الاحتياجات. أحد هذه الطرق يسمى تصميم ايغايل الذي يرتكز – على – المستخدم، وهو وسيلة من تسعة خطوات، والتي طبقناها وقدمناها على ثمانية خطوات في جدول (ب–٢).

لاحظ كيف ان العملية مجملا شبيهة باجتماع جاد (JAD). فقد اجتمع كل المتخصصين وعملوا بمساعدة المساعد. الشيء المميز عن تصميم (Agile). الذي يرتكز – على – المستخدم هي العملية التي تدعمها، بالتركيز على أدوار المستخدم والمهام المهمة لتحقيق تلك الأهداف. وبعد ذلك تجمع المهام وتحول الى نماذج من ورق – و – أقلام قبل انتهاء الاجتماع. المتطلبات التي تم الحصول عليها من المستخدمين والمطورين تم الحصول عليها كستر أنظمة نموذجية. أهم وجهين لهذا النهج هما دورة المنفذ، والتي تسمح لكل شخص أن يظهر شكواه علنا، وأن يستخدم بطاقات ٣ × ٥، التي تعمل كأدوات اتصال فعالة. وكما هو الحال لأي عملية تقنية تحليل أو تصميم، فإن تصميم ايغايل الذي يرتكز – على – المستخدم لا يصلح لكل مشروع أو كل شركة.

جدول ب–٢. الخطوات طريقة تصميم التركيز على الإستخدام الرشيق لتحديد المتطلبات.

١– تجميع مجموعه من الناس من ضمنهم محللون و مستخدمون و مبرمجون واختبار الموظفين في غرفه للتعاون على هذا التصميم. من ضمنهم ايضا" المسهل الذي يعرف معالجتها.

٢– اعطاء كل شخص فرصه حول النظام الحالي والتحدث عن المستقبل الذي يتعلق بالنظام الجديد.

٣– تحديد ما هي أدوار المستخدمين الأهم سوف تكون. تحديد من سوف يستخدمون النظام وما هو هدفهم من استخدام النظام. كتابة القوانين على ٣ × ٥ من الكروت. رتب الكروت حيث يكون المتماثلة قريبة لبعض. باتون (٢٠٠٢) يطلق عليها «قاعدة نموذج».

٤– تحديد ماذا سوف تكون أدوار مستخدم المهام وذلك من أجل تحقيق أهدافهم. اكتبها على ٣ × ٥ كروت. رتب الكروت بالأهم ثم التردد. ضع الكروت مع بعض بناءا تماثل المهام مع بعض باتون سماها نموذج المهام.

٥– كروت المهام سيصبح متثاقل مع بعض على الطاوله بناء على تماثلها. اقبض على كروت متماثله هذا يسمى سياق العمل.

٦– لكل كرت مهام بسياق التفاعل. اكتب وصف عن المهمة مباشرة وذلك على كرت المهام. قائمة من الخطوات الضرورية لتكملة المهمة. اجعل مواصفات التحدثية واضحة حتى يكون أسهل للقراءه والفهم.

٧– اعتبار كل جزء أو كتلة مجموعة مبدئية من المهام ليتم المساندة من جانب واحد لواجهة المستخدم مثل الشاشة وصفحة أو الحوار وانشاء صفحة ونموذج أولي لقلم رصاص. اظهار المقاس الأساسي وتوظيف عناصر الشاشة.

٨– عين دور المستخدم والخطوه خلال كل مهمة في سياق التفاعل كما هو معروض في الورق والنموذج الأولي لقلم الرصاص تأكّد من أن دور المستخدم يمكن أن يحقّق أهدافه باستخدام النّموذج الأوّليّ. نقّح النّموذج الأوّليّ وفقًا لذلك.

لعبة التخطيط The Planning Game: إن التقنيات المستخدمة في تحديد الاحتياجات البرمجة القصوى يتم الحصول عليها في «لعبة التخطيط». إن لعبة التخطيط هي في الحقيقة أسلوب ممنهج للتطوير والذي يسعى لتضخيم التفاعل المثمر fruitful interaction بين أولئك الذين يحتاجون إلى نظام جديد وأولئك الذين يقومون ببنائه. إن المشاركين في لعبة التخطيط، إذن، هما الأعمال والتطوير. إن الأعمال هو الزبون (العميل) ويمثله بمثالية شخص ملم تماما بالأسلوب مدعوما بالنظام المراد تطويره. أما التطوير يمثله أولئك الذين فعلا يقومون بتصميم وبناء النظام. إن جزئيات اللعبة هي التي يسميها بيك «بطاقات القصص». هذه البطاقات أنتجها الأعمال وهي تحتوي على وصف لطريقة أو شكل يمكن أن يضمن في النظام. كل بطاقة لها تاريخ وترقيم وبها فراغا لمتابعة وضعها طيلة زمن جهود التطوير.

إن لعبة التخطيط لها ثلاث مراحل: الاستكشاف exploration، الالتزام commitment التوجيه steering (الشكل ب-٤). في الاستكشاف فإن الشركة تنشأ بطاقة قصة لشيء يريد النظام الجديد عمله، أن الخلق يستجيب لتقدير كمية الوقت المطلوبة لتنفيذ الطريقة. وعند هذه النقطة فإن من العقلانية شق بطاقة القصة إلى عدة بطاقات قصة بينما مجال المظاهر والطريقة تصبح أكثر وضوحاً خلال النقاش.

شكل ب-٤
لعبة التخطيط للبرمجة القصوى (Extreme Programming).

وفي مرحلة الالتزام فإن العمل يفرز بطاقات قصة من ثلاثة مجموعات، واحد من المظاهر التي يكون ن الجميل امتلاكها. إن الانشاء والتطوير يفرز بطاقات القصة تبعاً للمجازفة قائماً على كيفية تقديرها بطريقة أحسن للوقت المحتاج إليه لخلق كل مظهر، العمل بعد ذلك يختار البطاقات التي ستضمن في التحرير التالي للمنتج. وفي

المرحلة النهائية وهي التوجيه فإن العمل له فرصة لرؤية كيف أن عملية الانشاء متقدمة والعمل مع التقدم لتعديل الخطة تبعاً لذلك. وغالباً مايحدث التوجيه مرة في كل ثلاثة أسابيع.

إن لعبة التخطيط بين العمل والانشاء تتابع بتكرار لعبة التخطيط، تلعب فقط بالمبرمجين. وبدلاً من بطاقات القصة، فإن المبرمجين يكتبون بطاقات المهمة والتي ترتكز على بطاقات القصة. ونموذجياً فإن عدة بطاقات عمل تولد لكل بطاقة قصة. إن تكرار لعبة التخطيط لها نفس الثلاث مراحل مثل لعبة التخطيط وهي الاستكشاف، الالتزام والتوجيه.

وخلال الاستكشاف فإن المبرمجين يحولون بطاقات القصة Story cards بطاقات عمل (مهام). وخلال الالتزام فهم يقبلون المسئولية للمهام وتوزن أحمال عملهم، وخلال التوجيه فإن المبرمجين يكتبون الرمز للمظهر، ويختبرونه وإذا عمل فإنهم يدخلون المظهر داخل المنتج الذي يتم إنشاؤه. إن إعادة لعبة التخطيط تحدث خلال فترة الوقت بين مقابلات مرحلة التوجيه في لعبة التخطيط.

يمكنك أن ترى كيف أن لعبة التخطيط هي مماثلة في بعض الأوجه للاستعمال الذكي للتصميم الممركز، وكلاهما يعتمد على المشاركة بالمستعملين ويعتمد على البطاقات كوسائل إتصال وتركز على المهام والذي من المفترض أن يؤديه النظام الذي تم تصميمه. ومع أن هذا الوصول يختلف عن بعض الطرق التقليدية لتحديد الاحتياجات مثل المقابلات والنموذج الأصلي، والكثير من المبادئ الأساسية هي متماثلة. أن المستهلكين أو المستعملين يبقون المصدر لماذا يفترض النظام أن يحمله المتطلبات لم تزل تجذب وتفاوض. وكل العملية لم تزل موثقة، مع أن المدى ورسمية التوثيق قد تختلف. وباعطاء طريق تعريف المتطلبات والتسجيل وتقسمها من قصص إلى مهام، فإن مواصفات التصميم يمكن تضمينها بسهولة لخواص متطلبات النوعية: وهي الاكتمال، التماسك، القابلية للعمل والمقدرة على اقتفاء الأثر.

مواصفات التصميم
Design Specifications

مع تركيزهم على برامج (الحاسب الآلي) على التوثيق، فإن الطرق الذكية تقدم حركة تكرار سريعة بين التصميم والكود. إن مواصفات التصميم تعتبر كجزء من عملية التصميم نفسها. وتتطور مع كل تكرار خلال دوره اختبار – الرمز – التصميم والتحليل. (الشكل ب-٤) تاركاً مساحة صغيرة في العملية لتوثيق التصميم مع الورقة أو الأشكال أو في أي طريقة أخرى بالمقارنة للرمز نفسه وفي نهاية قصوى فإن المشروع الذي يستعمل الطرق الذكية سوف له ينتج مواصفات تصميم من أي نوع آخر خلاف الرمز. وعند النهاية الثانية الأخرى حيث أكثر الطرق التقليدية أو الهندسية تستعمل فسوف يكون هناك مواصفات صميم مفصلة ومكتملة يتم توليدها. وإن هذه المواصفات سوف تسلم لفرق الاختبار والبرمجة عندما تكتمل. وهناك أيضاً الكثير من الاختيارات المتاحة بين هاتين النهائيتين. ومتى تم تنظيم مشروع إنشاء الأنظمة في محتوى الذكاء أو الطرق الأكثر تقليدية فذلك يعتمد على اعتبارات كثيرة مختلفة. وإذا اعتبر المشروع من خطورة كبيرة وأكثر تعقيداً وله فريق ان من مئات الناس، بالتالي فإن

أكثر الطرق التقليدية سوف تطبق. وأقل خطورة ومجازية وصغير من ناحية جهود الإنشاء فإن تلك تتطلب أكثر الطرق ذكاء وبعض العوامل المحددة تتضمن الممارسة الجهازية والمستويات والمدى للأجزاء المختلفة من النظام المتعاقد خارق للآخرين لينشأ. وبوضوح كلما كانت النسبة كبرية للنظام الذي سينتج، كلما كانت هناك الكثير من تفاصيل مواصفات التصميم يحتاج إليها بحيث يفهم المتعاقدين الفرعيين ما الذي يحتاج إليه.

إن الطرق الذكية تذهب من المتطلبات للتصميم الوظيفي مباشرة للرمز حيث النص الطويلة ومعدات المواصفات القائمة على الحاسب الآلي والأشكال الوسطية مثل خرائط الشكل تميل لأن تتجاهل في صالح إدراك مواصفات التصميم مباشرة من في الرمز نفسه.

عليك فقط أن تقرأ في البرمجة القصوى، وأن متطلبات المستعمل هذه ليست موضوعة في مواصفات تصميم مفرد مفصل مكتمل. وبالبدل فإن هذه المتطلبات الموجودة في الرمز هي تختبر كما في (الشكل ب–٢). **التصميم المبسط** هو مثل حفظ التصميم مبسطاً لكثير من المطورين والمبرمجين، لقد اتضح الشفره يمكن أن تستخدم للاتصال أو– لتعبير عن القصد التكتيكي، توصيف الخوازميا، بالإشارة إلى النقاط التوسيع والتقليص المستقبلي الممكن» (باك، ٢٠٠٠، ص ٤٤–٤٥). بعد الاختبار، في نهاية كل يوم عمل يمكن تجميع كل الشفرات التي تعمل وكذلك اصدار نسخ عمل النظام بشكل متكرر وغالباً ما يكون مرة في الأسبوع في بعض الحالات. مطوري البرمجة القصوى يصممون ويبنون الأنظمة في وقت قصير (نسبياً الى طرق المنظمة التقليدية) ويعملونها من دون كتابة أو مواصفات التصميم الرسومية. توظف البرمجة القصوى تقنيتين خاصة تستخدم لتطوير جودة التصميم بشكل مستمر كما أن المطورين يواصلوان الاعادة من خلال دورة تحليل – تصميم – برمجة – اختبار. هذه التقنيات هي التصميم البسيط وإعادة التقسيم.

التصميم البسيط
Simple design
خلق برامج عقل إلكتروني ومكوناته غير المعقدة والتي تعمل على المشكلة الجارية بدلاً عن خلق برامج عقل إلكتروني معقدة مصممة للمستقبل والذي قد لا يات. إن هذا المصطلح نعيش واحد من القيم الأساسية لاكستريم تبسيط البرمجة.

التصميم البسيط هو كما يبدو تماماً: المحافظة على بساطة التصميم. من السهل على كثير من المطورين والمبرمجين النظر للأمام وتوقع التغيرات في كيفية عمل النظام وعمل التصميم المناسب لهذه التغيرات . في كثير من الأحيان لا يمكن تحقيق الحالات المستقبلية المتوقعة و هذا ما يؤدي إلى ضياع الوقت والجهد في التصميم لمستقبل غير موثوق. وفقاً لباك، (٢٠٠٠) ينبغي أن يركز التصميم على حل المشاكل المباشرة وليس حال المشاكل المستقبلية التي قد تحصل أو لا. البرمجة القصوى.

وهناك أربعة معوقات للتصميم المبسط وهي:

- النظام يجب أن يتصل بأي شي تحتاج إليه.
- النظام يجب أن يشمل رمز مزدوج.
- النظام يجب أن يكون له رتب أقل ممكنة.
- النظام يجب أن يكون له طرق ممكنة اقل.

إعادة التقسيم
Refactoring
هو جعل النظام أكثر تبسيطاً بعد إضافة سمات جديدة.

تشير الطّبقات والطّرق إلى البرمجة الكائنيه والتصميم وفي عالم **إعادة التقسيم Refactoring** يمكن أن يمثلون قاعدة بيانات علائقيّة وفي صفوف القانون الّذي يعرض العمليّة في نظام، على التّوالي إعادة التقسيم Refactoring متعلّق بالهدف الكلّي «تصميم بسيط في البرامج القوية». إعادة التقسيم Refactoring هو مجرد فقط مبسط للنظام نموذجيًا بعد أن قد أُضيفَتْ سمة جديدة أو مجموعة سمات. بينما تُضَاف سمات أكثر لنظام، يصبح أعقد، وهذا التّعقيد سيُفَكَّر في القانون. بعد وقت تعقيد متزايد، المقاولون المبرمجون الشّديدون يقومون بإعادة تصميم النّظام. أما العملاء والمستخدمين فهم جاهزون لشراء هذه البرامج.

التنجيز
Implementation

رغم أن الكود والاختبار في حالات كثيرة هي أجزاء من عملية واحدة، فإنه ليس من الغريب في بيئات تطوير النظم الضخمة والمعقدة أن نجد العمليتين تمارس بمعزل عن بعضها البعض. وقد كيف أن هذه الحالة هي الشائعة (في الباب العاشر). فالشركات الكبيرة والمشاريع الكبيرة دائماً ما تخصص فرق عمل تقوم بتطوير خطط اختبار ومن ثم تستخدم نفس هذه الخطط لاختبار البرامج بعد كتابتها. وقد رأيت كيف أن هناك طرق متنوعة للاختبار، ومن ذلك يمكنك أن تستنتج كيف أن الاختبار يمكن أن يكون مفصلاً وشاملاً. وبطريقة البرمجة القصوى eXtreme Programming والمنهجيات الرشيقة الأخرى تكون عمليتا الاختبار والكود عمليتان متعلقتان ببعضهما لدرجة قوية. فالمبرمجون الذين يكتبون الكود هم أيضاً الذين يكتبون الاختبارات والفكرة العامة هنا أن الكود يختبر مباشرة بعد كتابته. وإذا اجتاز الكود الاختبار يتم استيعابه في النظام. وإن لم يجتاز الاختبار تعاد كتابته حتى يجتازه.

مراجعة النقاط الأساسية Key Points Review

١- تعريف المنهجيات الرشيقة Define Agile Methodologies.

هي عبارة عن مجموعة من المنهجيات المرتبطة مع بعضها البعض تختص بتطوير النظم. وتعتمد على غرضية التوجه والحاجة للسرعة. ويضحي هذا الاتجاه الأطوار الدقيقة والمتعددة للمنهج الهندسي الشائع حالياً في تطوير النظم مفضلة تعاون وثيق ما بين المطور والعميل، وجامعة الأطوار عديدة من دورات الحياة في أطوار قليلة، ومتخذه إصدارات عديدة سريعة للبرامج، ورغم أن هنالك طرق فردية عديدة تعكس هذا المنهج الجديد تعرف كلها باسم (المناهج الرشيقة) فإن هذه المناهج الفردية تشترك في كونها تركز على المنهجيات التكيفية، وعلى البشر وليس الأدوار. وعلى مجمل عملية التطور ذاتية التكيف.

٢- توضيح متى يجب استخدام المنهجيات الرشيقة ومتى يجب استخدام الطرق المعتمدة على الهندسة في تطوير النظم.

أولاً: تتناسب المنهجيات الرشيقة مع المشاريع الأصغر أكثر من المشاريع الأكبر. فالمنهجيات التقليدية تفضل للمشاريع الضخمة ولأنظمة الأمان الحرجة. ثانياً: تعمل المنهجيات الرشيقة بصورة أفضل في البيئات المتلاشية والسائلة، حيث أن حالة العمل تكون عادة غير مستقرة ومتغيرة بصورة متكررة، وذلك بسبب كبر كمية التخطيط الذي هو أصلاً جزء منها. والمنهجيات التقليدية تعمل أفضل عندما يكون النظام المطور يعمل في بيئة مستقرة. ثالثاً: بما أن المنهجيات الرشيقة تؤكد على أهمية الأفراد أكثر من الأدوار فإن المنهج الرشيق لا يعمل بصورة مثلى ما لم يكن هناك كمية مناسبة من الأفراد المدربين لاستخدامه بصورة جيدة. وهكذا نجد أن المنهج الرشيق يعمل بصورة أفضل في منظمة أو ثقافة تتصف بالتغير السريع وعدم الثبات. وتعمل المناهج التقليدية عندما يكون هناك قوانين تنظيمية واضحة غير متغيرة.

٣- تعريف البرمجة القصوى eXtreme Programming.

البرمجة القصوى eXtreme Programming طورت بواسطة "كنت بيك" في أواخر التسعينات. وتتميز هذه البرمجة بدورات تطويرها القصيرة. وطرق تخطيطها التزايدية وتركيزها على الاختبارات المؤتمتة التي تكتب بواسطة المبرمجين والزبائن لمراقبة عملية التطوير. وكذلك اعتمادها على طرق التطور للتطوير والتي تنفذ خلال مدة عمل النظام، وواحدة من المميزات الأساسية في البرمجة القصوى eXtreme Programming هي استخدامها لفريق من شخصين. ونقطة أساسية أخرى هي اتخاذ زبون في موقع أثناء عملية التطوير. والتقنية المستخدمة لتحديد المتطلبات في البرمجة القصوى eXtreme Programming تلتقط في (لعبة التخطيط) وهي طريقة مخصصة للتطوير تسعى لزيادة مستوى التفاعل المثمر بين الذين يحتاجون لنظام جديد و الذين يصممونه. ومن الخصائص المهمة الأخرى البرمجة القصوى eXtreme Programming تصميمها البسيط وقابليتها لإعادة ضبط عواملها.

٤- مناقشة طريقة المنهجيات الرشيقة لتحديد متطلبات النظام، خصائص التصميم والمزج ما بين الكود والاختبار.

هنالك ثلاثة طرق لتحديد المتطلبات من منظور المنهجيات الرشيقة قدمت في هذا الملحق:

أ) الاشتراك المستمر للمستخدم في عملية التطوير: وهي تقنية تعمل بصورة جيدة مع الفرق الصغيرة والمختارة.

ب) تصميم يرتكز على المنهجيات الرشيقة، وهي طريقة تشبه طريقة الـ (JAD).

ج) لعبة التخطيط التي طورت كجزء من البرمجة القصوى eXtreme Programming.

والمنهجيات الرشيقة ترقي طريقة تفاعلية سريعة ما بين التصميم والكود بحيث تلتقط خصائص عملية الكود كجزء من عملية التصميم نفسها. الخواص المشتركة في كل تفاعل خلال دورة اختبار كود تحليل التصميم تترك القليل في عمليتها لتوثيق التصميم بورقة أو شكل بياني أو أي طريقة أخرى غير الكود نفسه وبالنسبة للمنهجيات الرشيقة فإن الكود والاختبار هي أجزاء مترابطة بتلاحم داخل العملية نفسها. والمبرمجون الذين يكتبون الكود هم نفسهم من يكتبون الاختبار والفكرة العامة هي أن الكود يختبر مباشرة بعد كتابته.

المصطلحات الأساسية Key Terms Checkpoints

أدناه مصطلحات رئيسية وبين الأقواس ترى أرقام الصفحات التي ذكرت فيها هذه المصطلحات للمرة الأولى.

١ – **Refactoring (إعادة وضع العوامل).**

٢ – **Simple Design (التصميم البسيط).**

ضع كل مصطلح في مكانه المناسب من التعريفين الواردين أدناه.

..... ١ – جعل برنامج ما أبسط بعد إضافة خاصية معينة.

..... ٢ – صنع برنامج غير معقد ويكون برنامج يعمل لحل المشكلة الحالية بدلاً عن صنع برنامج معقد ومصمم لمستقبل قد لن يأتي.

أسئلة مراجعة Review Questions

١ – ما هي المنهجيات الرشيقة؟

٢ – صف المراحل الثلاثة التي يمر بها تطوير نظم المعلومات؟

٣ – ما هو قلب عملية تطوير النظم؟ وكيف تتناسب مع شبكات SDLC الأكبر حجماً؟

٤ – وضح عملية البرمجة القصوى eXtreme Programming؟

٥ – ما هو اشتراك المستخدم المستمر؟

٦ – ما هو التصميم المرتكز على المنهجيات الرشيقة؟ وكيف تستخدم لتحديد المتطلبات؟

٧ – ما هي لعبة التخطيط؟ وفيما تستخدم؟

٨ – كيف يتم التحكم في خصائص التصميم في المنهجيات الرشيقة؟

٩ – ما هو التصميم البسيط؟ و ما هي إعادة التعميل (وضع العوامل)؟

١٠ – وضح كيف تختلف عملية الاختبار في المنهجيات الرشيقة عنها في المناهج التقليدية؟

مسائل وتمارين — Problems and Exercises

١- متى يجب عليك استخدام المنهجيات الرشيقة، ومتى يجب أن تستخدم المناهج الهندسية في تطوير نظام ما؟ ادعم إجابتك.

٢- احصل على كتب ومقالات عن منهجيات رشيقة أخرى مثلاً Scrum and Feature Driven Development وقارن ما وجدته بما قرأته في هذا الملحق عن البرمجة المطلقة ثم أكتب تقريراً يوضح التشابه والاختلاف في هذه المنهجيات.

٣- ما مدى انتشار المنهجيات الرشيقة في صناعة تقنية المعلومات؟ ابحث عن إجابة هذا السؤال ثم أكتب تقريراً يوضح ما وجدت.

٤- مستخدماً الشبكة العنكبوتية العالمية، أوجد شركة أو مؤسسة قريبة منك تستخدم المنهجيات الرشيقة. اعقد مقابلة مع المطورين لتكتشف كيف يبدو منهجهم، وحضر تقريراً يعتمد على ما تحصلت عليه.

٥- افترض أنك أعطيت مهمة قيادة فريق يطور نظام إدخال أوامر مباشر (متصل)، كيف سيبدو مشروعك مستخدماً التقنيات البنيوية الموضحة في هذا الكتاب؟ وكيف يبدو مشروعك حين تستخدم منهجية رشيقة؟ اعقد مقارنة بين هذين المنهجين المختلفين لتطوير هذا النظام، تأكد من إدراج ذكر كافة الاعتبارات المهمة كمدة المشروع، ومسائل شخصية، وكمية ونوعية اشتراك العميل.

المراجع
References

Chapter 1: The Systems Development Environment

"2004 Software 500." Accessed October 11, 2004, www.softwaremag.com.

Alatas, A. Z. *Structured Analysis and Design of Information Systems.* Upper Saddle River, NJ: Prentice Hall, 1987.

Bohm, C., and I. Jacopini. "Flow Diagrams, Turing Machines, and Languages with only Two Formation Rules." *Communications of the ACM* 9 (May 1966): 366–71.

Bourne, K. C. "Putting Rigor Back in RAD." *Database Programming & Design* 7 (8) (August 1994): 25–30.

Chabrow, E. "Military IT spending surpasses all other federal agencies." *InformationWeek* (April 7, 2004). Accessed October 27, 2004, www.informationweek.com/story/showArticle.jhtml?articleID=18900493.

Cowley, S. "JP Morgan cancels $5bn IBM outsourcing deal." ComputerWeekly.com, (September 16, 2004). Accessed October 27, 2004, www.computerweekly.com/Article133359.htm.

Coy, P. "The Future of Work." *BusinessWeek* (March 22, 2004): 50–52.

DeMarco, T. *Structured Analysis and System Specification.* Upper Saddle River, NJ: Prentice Hall, 1979.

Fowler, M. "The New Methodology." April 2003. Accessed June 24, 2003, www.martinfowler.com/articles/newMethodology.html.

Gallivan, M. J., D. P. Truex, and L. Kvasny. "Changing perspectives in IT skill sets 1988–2003: A content analysis of classified advertising." *Data Base for Advances in Information Systems,* 35 (3) (2004): 64–87.

ITAA (Information Technology Association of America). *Workforce 2003 Report.* Presented at the National IT Workforce Convocation, Arlington, VA. Accessed July 9, 2003, www.itaa.org.

Mumford, E. "Participative Systems Design: A Structure and Method." *Systems, Objectives, Solutions* 1 (1) (1981): 5–19.

Naumann, J. D., and A. M. Jenkins. "Prototyping: The New Paradigm for Systems Development. *MIS Quarterly* 6 (3) (1982): 29–44.

Yourdon, E., and L. L. Constantine. *Structured Design.* Upper Saddle River, NJ: Prentice Hall, 1979.

Chapter 2: Managing the Information Systems Project

Abdel-Hamid, T. K., K. Sengupta, and C. Swett. "The Impact of Goals on Software Project Management: An Experimental Investigation." *MIS Quarterly* (23) (1999): 4.

Boehm, B. W. *Software Engineering Economics.* Upper Saddle River, NJ: Prentice Hall, 1981.

Butler, J. "Automating Process Trims Software Development Fat." *Software Magazine* 14 (5) (August 1994): 37–45.

Georgs, J. F., D. Batra, J. S. Valacich, and J. A. Hoffer. Object-Oriented Analysis and Design. Upper Saddle River, NJ: Prentice Hall, 2004.

Grupe, F. H., and D. F. Clevenger. "Using Function Point Analysis as a Software Development Tool." *Journal of Systems Management* 42 (December 1991): 23–28.

Gulman, P. J., J. G. Coopridar, and S. Faraj. "Enabling Software Development Team Performance during Requirements Definition: A Behavioral versus Technical Approach." *Information Systems Research* 9 (2) (1998): 101–25.

Hoffer, J. A., M. B. Prescott, and F. R. McFadden. *Modern Database Management.* 7th ed. Upper Saddle River, NJ: Prentice Hall, 2005.

Keil, M., B. C. Y. Tan, K. K. Wei, T. Saarinen, V. Tuunainen, and A. Wassenaar. "A Cross-Cultural Study on Escalation of Commitment Behavior in Software Projects." *MIS Quarterly* (24) (2000): 2.

Kettellust, M. C. "Avoiding Group-Induced Errors in Systems Development." *Journal of Systems Management* 42 (December 1991): 13–17.

King, J. "IT's Global Itinerary: Offshore Outsourcing Is Inevitable." Accessed September 15, 2003, www.cio.com. Information verified on September 17, 2003.

Kinch, L. J. "Software Project Management: An Integrated Perspective for an Emerging Paradigm." In *Framing the Domains of IT Management: Projecting the Future from the Past.* Edited by R. W. Zmud, Chapter 15, 285–304. Cincinatti: Pinnaflex Educational Resources.

Miller, R. W. "How to Plan and Control with PERT." *Managing Projects and Programs.* Harvard Business School Press. Boston: Harvard, 1988.

March, R. *Project Management: Best Practices for IT Professionals.* Upper Saddle River, NJ: Prentice Hall, 2001.

Nicholas, J. *Project Management for Business and Technology: Principles and Practice.* Upper Saddle River, NJ: Prentice Hall, 2001.

Page-Jones, M. *Practical Project Management.* New York: Dorset House, 1985.

Project Management Institute. *Work Breakdown Structures.* Newton Square, PA: Project Management Institute, 2001.

Rettig, M. "Software Teams." *Communications of the ACM* 33 (10) (1990): 23–27.

Thomsett, R. Foreword to *Practical Project Management,* by M. Page-Jones. New York: Dorset House, 1985.

Verma, V. K. *Human Resource Skills for the Project Manager.* Newtown Square, PA: Project Management Institute, 1996.

Verma, V. K. *Managing the Project Team.* Newton Square, PA: Project Management Institute, 1997.

Wideman, R. M. *Project and Program Risk Management.* Newton Square, PA: Project Management Institute, 1992.

Chapter 3: Systems Planning and Selection

Applegate, L. M., R. D. Austin, and F. W. McFarlan. *Corporate Information Strategy and Management: The Challenges of Managing in a Network Economy.* 6th ed. Boston, MA: Irwin/McGraw-Hill, 2003.

Atkinson, R. A. "The Motivations for Strategic Planning." *Journal of Information Systems Management* 7 (4) (1990): 53–56.

Atkinson, R. A., and J. Montgomery. "Reshaping IS Strategic Planning." *Journal of Information Systems Management* 7 (4) (1990): 9–16.

Carlson, C. K., E. P. Gardner, and S. R. Ruth. "Technology-Driven Long-Range Planning." *Journal of Information Systems Management* 6 (3) (1989): 24–29.

Cash, J. I., F. W. McFarlan, J. L. McKenney, and L. M. Applegate. *Corporate Information Systems Management.* 3d ed. Boston: Irwin, 1992.

DeGiglio, M. "Measure for Measure: The Value of IT." Accessed June 17, 2003, www.cio.com. Information verified September 29, 2003.

Dewan, S., S. C. Michael, and C-K. Min. "Firm Characteristics and Investments in Information Technology: Scale and Scope Effects." *Information Systems Research* 9 (3) (1998): 219–232.

Hasselbring, W. "Information System Integration." *Communications of the ACM* 43 (6) (2000): 33–38.

IBM. "Business Systems Planning." In *Advanced System Development/Feasibility Techniques*. Edited by J. D. Couger, M. A. Colter, and R. W. Knapp, 236–314. New York: Wiley, 1982.

Kerr, J. "The Power of Information Systems Planning." *Database Programming & Design* 3 (December 1990): 60–66.

King, J. "IT's Global Itinerary: Offshore Outsourcing Is Inevitable." Accessed September 15, 2003, www.cio.com. Information verified September 17, 2003.

King, J. L., and E. Schrems. "Cost Benefit Analysis in Information Systems Development and Operation." *ACM Computing Surveys* 10 (1) (1978): 19–34.

Kirsch, L. J. "Software Project Management: An Integrated Perspective for an Emerging Paradigm." In *Framing the Domains of IT Management: Projecting the Future from the Past*. Edited by R. W. Zmud, Chapter 15, 285–304. Cincinnati: Pinnaflex Educational Resources, 2000.

Koory, J. L., and D. B. Medley. *Management Information Systems: Planning and Decision Making*. Cincinnati, OH: South-Western, 1987.

Lederer, A. L., and J. Prasad. "Nine Management Guidelines for Better Cost Estimating." *Communications of the ACM* 35 (2) (1992): 51–59.

Luftman, J. N. *Managing the Information Technology Resource*. With C.V. Bullen, D. Liao, E. Nash, and C. Neumann. Upper Saddle River, NJ: Prentice Hall, 2004.

Pressman, R. S. *Software Engineering*. 5th ed. New York, NY: McGraw-Hill, 2001.

Martin, J. *Information Engineering*. Upper Saddle River, NJ: Prentice Hall, 1990.

McKeen, J. D., T. Guimaraes, and J. C. Wetherbe. "A Comparative Analysis of MIS Project Selection Mechanisms." *Data Base* 25 (February 1994): 43–59.

Moriarty, T. "Framing Your System." *Database Programming & Design* 4 (June 1991): 57–59.

Morton, C. "Information Competition: Can OLTP and DSS Peacefully Coexist?" *Data Base Management* 2 (June 1992): 24–28.

Parker, M. M., and R. J. Benson. *Information Economics*. Upper Saddle River, NJ: Prentice Hall, 1988.

Parker, M. M., and R. J. Benson. "Enterprisewide Information Management: State-of-the-Art Strategic Planning." *Journal of Information Systems Management* 6 (Summer 1989): 14–23.

Porter, M. *Competitive Strategy: Techniques for Analyzing Industries and Competitors*. New York: Free Press, 1980.

Porter, M. *Competitive Advantage*. New York: Free Press, 1985.

Pressman, R. S. *Software Engineering*. 3d ed. New York: McGraw-Hill, 1992.

Radosevich, L. "Can You Measure Web ROI?" *Datamation* (July 1996): 92–96.

Reingruber, M. J., and D. L. Spahr. "Putting Data Back in Database Design." *Data Base Management* 2 (March 1982): 19–21.

Ross, J., and D. Feeny. "The Evolving Role of the CIO." In *Framing the Domains of IT Management: Projecting the Future from the Past*. Edited by R. W. Zmud, Chapter 19, 385–402. Cincinnati, OH: Pinnaflex Educational Resources.

Segars, A. H., and V. Grover. "Profiles of Strategic Information Systems Planning." *Information Systems Planning* 10 (3) (1999): 199–232.

Shank, J. K., and V. Govindarajan. *Strategic Cost Management*. New York: Free Press, 1993.

Sowa, J. F., and J. A. Zachman. "Extending and Formalizing the Framework for Information Systems Architecture." *IBM Systems Journal* 31 (3) (1992): 590–616.

Yourdon, E. *Structured Walkthroughs*. 4th ed. Upper Saddle River, NJ: Prentice Hall, 1989.

Zachman, J. A. "A Framework for Information Systems Architecture." *IBM Systems Journal* 26 (March 1987): 276–92.

Chapter 4: Determining System Requirements

Carmel, E. "Supporting Joint Application Development with Electronic Meeting Systems: A Field Study." Unpublished doctoral dissertation, University of Arizona, 1991.

Carmel, E., J. F. George, and J. F. Nunamaker, Jr. "Supporting Joint Application Development (JAD) with Electronic Meeting Systems: A Field Study." *Proceedings of the thirteenth international conference on information systems*. Dallas, TX, December 1992: 223–32.

Carmel, E., R. Whitaker, and J. F. George. "Participatory Design and Joint Application Design: A Transatlantic Comparison." *Communications of the ACM* 36 (June 1993): 40–48.

Davenport, T. H. *Process Innovation: Reengineering Work through Information Technology*. Boston: Harvard Business School Press, 1993.

Dennis, A. R., J. F. George, L. Jessup, J. F. Nunamaker, Jr., and D. R. Vogel. "Information Technology to Support Electronic Meetings." *MIS Quarterly* 12 (December 1988): 591–624.

Dobyns, L., and C. Crawford-Mason. *Quality or Else*. Boston: Houghton Mifflin, 1991.

Hammer, M., and J. Champy. *Reengineering the Corporation*. New York: Harper Business, 1993.

Lucas, M. A. "The Way of JAD." *Database Programming & Design* 6 (July 1993): 42–49.

Mintzberg, H. *The Nature of Managerial Work*. New York: Harper & Row, 1973.

Moad, J. "After Reengineering: Taking Care of Business." *Datamation* 40 (20) (1994): 40–44.

Wood, J., and D. Silver. *Joint Application Design*. New York: Wiley, 1989.

Chapter 5: Structuring System Requirements: Process Modeling

Celko, J. "I. Data Flow Diagrams." *Computer Language* 4 (January 1987): 41–43.

DeMarco, T. *Structured Analysis and System Specification*. Upper Saddle River, NJ: Prentice Hall, 1979.

Gane, C., and T. Sarson. *Structured Systems Analysis*. Upper Saddle River, NJ: Prentice Hall, 1979.

Gibbs, W. W. "Software's Chronic Crisis." *Scientific American* 271 (September 1994): 86–95.

Hammer, M., and J. Champy. *Reengineering the Corporation*. New York: Harper Business, 1993.

Yourdon, E. *Managing the Structured Techniques*. 4th ed. Upper Saddle River, NJ: Prentice Hall, 1989.

Yourdon, E., and L. L. Constantine. *Structured Design*. Upper Saddle River, NJ: Prentice Hall, 1979.

Chapter 6: Structuring System Requirements: Conceptual Data Modeling

Aranow, E. B. "Developing Good Data Definitions." *Database Programming & Design* 2 (8) (1989): 36–39.

Booch, G. *Object-Oriented Analysis and Design with Applications*. 2d ed. Redwood City, CA: Benjamin Cummings, 1994.

Bruce, T. A. *Designing Quality Databases with IDEF1X Information Models*. New York: Dorset House Publications, 1992.

Chen, P. P-S. "The Entity-Relationship Model—Toward a Unified View of Data." *ACM Transactions on Database Systems* 1 (March 1976): 9–36.

Codd, E. F. "A Relational Model of Data for Large Relational Databases." *Communications of the ACM* 13 (6) (1970): 77–87.

Dutka, A. F., and H. H. Hanson. *Fundamentals of Data Normalization*. Reading, MA: Addison-Wesley, 1989.

Finkelstein, R. "Breaking the Rules Has a Price." *Database Programming & Design* 1 (June 1988): 11–14.

Fleming, C. C., and B. von Halle. "An Overview of Logical Data Modeling." *Data Resource Management* 1 (1) (1990): 5–15.

Fowler, M. *UML Distilled: A Brief Guide to the Object Modeling Language*. 2d ed. Reading, MA: Addison-Wesley, 2000.

Gibson, M., C. Hughes, and W. Remington. "Tracking the Trade-Offs with Inverted Lists." *Database Programming & Design* 2 (January 1989): 28–34.

Gottesdiener, E. "Turning Rules into Requirements." *Application Development Trends* 6 (7) (1999): 37–50.

Hay, D. *Data Model Patterns: Conventions of Thought*. New York: Dorset House Publishing, 1996.

Hoffer, J. A., M. B. Prescott, and F. R. McFadden. *Modern Database Management*. 7th ed. Upper Saddle River, NJ: Prentice Hall, 2005.

Inmon, W. H. "Using the Generic Data Model." Accessed January 12, 2004, www.billinmon.com. Specific URL: www.dmreview.com/master.cfm?NavID=55&EdID=4820.

Kimball, R., and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Data Modeling*. 2d ed. New York: John Wiley & Sons, Inc., 2002.

Moody, D. "The Seven Habits of Highly Effective Data Modelers." *Database Programming & Design* 9 (October 1996): 57, 58, 60–62, 64.

Rodgers, U. "Denormalization: Why, What, and How?" *Database Programming & Design* 2 (12) (1989): 46–53.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-Oriented Modeling and Design*. Upper Saddle River, NJ: Prentice Hall, 1991.

Sandifer, A., and B. von Halle. "A Rule by Any Other Name." *Database Programming & Design* 4 (2) (1991a): 11–13.

Sandifer, A., and B. von Halle. "Linking Rules to Models." *Database Programming & Design* 4 (3) (1991b): 13–16.

Silverston, L. *The Data Model Resource Book. Vol. 1. A Library of Universal Data Models for All Enterprises*. New York: John Wiley & Sons, Inc., 2001a.

Silverston, L. *The Data Model Resource Book. Vol 2. A Library of Data Models for Specific Industries*. New York: John Wiley & Sons, Inc., 2001b.

Silverston, L. "A Universal Data Model for Relationship Development." *DM Review* 12 (3) (2002): Accessed January 12, 2004, www.dmreview.com/master.cfm?NavID=55&EdID=4820.

Storey, V. C. "Relational Database Design Based on the Entity-Relationship Model." *Data and Knowledge Engineering* 7 (1) (1991): 47–83.

Teorey, T. J., D. Yang, and J. P. Fry. "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model." *Computing Surveys* 18 (2) (1986): 197–221.

UML Notation Guide. Document accessed at: www.rational.com/uml. Copyright held by Rational Software Corporation, Microsoft Corporation, Hewlett-Packard Company, Oracle Corporation, Sterling Software, MCI Systemhouse Corporation, Unisys Corporation, ICON Computing, IntelliCorp, i-Logix, IBM Corporation, ObjecTime Limited, Platinum Technology Incoporated, Ptech Incoporated, Taskon A/S, Reich Technologies, Softeam.

Chapter 7: Selecting the Best Alternative Design Strategy

Applegate, L. M., and R. Montealegre. "Eastman Kodak Company: Managing Information Systems through Strategic Alliances." Harvard Business School case 9-192-030. Cambridge, MA: President and Fellows of Harvard College, 1991.

Harper, D. "Seek a Partner, Not a Vendor." *Industrial Distribution* 83 (April 1994): 97.

Hartmann, C. R. "How to Write a Proposal." *D & B Reports* 42 (March–April 1993): 62.

King, R. T., Jr., and J. E. Rigdon. "Hewlett Prints Computer Services in Big Capital Letters." *Wall Street Journal* (June 3, 1994): B8.

"Microcomputer Procurement Guidelines." *Public Works* 125 (April 15, 1994): G23 1.

Mikulski, F. A. *"Managing Your Vendors: The Business of Buying Technology."* Upper Saddle River, NJ: Prentice Hall, 1993.

Moad, J. "Inside an Outsourcing Deal." *Datamation* 39 (February 15, 1993): 20–27.

"More Companies Are Chucking Their Computers." *BusinessWeek* (June 19, 1989): 72–74.

Semich, J. W. "Is it Bye-Bye Borland?" *Datamation* 40 (June 1994, 12): 52–54.

Stein, M. "Don't Bomb Out When Preparing RFPs." *Computerworld* 27 (February 15, 1993): 102.

Zachary, G. P. "How Ashton-Tate Lost Its Leadership in PC Software Arena." *Wall Street Journal* (April 1, 1990): A1–A2.

Zachary, G. P., and W. M. Bulkeley. "Ashton-Tate Loses Flagship Software's Copyright Shield." *Wall Street Journal* (December 14, 1990): B1, B4.

Chapter 8: Designing the Human Interface

Apple Computer. *Macintosh Human Interface Guidelines*. Reading, MA: Addison-Wesley, 1993.

Benbasat, I., A. S. Dexter, and P. Todd. "The Influence of Color and Graphical Information Presentation in a Managerial Decision Simulation." *Human-Computer Interaction* 2 (1986): 65–92.

Blattner, M., and E. Schultz. "User Interface Tutorial." Presented at the 1988 Hawaii International Conference on System Sciences, Kona, Hawaii, January 1988.

Carroll, J. M. *Designing Interaction*. Cambridge: Cambridge University Press, 1991.

Castro, E. *XML for the World Wide Web*. Berkeley, CA: Peachpit Press, 2001.

Cooper, A., and R. M. Reimann. *About Face 2.0: The Essentials of Interaction Design*. New York: John Wiley & Sons, 2003.

Dumas, J. S. *Designing User Interfaces for Software*. Upper Saddle River, NJ: Prentice Hall, 1988.

Flanders, V., and M. Willis. *Web Pages That Suck: Learn Good Design by Looking at Bad Design*. Alameda, CA: Sybex Publishing, 1998.

Hoffer, J. A., M. B. Prescott, and F. R. McFadden. *Modern Database Management*. 7th ed. Upper Saddle River, NJ: Prentice Hall, 2005.

Huff, D. *How to Lie with Statistics*. New York: W. W. Norton, 1954.

Jarvenpaa, S. L., and G. W. Dickson. "Graphics and Managerial Decision Making: Research Based Guidelines." *Communications of the ACM* 31 (6) (1988): 764–74.

Johnson, J. *GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers*. San Diego: Academic Press, 2000.

Lazar, J. *User-Centered Web Development: Theory into Practice*. Sudbury, MA: Jones & Bartlett, 2004.

Lindholm, C., and T. Keinonen. *Mobile Usability: How Nokia Changed the Face of the Mobile Phone*. Chicago: McGraw-Hill Professional, 2003.

McCracken, D. D., R. J. Wolfe, and J. M. Spoll. *User-Centered Web Site Development: A Human-Computer Interaction Approach*. Upper Saddle River, NJ: Prentice Hall, 2003.

McKay, E. N. Developing User Interfaces for Microsoft Windows. Redmond, WA: Microsoft Press, 1999.

Nielsen, J. "Marginalia of Web Design." November 1996; www.useit.com/alertbox/9611.html.

Nielsen, J. "Loyalty on the Web." August 1, 1997; www.useit.com/alertbox/9708a.html.

Nielsen, J. "Using Link Titles to Help Users Predict Where They Are Going." January 11, 1998a; www.useit.com/alertbox/980111.html.

Nielsen, J. Personalization Is Over-Rated. October 4, 1998b; www.useit.com/alertbox/981004.html.

Nielsen, J. "Web Pages Must Live Forever." November 29, 1998c; www.useit.com/alertbox/981129.html.

Nielsen, J. "User Interface Directions for the Web." *Communications of the ACM*, 42 (1) (1999a), 65–71.

Nielsen, J. "Trust or Bust: Communicating Trustworthiness in Web Design." March 7, 1999b; www.useit.com/alertbox/990307.html.

Nielsen, J. *Designing Web Usability: The Practice of Simplicity*. Indianapolis, IN: New Riders Publishing, 2000.

Norman, K. L. *The Psychology of Menu Selection*. Norwood, NJ: Ablex, 1991.

Porter, A. *C11 Programming for Windows*. Berkeley, CA: Osborne McGraw-Hill, 1990.

Seffah, A., and H. Javahery. *Multiple User Interfaces: Cross Platform Applications and Context-Aware Interfaces*. New York: John Wiley & Sons, 2003.

Shneiderman, B., and C. Plaisant. *Designing the User Interface*. 4th ed. Reading, MA: Addison-Wesley, 2004.

Snyder, C. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. San Francisco: Morgan Kaufmann Publishers, 2003.

Sun Microsystems. *Java Look and Feel Design Guidelines*. Reading, MA: Addison-Wesley, 1999.

Wagner, R. "A GUI Design Manifesto." *Paradox Informant* 5 (June 1994): 36–42.

Chapter 9: Designing Databases

Babad, Y. M., and J. A. Hoffer. "Even No Data Has a Value." *Communications of the ACM* 27 (August 1984): 748–56.

Codd, E. F. "A Relational Model of Data for Large Relational Databases." *Communications of the ACM* 13 (June 1970): 77–87.

Gibson, M., C. Hughes, and W. Remington. "Tracking the Trade-Offs with Inverted Lists." *Database Programming & Design* 2 (January 1989): 28–34.

Hoffer, J. A., M. B. Prescott, and F. R. McFadden. *Modern Database Management*. 6th ed. Upper Saddle River, NJ: Prentice Hall, 2002.

Navathe, S., R. Elmasri, and J. Larson. "Integrating User Views in Database Design." *Computer* (January 1986): 50–62.

Rodgers, U. "Denormalization: Why, What, and How?" *Database Programming & Design* 2 (12) (December 1989): 46–53.

Viehman, P. "24 Ways to Improve Database Performance." *Database Programming & Design* 7 (2) (February 1994): 32–41.

Chapter 10: Systems Implementation and Operation

Bell, P., and C. Evans. *Mastering Documentation*. New York: Wiley, 1989.

Bloor, R. "The Disappearing Programmer." *DBMS* 7 (9) (August 1994): 14–16.

Brooks, F. P., Jr. *The Mythical Man-Month*. Anniversary edition. Reading, MA: Addison-Wesley, 1995.

Cole, K., O. Fischer, and P. Saltzman. "Just-in-Time Knowledge Delivery." *Communications of the ACM* 40 (7) (1997): 49–53.

Crowley, A. "The Help Desk Gains Respect." *PC Week* 10 (November 15, 1993): 139.

Dillon, N. "Internet-Based Training Passes Audit." *Computerworld* (November 3, 1997): 47–48.

Eason, K. *Information Technology and Organisational Change*. London: Taylor & Francis, 1988.

Galvin, T. "Industry Report." *Training* (October 2002): 24–33.

Ginzberg, M. J. "Early Diagnosis of MIS Implementation Failure: Promising Results and Unanswered Questions." *Management Science* 27 (4) (1981): 459–78.

Ginzberg, M. J. "Key Recurrent Issues in the MIS Implementation Process." *MIS Quarterly* 5 (2) (June 1981): 47–59.

Haruna, M. "Using Documentation as a Life-Cycle Tool." *Software Magazine* (December 1992): 41–46.

Henderson, J. C., and M. E. Treacy. "Managing End-User Computing for Competitive Advantage." *Sloan Management Review* (Winter 1986): 3–14.

Ives, B., and M. H. Olson. "User Involvement and MIS Success: A Review of Research." *Management Science* 30 (5) (1984): 586–603.

Jones, C. "How to Measure Software Costs." *Application Development Trends* (May 1997): 32–38.

Kaplan, S. "Now Is the Time to Pull the Plug on Your Legacy Apps." *CIO Magazine*. (March 15, 2002). Accessed November 1, 2004, www.cio.com.

Kling, R., and S. Iacono. "Desktop Computerization and the Organization of Work." *Computers in the Human Context*. Edited by T. Forester, 336–56. Cambridge, MA: MIT Press, 1989.

Lee, D. M. S. "Usage Pattern and Sources of Assistance for Personal Computer Users." *MIS Quarterly*, 10 (December 1986): 313–25.

Markus, M. L. "Implementation Politics: Top Management Support and User Involvement." *Systems/Objectives/Solutions* 1 (4) (1981): 203–15.

Martin, E. W., C. V. Brown, D. W. DeHayes, J. A. Hoffer, and W. C. Perkins. *Managing Information Technology: What Managers Need to Know*. 4th ed. Upper Saddle River, NJ: Prentice Hall, 2002.

Mosley, D. J. *The Handbook of MIS Application Software Testing*. Englewood Cliffs, NJ: Yourdon Press, 1993.

Nelson, R. R., and P. H. Cheney. "Training End Users: An Exploratory Study. *MIS Quarterly* 11 (December 1987): 547–59.

Pressman, R. S. *Software Engineering: A Practitioner's Approach*. 5th ed. New York, NY: McGraw-Hill, 2001.

Schrage, M. "Unsupported Technology: A Prescription for Failure." *Computerworld* 27 (May 10, 1993): 31.

Schurr, A. "Support Is No. 1." *PC Week* 10 (November 15, 1993): 126.

Tait, P., and I. Vessey. "The Effect of User Involvement on System Success: A Contingency Approach." *MIS Quarterly* 12 (1) (March 1988): 91–108.

Torkzadeh, G., and W. J. Doll. "The Place and Value of Documentation in End-User Computing." *Information & Management* 24 (3) (1993): 147–58.

Yourdon, E. *Managing the Structured Techniques*. 4th ed. Upper Saddle River, NJ: Prentice Hall, 1989.

Appendix A: Object-Oriented Analysis and Design

Booch, G. *Object-Oriented Analysis and Design with Applications*. 2d ed. Redwood City, CA: Benjamin Cummings, 1994.

Coad, P., and E. Yourdon. *Object-Oriented Analysis*. 2d ed. Upper Saddle River, NJ: Prentice Hall, 1991.

Coad, P., and E. Yourdon. *Object-Oriented Design*. Upper Saddle River, NJ: Prentice Hall, 1991.

Eriksson, H., and M. Penker. *UMLToolkit*. New York: Wiley, 1998.

Fowler, M. *UML Distilled: A Brief Guide to the Object Modeling Language*. Reading, MA: Addison-Wesley, 2000.

Jacobson, I., M. Christerson, P. Jonsson, and G. Overgaard. *Object-Oriented Software Engineering: A Use-Case Driven Approach*. Reading, MA: Addison-Wesley, 1992.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-Oriented Modeling and Design*. Upper Saddle River, NJ: Prentice Hall, 1991.

UML Document Set. Version 1.0 Santa Clara: Rational Software Corp, January 1997.

UML Notation Guide. Version 1.0 Santa Clara: Rational Software Corp, January 1997.

Appendix B: Agile Methodologies

Beck, K. *eXtreme Programming eXplained.* Upper Saddle River, NJ: Addison-Wesley, 2000.

Bedoll, R. "A Tale of Two Projects: How 'Agile' Methods Succeeded After 'Traditional' Methods Had Failed in a Critical System-Development Project." *Proceedings of 2003 XP/Agile Universe Conference* (2003). Accessed September 19, 2003, www.agileuniverse.com/home.

Boehm, B., and R. Turner. *Balancing Agility and Discipline.* Boston: Addison-Wesley, 2004.

Constantine, L. "Process Agility and Software Usability: Toward Lightweight Usage-Centered Design." *Information Age* (August/September 2002).

Fowler, M. "The New Methodology." April 2003. Accessed June 24, 2003, www.martinfowler.com/articles/newMethodology.html.

Martin, R. C. "Iterative and Incremental Development (IID)." Accessed July 8, 2003, www.objectmentor.com/resources/articleIndex.

Patton, J. "Designing Requirements: Incorporating Usage-Centered Design into an Agile SW Development Process." In *XP/Agile Universe 2002.* Edited by D. Wells and L. Williams, 1–12. LNCS 2418. Berlin: Springer-Verlag, 2002.

قائمة المصطلحات
Glossary of Terms

المصطلح الإنجليزي	المصطلح بالعربي	الشرح
Agile methodologies	المنهجية الرشيقة	الطرق الحالية لتطوير الأنظمة التي تركز على: المنهجيات الملائمة، والأشخاص بدلا من الأدوار، وعملية التطوير ذات التلائم الشامل.
Baseline project plan (BPP)	خطة العمل الاساسية	تمثل نتيجة رئيسية من نتائج مرحلة بدء وتخطيط المشروع. وتحتوي على تقدير لمجال المشروع وفوائده وتكاليفه ومخاطره وموارده اللازمة.
Boundary	الحد	هو الفاصل الذي يفصل داخل النظام وخارجه والذي يفصل نظام عن الأخر في المنظمات.
Break-even analysis	التحليل المتعادل	هو نوع من تحليل الفوائد المالية للتعرف علي النقطة التي تتساوي فيها قيمتي الفوائد والتكاليف(ان امكن)
Business case	حالة العمل	تقرير مكتوب يوضح سبب وضع احد نظم المعلومات. ويوضح هذا التقرير الفوائد الاقتصادية والتكاليف والجدوي الفنية والتنظيمية للنظام المقترح.
Business process reengineering (BPR)	إعادة هندسة إجراءات (عمليات) الأعمال (الهندرة)	البحث عن تغيير جذري في عمليات الأعمال لتحسين المنتجات والخدمات، وتنفيذ ذلك التغيير.
Closed-ended questions	الأسئلة المغلقة	هي أسئلة تطرح في المقابلات (والاستبانات) يطالب المجيب عنها بأن يختار جوابه من بين إجابات محددة.
Cohesion	التلاحمية	يقيس مدى قيام النظام أو النظام الفرعي بمهمة منفردة

Component	المكونة	جزء لا اختزالي أو مجموعة من الأجزاء التي تصنع النظام، وتسمى نظام فرعي subsystem
Computer-aided software engineering (CASE)	هندسة البرمجيات بمساعدة الحاسوب	هي أدوات برمجيات تقوم بتقديم الدعم الأوتوماتيكي لبعض نظم معالجة التطوير.
Context diagram	مخطط تدفق البيانات النطاقي	رسم بياني لنطاق نظام المؤسسة والذي يظهر حدود النظام والكيانات الخارجية التي تتفاعل مع النظام وتدفقات البيانات المهمة بين النظام والكيانات الخارجية.
Coupling	الاقتران	اعتمادية الأنظمة الفرعية على بعضها البعض.
Critical path	المسار الحرج	أقل زمن ممكن لانهاء المشروع.
Critical path scheduling	جدولة الممرات الخطيرة	تقنية جدولية والتي تنظم وتؤقت ترتيب مهام الأنشطة والتي تؤثر مباشرة في زمن انهاء المشروع.
Data store	مخازن البيانات	بيانات ساكنة قد تأخذ شكل العديد من التمثيلات الحسية.
Data-flow diagram (DFD)	مخطط تدفق البيانات	رسم يوضح حركة البيانات بين الكيانات الخارجية والعمليات ومخازن البيانات داخل النظام.
Decomposition	التفكيك	هو عملية تفكيك النظام لمكونات أصغر، كذلك تعرف بمهمة التفكيك.
Deliverable		الناتج النهائي من مرحلة SDLC
Depicts	رسم	تصوير
Discount rate	معدل الخصم	هو معدل يستخدم لحساب نسبة التدفق المالي المستقبلي
Disruptive technologies	التقنيات القالبة	هي التقنيات التي تمكن من كسر قواعد العمل القديمة التي تمنع المنظمات من القيام بتغييرات جذرية.
Economic feasibility	الجدوي الاقتصادية	هي عملية تحديد الفائدة المادية والتكاليف المرتبطة بمشروع تطويري.
Electronic commerce	التجارة الالكترونية	الاتصال المبني على الانترنت لتدعيم أنشطة العمل يوماً بيوم
Electronic data interchange (EDI)	تبادل البيانات إلكترونيا	استخدام تكنولوجيا الاتصالات لنقل وثائق العمل بشكل مباشر بين المنظمات
Environment	البيئة	هي كل ما يحيط بالنظام من الخارج ويتفاعل مع النظام.
Feasability study	دراسة الجدوى	تحدد ما اذا كان نظام المعلومات منطقي بالنسبة لنقاط العمليات الاقتصادية
Formal system	النظام الرسمي	هو الطريقة الرسمية لعمل النظام كما وثقتها المنظمة

Gantt chart	مخطط Gantt	هو رسم يعرض مهام المشروع بشكل مستطيلات أفقية والتي طولها يكون نسبة الى زمن انهائها
Incremental commitment	الالتزام الاضافي	هي الاستراتيجية المستخدمة في تحليل النظم والتصميم الذي يجري فيه مراجعة المشروع بعد انتهاء كل مرحلة من المراحل، و مواصلة العمل بالمشروع تجد ما يبررها في كل مراجعة من هذه المراجعات.
Informal system	النظام غير الرسمي	هو الطريقة التي يعمل بها النظام فعلياً
Information system analysis and design	تحليل وتصميم نظم المعلومات	عملية المعالجة لتطوير وصيانة نظام المعلومات
Intangible benefits	الفائدة غير الملموسة	هي الفائدة الناتجة من ابتكار نظام معلومات والتي يتعذر حسابها فعليا بالدولارات.
Intangible cost	التكاليف غير الملموسة	هي تلك التكاليف الناتجة عن مشروع نظام المعلومات والتي يصعب تقييمها و حسابها فعليا بالدولارات
Interface	الواجهة	نقطة الاتصال بين النظام والبيئة وأيضا ملتقى الأنظمة الفرعية.
Internet	الإنترنت	شبكه عالمية كبيرة من الشبكات التي تستخدم بروتوكول شائع للاتصال مع بعضها البعض ؛ شبكه الكمبيوتر العالمية لتدعيم العمل من أجل التجارة الإلكترونية للعميل
Interrelated	الترابطية	اعتمادية جزء واحد من النظام على جزء أو أجزاء أخرى من النظام
Intranet	الانترانت	الاتصال المبني على الإنترنت لدعم أنشطة العمل من خلال منظمة فردية
JAD session leader	قائد جلسات التصميم التشاركي	هو فرد مُدرَّب يُعد لجلسات التصميم التشاركي ويقودها.
Joint application design (jad)	تصميم التطبيقات المشترك	هي عبارة عن معالجة هيكلية حيث يجتمع كل من مستخدمي النظام والمدراء والمحللين لعدة أيام ليعملوا على تحديد ومراجعة متطلبات النظام.
Key business processes	عمليات الأعمال الأساسية	ي مجموعة مهيكلة ومقيسة (قابلة للقياس) من الأنشطة المصممة لإنتاج مُخرج معين ولزبون أو سوق معينة.
Legal and contractul feasibility	الإمكانية القانونية والتعاقدية	عملية تقييم النتائج الممكنة القانونية والتعاقدية الناشئة عن بناء النظام.

Level-0 diagram	رسم المستوى – ٠	DFD يمثل العمليات وتدفقات البيانات ومخازن البيانات المهمة بمستوى عالي من التفصيل
Level-n diagram	مخطط المستوى – ن	هو مخطط DFD ناتج من ن تفكيك متداخل من سلسلة من عمليات فرعية من العملية بمخطط المستوى – ٠.
Modularity	الاجتزائية	هو تقسيم النظام إلى مجتزاءات متماثلة الأحجام.
Network diagram	المخطط الشبكي	رسم يشير الى المهام والعلاقة فيما بينها
One-time cost	تكاليف المرة الواحدة	هي التكاليف المرتبطة ببدء المشروع، وتطويره، أو بداية تطبيق نظام المعلومات.
Open-ended questions	الأسئلة المفتوحة	هي أسئلة تطرح في المقابلات (والاستبانات) وليس لها إجابات محددة مسبقاً
Operational feasibility	الجدوى التشغيلية	عملية تقييم مدي حل النظام المقترح لمشكلات المشروع أو اخذ ميزة فرص العمل
Outsourcing	التعهيد الخارجي	هو عملية إرجاء مسؤولية بعض أو كل تطبيقات نظم المعلومات للمنظمة إلى الشركات الخارجية.
Participatory design	التصميم المشترك	هي طرق لتطوير النظم مبتكرة في شمال أوروبا حيث أن المستخدمين والتحسينات التي تتم في حياتهم العملية هي عبارة عن بؤرة متوسطة.
PERT (program evaluation and review technique)	تقنية تقييم ومراجعة البرنامج	تقنية تستخدم التخمين والتوقع والواقع الزمني لحساب الزمن المتوقع لمهمة معينة.
Political feasibility	الامكانية السياسية	عملية تقدير كيفية رؤية حاملي الاسهم الرئيسية داخل المشروع للنظام المقترح
Present value		هي القيمة الحالية لتدفق المالي المستقبلي
Process	عملية	الأفعال أو الأعمال المنفذة على البيانات حتى تحول أو تخزن أو توزع.
Process modeling	نمذجة العمليات	هي تمثيل رسومي للعمليات والأفعال التي تلتقط وتؤثر في وتخزن و تنشر البيانات بين النظام والبيئة وبين مكونات النظام نفسه
Project	المشروع	هو مهمة مدروسة من عدة أنشطة مرتبطة للوصول الى هدف، وللمشروع بداية ونهاية.

Project charter	صك المشروع	وثيقة قصيرة من مرحلة عالية يضعها اصحاب القرار الداخليون والخارجيون للاعلان الرسمي عن التأسيس، وكتابة وصف بسيط للاهداف مفاتح النجاح، وأصحاب القرار.
Project closedown	إنهاء المشروع	المرحلة الأخيرة من عملية إدارة المشروع والتي تركز على ايصال المشروع الى النهاية
Project execution	تنفيذ المشروع	المرحلة الثالة من عملية ادارة المشروع حيث المخططات مبنية على الاولوية (بدء المشروع والتخطيط له) في وضع العمل
Project initiation	البدء بالمشروع	هو أول مرحلة من ادارة المشروع والتي تقوم الأنشطة فيها بتحديد حجم وهدف وتعقيدات المشروع، كما تكون اجرائيات تدعم أنشطة المشروع اللاحقة.
Project management	إدارة المشروع	معالجة منظمة من بدء وتخطيط وتنفيذ وإنهاء المشروع
Project manager	مدير المشروع	وهو محلل نظم يتمتع بمهارات الإدارة، القيادية، التخطيط، العلاقات مع الزبائن، ويكون مسؤول عن بدء وتخطيط وتنفيذ وإنهاء المشروع.
Project planing	التخطيط للمشروع	وهي المرحلة الثانية من عملية ادارة المشروع والتي تركز على تعريف واضح للأنشطة بصف مستقلة والعمل المطلوب انجازه من كل نشاط كمشروع مستقل
Project scope statement	بيان هدف المشروع	وثيقة معدة للعميل تصف ما سيجلبه المشروع وتمهيد عام على مستوي عالي لكل العمل المحتاج لاتمام المشروع.
Project worbook	دفتر أعمال المشروع	أوراق متوفرة على الشبكة لاجزاء المشروع من مدخلات ومخرجات واجرائيات ومقاييس والمستخدمة في تشغيل المشروع، اعتماد اعضاء جدد للفرق، والاتصال بالإدارة والزبائن وتعريف مشاريع مستقبلية ووضع مراجعة للمشاريع السابقة
Prototyping	الأسلوب التجريبي	بناء نسخه مصغرة من النظام المعلوماتي المرغوب.
Purpose	الغرض	هو الهدف الكلي أو عمل النظام
Rapid application design (RAD)	تطوير التطبيقات السريع	هو عبارة عن إنشاء منهجية جوهرية لتطوير النظم، وذلك للتقليل من الوقت اللازم لتصميم وتنفيذ نظم المعلومات.

Recurring cost	التكاليف المتكررة	هي التكاليف الناتجة عن الاستخدام والتطوير المستمر لنظام المعلومات.
Repository	المخزن	هي عبارة عن قاعدة بيانات تحتوي على كل الرسمات التوضيحية والأشكال والتقارير وبناء البيانات وتعريف البيانات والمعالجة والمنطق ومحتوى النظام. وذلك لتقديم ميكانيكيات لضم البيانات مع الأدوات للنظام وكذلك البيانات مع البيانات.
Resources	المصادر	أي شخص، مجموعة من الاشخاص، معدات، معدن استخدم في انجاز أي نشاط.
Schedule feasibility	الامكانية الجدولية	عملية تقييم مدي قدرة الهيكل الزمني المتاح وتواريخ الاكمال لكل النشاطات الرئيسية داخل المشروع علي مواجهة أخر وقت لانجاز العمل التنظيمي والارتباكات
Scribe	الناسخ	هو الشخص الذي يسجل ملاحظات تفصيلية عما يجري في جلسة التصميم التشاركي.
SDLC	دورة حياة تطوير النظام	سلسلة الخطوات المستخدمة لتعين أوجه التطوير لنظام المعلومات.
Slack time		حجم الوقت الذي يمكن تأجيل النشاط فيه من دون تأجيل المشروع.
Software application	برمجيات تطبيقية	مجموعة من الإجراءات المتبادلة المستخدمة في الأعمال، بالإضافة إلى الحدود المتماثلة حيث يعملون معا لأغراض معينة.
Source/Sink	مصدر/ مصب	أصل أو غاية البيانات. غالباً ما يشار إليهم بالكيانات الخارجية.
System	النظام	هو مجموعة من الإجراءات المترابطة، تستخدم في وظائف الأعمال، لخدمة هدف معين.
System analysis	محلل النظم	إن وظيفة المنظمة هي المسؤولة بشكل كبير عن تحليل وتصميم نظم المعلومات.
Systems design	تصميم النظام	مرحلة من مراحل دورة حياة تطوير النظام حيث يقوم محلل النظم بوصف للحلول البديلة للنظام الذي يكون تحت التطوير (logical design) ثم يقوم بتحويل الوصف إلى وصف تقني مفصل لكي يتم فهمه وانجاز بناء النظام من قبل المبرمجين.

Systems development methodology	منهجية تطوير النظم	عملية المعالجة العادية تتبعها المنظمة لتوصيل الخطوات الضرورية لتحليلها، وتصميمها، وانجازها وصيانة نظم المعلومات.
Systems implementation and design	تنفيذ النظم وتشغيلها	المرحلة الأخيرة من مراحل دورة حياة تطوير النظام (SLDC) حيث تقوم في هذه المرحلة بتحويل مواصفات النظم إلى نظام مجرب ثم تضعه في حيز الاستخدام شاملةً عملية التنفيذ (coding) والفحص والتنصيب وكذلك التحسينات اللازمة للحفاظ على سير عمل النظام وفائدته.
Tangible benefit	الفائدة الملموسة	هي الفائدة الناتجة من ابتكار نظام معلومات والتي يمكن حسابها فعليا بالدولارات.
Tangible cost	التكاليف الملموسة	هي تلك التكاليف الناتجة عن مشروع نظام المعلومات والتي يمكن تقييمها و حسابها فعليا بالدولارات.
Technical feasibility	الامكانية التقنية	عملية تقييم قدرة التنظيم المطور على بناء نظام مقترح
The Time Value of Money	قيمة النقود الزمنية	هي نسبة المقارنة بين النفقات الحالية والعائدات المستقبلية
Work breakdown structure	بنية تقسيم العمل	وهي عملية تقسيم المشروع إلى مهام قابلة للادارة والتنفيذ والتنظيم المنطقي لهم والذي يضمن سهولة الانتقال بين مهام المشروع.

قاموس الترجمة
Translation Dictionary

المصطلح الإنجليزي	المصطلح العربي	مرادفات
Systems design	تصميم النظم	
Systems planning and selecting	تخطيط واختيار النظم	
Abstract class	صنف مجرد	راجع مرادفات class أدناه
Abstraction	تجريد	
Action	فعل	
Active class	صنف نشط	راجع مرادفات class أدناه
Activity	نشاط	
Activity diagram	مخطط نشاط	
Activity state	حالة نشاط	
Actor	فاعل	
Aggregation	تجميع	
Alternative	بديل	
Alternative generation and selection	توليد واختيار البدائل	إنتاج واختيار البدائل
An instance of an association	مثال لرابطة	
Analysis	تحليل	
Analysis class	صنف تحليل	راجع مرادفات class أدناه
Analysis mechanism	آلية تحليل	
Analysis phase	مرحلة التحليل	
Analysis>s	المحللين	
Analyst	محلل	

Approach	طريقة	منهج، مدخل، مقاربة، تمهيد، أسلوب
Appropriate	ملائم	
Architectural	معماري	راجع مرادفات architecture أدناه
Architectural mechanism	آلية معمارية	راجع مرادفات architecture أدناه
Architecture	عمارة	بنية (بنك المصطلحات الموحدة)، بناء(الكيلاني)، بنيان (السورية)
Artifact	منتج	تُحفَة!، نتاج صُنعي، نتاج اصطناعي
Assessing economic feasibility	تقييم الجدوى الاقتصادية	
Assessing project feasibility	تقييم جدوى المشروع	
Assign	يعيّن	
Assistance	مساعدة	
Associated	مترابط	
Association	علاقة	
Association class	صنف علاقة	راجع مرادفات class أدناه
Attribute	صفة	خاصية، سمة
Automation	أتمتة	
Background	خلفية	
Balanced DFD	مخطط تدفق بيانات متوازن (DFD)	
Balancing	موازنة	
Baseline modules	المجتزءات القاعدية	راجع مرادفات module أدناه
Baseline project	خطة المشروع الأساسية	
Baseline project plan (BPP)	خطة المشروع القاعدية	
Batch	دُفعة	
Behavior	سلوك	
Beta testing	اختبار بيتا	

Bottom-up approach	طريقة أسفل-أعلى	صعودي، تصاعدي، من القاعدة إلى القمة. وتراجع مرادفات approach أعلاه
Boundary (adjective)	حدي أو حدودي	
Boundary (noun)	حد (حدود)	
Boundary class	صنف حد	راجع مرادفات class أدناه
Break-even analysis	تحليل التعادل	
Build routines	روتينات البناء	راجع مرادفات routine أدناه
Business	أعمال	أعمال تجارية، عمل، متاجرة، مشروع، مؤسسة، منشآت تجارية
Business case	مبررات مشروع	
Business process reengineering (BPR)	إعادة هندسة عمليات الأعمال	الهندرة، إعادة هندسة العمل الإداري، إعادة تقييم نظم العمليات
Business processes	عمليات الأعمال	العمليات المؤسسية
Calculated (or computed or derived) field	حقل مُشتقّ	
Candidate key	مفتاح مرشح	مفتاح للبحث
Capacity	سعة	
Capture	يلتقط	
Cardinalities	تعدادية	
Cardinality	التعدادية	
Characteristic	خاصية	
Check (noun)	فحْص	
Check (verb)	فَحَص	
Checklist	قائمة تفقد	
Class	صنف	صَفّ، طبقة، فصيلة، فئة، نوع
Class diagram	مخطط صنف	راجع مرادفات class أعلاه
Classifier	مصنف	
Close-ended questions	أسئلة مغلقة النهاية	
Code	كود	

Code generation	توليد الكود	إنتاج الكود، إصدار الكود
Coding	كتابة الكود	
Coherence	تلاصق	
Cohesion	تلاحم	
Collaboration	تعاون	
Collaboration diagram	مخطط تعاون	
Combined	مركب	
Compile	يترجم	يصرّف، يؤلّف
Component	مكوِنة	
Composite	مؤلَّف	
Composition	تكوين	
Compound	مركب	
Computer	حاسوب	حاسب، حاسب آلي
Computer display = display	شاشة الحاسوب	
Computer-aided software engineering (case)	هندسة البرمجيات بمساعدة الحاسوب	
Computer-based	معتمد على الحاسوب	
Computer-related	حاسوبي، مرتبط بالحاسوب	
Conceptual data modeling	نمذجة مفاهيمية للبيانات	
Conceptually	مفهومي	
Concrete	محدد، متماسك	ملموس، محسوس
Concurrency	تزامن	
Condition stubs	حيز الشرط	
Configuration	تشكيل	
Configuration management	إدارة التشكيل	
Connections	صلات	
Consistency	اتساق	
Consistent (ومشتقاتها الأخرى)	متسق	متساوق، متماسك، منسجم، ثابت، منتظم

Constraint	قيد	
Construct (noun)		
Context diagram	مخطط تأطيري	مخطط إطاري، مخطط نطاقي
Context-sensitive	حساس (حساسة) للسياق	
Contrasted	مغاير	
Control	تحكم	ضبط، تنظيم
Control (in windows & interfaces)	تحكمية	كائن تحكم
Control class	صنف تحكم	راجع مرادفات class أعلاه
Conversion	تحويل	
Cookie	كوكي	
Corrective maintenance	صيانة تصحيحية	
Coupling	الاقتران	وصل، تقارن
Customer relations management (CRM)	إدارة علاقات الزبائن	
Data	بيانات	معطيات
Data dictionary	معجم بيانات	قاموس بيانات. راجع مرادفات data أعلاه
Data flow	تدفق بيانات	راجع مرادفات data أعلاه
Data integrity	تكامل البيانات	راجع مرادفات data أعلاه
Data sink	مصبّ بيانات	راجع مرادفات data أعلاه
Data source	مصدر بيانات	راجع مرادفات data أعلاه
Data store	مخزن بيانات	راجع مرادفات data أعلاه
Data type	نوع بيانات	راجع مرادفات data أعلاه
Database schema	مخطط قاعدة بيانات	مختطة قاعدة بيانات. راجع مرادفات data أعلاه
Data-flow diagram (DFD)	مخطط تدفق بيانات	راجع مرادفات data أعلاه
Data-oriented approaches	طرق بياناتية التوجّه	طرق بياناتية. راجع مرادفات data أعلاه
Debug	يصوب الأخطاء	
Decision	قرار	
Decision support	مساندة القرار	

Decision support system	نظم مساندة القرار	
Decision table	جدول قرار (قرارات)	
Declarative	ايضاحي	
Decomposition	تفكيك	
Default	افتراضي	مغتفل، فرضي، أساسي، مبدئي
Deliverable	مستلمة	نواتج
Denormalization	نقض التطبيع	راجع مرادفات normalization أدناه
Dependency	اعتمادية	اعتماد (وأراها أفضل حتى من اعتمادية)، تبعيّة (وهي تستحق التأمل)
Deployment diagram	مخطط انتشار	
Deployment view	منظر انتشار	
Derived attribute	صفة مشتقة	راجع مرادفات attribute أعلاه
Descending priority	أولوية تنازلية	
Design	تصميم	
Design mechanism	آلية تصميم	
Design model	نموذج تصميم	
Design strategy	إستراتيجية تصميم	
Desk checking	الفحص المكتبي	
Development	تطوير	
Development project	مشاريع التطوير	
DFD completeness	تمام مخطط سيران بيانات (DFD)	
DFD consistency	اتساق مخطط سيران بيانات (DFD)	
Diagram	مخطط	
Diagramming	تخطيط (إعداد الرسومات التخطيطية)	تصوير، رسم المخططات، ترسيم
Dialogue	حوار	
Dialogue sequence	تتابع الحوار	

Digital	رقمي	
Digitize	يُرَقْمِن (يجعل أو يحول الشيء رقمياً)	يرقّم
Discipline	خطوة نظامية	
Display (noun)	شاشة	عرض، شاشة العرض، مظهار
Display (verb)	يعرض	يُظهر
Disruptive technology	تقنية قالبة	
Document	مستند	وثيقة
Documentation	توثيق	
Dynamic	ديناميكي (متحرك باستمرار)	متحرك، تحراكي
Economic feasibility	الجدوى الاقتصادية	
Effectiveness	فعّالية	
Efficiency	كفاءة	
Electronic commerce (ec)	تجارة إلكترونية	
Electronic data interchange (edi)	تبادل البيانات إلكترونيا	
Electronic performance support systems (epss)	نظم مساندة الأداء الإلكتروني	
Encapsulation	احتوائية	أقترح إعادة النظر في إلحاق «يّة» بآخرها وإن كان ما ورد في معاجم الكمبيوتر لم يتعد هذه الألفاظ: كبسلة، تغليف، تعليب، حوصلة
Engineering-based Techniques	أساليب هندسية	راجع مرادفات techniques أدناه
Enhanced information System	نظام المعلومات المطور	
Enhanced information system	نظام معلومات مُحسّن	
Enterprise	منشأة	
Enterprise resource planning (ERP)	نظم تخطيط موارد المنشآت	
Enterprise solutions software == ERP	برمجيات حلول المنشآت	

Entity	كينونة	
Entity class	صنف كينونة	
Entity instance	مثال كينونة	راجع مرادفات instance أدناه
Entity type	نوع كينونة	
Entity-relationship diagram	مخطط الكينونة–العلاقة	
Environment	بيئة	
Evaluate	يقيّم	
Evaluating	تقييم	
Event	حدث	
Execution	تنفيذ	
Executive	تنفيذي	
Executive information system (EIS)	نظم المعلومات التنفيذية (نظم معلومات الإدارة العليا)	
External documentation	التوثيق الخارجي	
External entity	كينونة خارجية	
Extracting	استخلاص	
Extranet	اكسترانت	
Extreme	قصوى	
Façade class	صنف واجهة	راجع مرادفات class أعلاه
Feasibility assessment	عملية تقدير الجدوى	
Feature driven development	تطوير مقود بالمزايا	
Fictional	خيالي	
Field	حقل	
File	ملف	
Firewall server	خادم الجدار الناري	
Firm	مؤسسة	مكتب
Flexible design	تصميم مرن	
Flow	تدفق	انسياب
Flowchart	خريطة تدفق	
Fluid	مائع	

Focus	تركيز	
Focus of Control	تركيز التحكم	
Forecasting	خطة	
Foreign key	مفتاح خارجي	
Form	نموذج	
Formal system	نظام رسمي	
Format (noun)	نَسَقٌ	هيئة، صيغة
Format (verb)	نسَّق	
Formatting	تنسيق	تهيئة، تشكيل
Forward engineering	هندسة أمامية	
Foundation	أساس	
Four-phased	رباعية الأطوار	
Framework	إطار عمل	
Function	وظيفة	
Function (in business, biology, ….etc)	وظيفة	
Function (in programming & math)	دالّة (برمجية أو رياضياتية)	
Functional dependency	اعتماد وظيفي	
Gap analysis	تحليل الفجوة	
Generalization	تعميم	
Generalization (The principle of generalization)	التعميمية	هلا أعدنا النظر في هذا؟
Generate	يولد	ينتج، يصدر
Generation	توليد	إنتاج، إصدار
Genius	عبقري	
Goal	هدف	
Growing	متنامي	
Guard condition	شرط حام (واقٍ)	
Guideline	خط إرشادي	
Guidelines	الخطوط الإرشادية	
Handle	يتناول	
Hardcopy	نسخة مطبوعة	
Hardware	عتاد	

Hashed file organization	تنظيم ملفات ملبّد	
Help	مساعدة	
Help desk	مكتب المساعدة	
Help screen	شاشة مساعدة	
Hierarchy	هرمي	
Homonym	مجانس	
Hypertext	نص تشعبي	نص فائق، نص مترابط
Hypertext-based	مبني على النصوص التشعبية	
Implementation	تنجيز	تنفيذ
Implementation mechanism	آلية تنجيز	
Implementation view	منظر تنجيزي	
Inconsistency	تضارب	عدم اتساق، تناقض، تعارض
Incremental commitment	التزام تزايدي	تعهد تدريجي
Index	فهرس	
Indexed file organization	تنظيم ملفات مُفَهرس	
Indifferent condition	شروط غير مؤثرة	
Individual	فرد	فردي، منفرد
Informal system	نظام غير رسمي	
Information	معلومات	
Information system	نظام المعلومات	
Information systems	نظم المعلومات	
Information Systems Analysis and Design	تحليل وتصميم نظم المعلومات	
Inheritance	وراثة	توريث
In-house development	التطوير الداخلي	
Input	مُدخل، مُدخلات	دَخْل
Input (verb)	يُدخل	
Inquiry	استعلام	
Insignificant	غير هام	
Inspect	تَفَحَّصَ	

Inspection	تَفَحُّص	
Install	يُنصّب	يركب
Installation	تنصيب	تركيب
Instance	مثال	منتسخ، حالة، طبعة
Instantiate	يستمثل	يستنسخ
Instantiated	مستمثل	مُستنسَخ
Intangible benefits	فوائد غير ملموسة	
Integration testing	اختبار التكامل	
Interaction diagram	مخطط تفاعل	
Interchangeable	قابل للتبادل	
Interdependency	اعتماد بيني	توقف شيء على آخر
Interface	واجهة	
Internal documentation	التوثيق الداخلي	
Internet	الإنترنت	
Internet connection	اتصال بالإنترنت	
Interpersonal	اتصالية	
Interrelated	مترابطة بينياً	
Intranet	إنترانت	
Inventory	مخزون	
Inventory Control System	نظام لمراقبة المخزون	
Inventory control system	نظام مراقبة مخزون	
Inventory management	إدارة المخزون	
Irreducible	لا اختزالي	
Iteration	تكرار	
Joint (adjective)	متصل، متحد، مشترَك	
Joint (noun)	وصلة، مفصل	
Joint Application Design [or Development] (JAD)	تصميم [أو تطوير] تطبيقات مشترك	
Key business process	عمليات المنشأة الأساسية	
Key processes	العمليات الأساسية	

Keystroke	ضربة مفتاح	
Label	لافتة	
Layering	تشكيل الطبقات، طَبْقَنَة (كلاهما من معجم المعلوماتية السوري)	ترتيب الطبقات (الكيلاني)، الترتيب على طبقات (معجم شعاع)، تتابع الطبقات، ترقيد، تنضيد
Legal and contractual feasibility	الإمكانية القانونية والتعاقدية	
Level-0 diagram	مخطط المستوى	
Level-n diagram	مخطط المستوى ن	
Life cycle	دورة الحياة	
Lifeline	خط حياة	
Likely alternative	انسب البدائل	
Link	وصلة	
Localization	توطين	
Logical DFD	مخطط تدفق بيانات منطقي	
Logical view	منظر منطقي	
Maintainability	صيانية	سهولة الصيانة
Maintenance	صيانة	
Management information system	نظم المعلومات الإدارية	
Mandatory	إجباري	
Manifesto	بيان	
Manipulate	يعالج	
Manipulation	معالجة	
Manual system	نظام يدوي	
Mapping	تقابل	
Mastering	اتقان	
Mean time between failures (MTBF)	متوسط الفترة بين عُطلَيْن	الزمن الوسطي بين الأعطال
Media (plural)	وسائط	
Medium (singular)	وسط	

Message	رسالة	
Meta-data	بيانات ورائية	
Method	طريقة	منهج، نهج
Methodology	منهجية	
Microfilm	مصغر فلمي	
Middleware software	برمجيات وسطيّة (وسطى)	
Milestones	معالم	
Modeling	نمذجة	
Modular	اجتزائي	تجزيئي. راجع مرادفات module أدناه لاشتقاق مرادفات أخرى
Modularity	الاجتزائية	التجزيئية (قابلية التجزيء). راجع مرادفات module أدناه لاشتقاق مرادفات أخرى
Modularization	جزأنة	راجع مرادفات module أدناه لاشتقاق مرادفات أخرى
Module	مجتزأ	وحدة متكاملة، تركيبة، نسيقة، وحدة نمطية، وحدة جزئية، قالب، معيار
Monitor (as in monitor program…etc)	مُراقب	
Monitor (for displaying output)	شاشة	مِرْقاب
Multiple inheritance	وراثة متعددة	
Multiplicity	تعددية	
Multivalued attribute	صفة متعددة القيم	راجع مرادفات attribute أعلاه
Navigability	الملاحية	
Navigable	صالح للملاحة	
Node	عقدة	
Non-interruptible	لامستوقف	
Normal form	شكل طبيعي	
Normalization	تطبيع	استنظام، تسوية
Notation	تمثيل	

Nuclear	نووي	
Numeric	عددي	
Numerical data	بيانات عددية	
Object	كائن	
Object technology	تكنولوجيا كائنية	
Object-orientation (OO)	توجه كائني	
Object-oriented	كائني	كائني، شيئي، كائني التوجه، شيئي التوجه، كائني المنحى، شيئي المنحى
Object-oriented approaches	طرق كائنية	
Occurrence	ظهور	
Off-the-shelf	جاهز	
One-time cost	تكاليف المرة الواحدة	
Ongoing	استمرار	
Online	مباشر (عَلْخَطّ)	
Open-ended questions	أسئلة مفتوحة النهاية	
Open-source Software	برمجيات مفتوحة المصدر	
Operating environment	بيئة التشغيل	
Operation	عملية	
Operation signature	توقيع عملية	
Operational feasibility	الإمكانية التشغيلية	
Orders	أوامر	
Organization	منظمة	
Organizational chart	مخطط تنظيمي (هيكل إداري)	
Organizational problem	مشكلة مؤسسية	
Organizational process	عملية مؤسسية (إدارية)	
Organizational system	نظام مؤسسي (إداري)	
Outcome	ناتج	حصيلة، نتيجة
Outline	خطوط عريضة	
Output (noun)	مخرج (مخرجات)	خرْج

Output (verb)	يخرج	
Outsourcing	تعهيد خارجي	
Package	حزمة	رزمة
Participatory design	التصميم التشاركي	
Pattern	نمط	
Payroll	كشف الرواتب	
Perfective maintenance	الصيانةُ المتممة	
Perform	أداء	
Persistence	ديمومة	
Phase	طوْر	
Physical	واقعي	مادي، فعلي، حقيقي
Physical DFD	مخطط تدفق بيانات واقعي	
Physical file	ملف واقعي	
Physical table	جدول واقعي	
Planning	تخطيط	
Platform	منصّة	
Pointer	مؤشر	
Policy	سياسة	نهج
Political feasibility	الإمكانية السياسية	
Polymorphism	تعددية الشكل	
Predefined	مسبقة التعريف	
Preventive maintenance	الصيانةُ الوقائية	
Primary	أولي	
Primary key	مفتاح أولي	
Primitive DFD	مخطط تدفق بيانات أولي	
Priority	أولوية	أفضلية، أسبقية
Procedure	إجراء	
Procedure (in programming)	إجراء (إجرائية)	
Process	عملية	
Process modeling	نمذجة العمليات	
Process view	منظر عملية	

Processing	معالجة	
Process-oriented approaches	طرق عملياتية التوجّه	
Project	مشروع	
Project initiation	استهلال المشروع	
Project scope statement	غرض المشروع	
Promote	يعلي – يرقي	
Property	خاصية	
Proponent	مؤيد – مناصر	
Proposed	ينوي	
Prototype	نموذج تجريبي	
Prototyping	النمذجة التجريبية	
Query	استفسار	
Rapid application development (RAD)	تطوير التطبيقات السريع	
Realization	إدراك	
Recovery	استرداد	
Recurring cost	تكاليف متكررة	
Recursive foreign key	مفتاح خارجي ارتدادي	
Refactoring	إعادة التقسيم	
Referential integrity	التكامل المرجعي	
Refinement	تصفية	
Reflect	يعكس	
Relation (in math and relational databases)	علاقة (رياضياتية أو في قواعد البيانات العلاقاتية)	
Relational database	قاعدة بيانات علاقاتية	قاعدة بيانات علاقية
Relational database model	نموذج قاعدة بيانات علاقاتية	راجع مرادفات relational database أعلاه
Relationship	علاقة	
Release	نشر	
Reliability	اعتمادية	وثوقية ثبات (مجمع اللغة باقاهرة)، موثوقية، جدارة (!)

Replaceable	ممكن الاستبدال	
Replacement system	النظام البديل	
Replicable	تكراري	
Replication	إعادة نسخ	
Report	تقرير	
Repository	مستودع	
Requirements	متطلبات	
Requirements determination	تحديد المتطلبات	
Responsibility	مسؤولية	
Restrictions	قيود	
Reverse engineering	هندسة عكسية	
Rigorous	صارم	
Role	دور	
Router	موجه	
Routine	روتين	برنامج محدد
Scenario	سيناريو	
Schedule feasibility	الإمكانية الجدولية	
Secondary key	مفتاح ثانوي	
Selecting the best alternative design strategies	واختيار أحسن إستراتيجية للتصميم البديل	
Selection	اختيار	
Semantic	دلالي	
Semaphore	علم اشارات	
Sequence diagram	مخطط تسلسل	
Sequential file organization	تنظيم ملفات تتابعي	
Simultaneous	متزامن	
Single inheritance	وراثة فريدة	
Software	برمجيات	
Software application testing	اختبار تطبيق البرمجيات	
Software plug-ins	إضافات البرمجيات	

Spawned	منتج	
Spreadsheet	لوحة جدولية	جدول محاسبي
Stable	ثابت	
Standard	قياسي	قياسي، معياري
Standardization	مَقْيسة	معيَرة
State	حالة	
Statechart diagram	مخطط خريطة حالة	
Static	ساكن	
Status	حالة	
Steady	ترسخ	
Stereotype	تصفيحية	
Structure	هيكل	
Structured	هيكلي	مهيكل، بنيوي
Structured approach	طريقة هيكلية	
Structured english	الإنجليزية الهيكلية	
Structured walkthrough	مسير هيكلي	
Structuring	هيكلة كذا	
Stub testing	اختبار الجذل	
Stylized	مُأسلب	
Subclass	صنف فرعي	راجع مرادفات **class** أعلاه
Subordinate	ثانوي	
Subprograms	برنامج فرعي	
Subsystem	نظام فرعي	
Superclass	صنف كلي	راجع مرادفات class أعلاه
Sustainable	مستدام	
Swimlane	مسارسباحة	
Synchronous	متزامن	
Synonyms	مرادف	
System analysis	تحليل النظم	
System librarian	أمين مكتبة النظام	
System scope	نطاق النظام	

System testing	اختبار النظام	
Systems analysis	تحليل النظم	
Systems analyst	محلل نظم	
Systems design	تصميم النظم	
Systems development methodology	منهجية تطوير النظم	
Systems development life cycle (sdlc)	دورة حياة تطوير النظم	
Systems development life cycle sdlc	دورة حياة تطوير النظام	
Systems documentation	توثيق النظم	
Systems implementation and operation	تشغيل وتطبيق النظم	
Taxonomic	تصنيفي	
Technical feasibility	الإمكانية التقنية	
Technique	أسلوب	طريقة، تقنية، فَنيّة (الأخيرتان هما فقط المعتمدتان في بنك المصطلحات الموحدة)
Template	قالب	
Template-base	مُقَولب	
Test case	حالة اختبار	
The Time Value of Money (TVM)	قيمة النقود الزمنية	
Thread	خيط	مُسلك (في تراكيب البيانات)، سِلْك، نَيْسب (في البرمجة وفي مجموعات الأخبار)
Threaded		متعاقب، مُسلَك، مُنيْسَب
Threading		نَيْسبة، إسلاك، تعاقب
Timescale	المدة	
Top-down approach	طريقة أعلى–أسفل	نزولي، تنازلي، فوقي، فوق – تحت، عام – خاص. وتراجع مرادفات approach أعلاه
Toxonomic	تصنيفي	
Traditional	تقليدي	
Transaction	عملية جارية	معاملة، واقعة، مداولة
Transaction processing systems	نظم معالجة العمليات الجارية	

Transition	انتقال	
Trend	الاتجاه	
Uncommon	غريب – نادر	
Underlying	ضمني	
Understood	مفهوم	
Undisciplined	غير منظم	
Unified modeling language (uml)	لغة النمذجة الموحدة	
Unit testing	اختبار الوحدة	
Unrelated	غير متصل	
Usability	الاستخدامية	قابلية الاستخدام، سهولة الاستخدام
Use case	حالة استخدام	
Use case model	نموذج حالة استخدام	
Use case realization	ادراك حالة استخدام	
Use case view	منظر حالة استخدام	
User	مستخدم	مستفيد
User documentation	توثيق المستخدم	
Utility class	صنف منفعة	صنف خِدْمِيّ. راجع مرادفات class أعلاه
Utilizing (utilization)	انتفاع	استخدام
Validating	تثبُّت من الصحة (تصحيح)	
Verification	إثبات	
Virtual	افتراضي	
Visibility	ارتآئية (وأرى أنه لا بأس بكلمة «رؤية» وإن كانت أيضا تستخدم مقابل vision)	رؤية، وضوح الرؤية
Visual modeling	نمذجة مرئية	
Walkthrough	مسير	
Web server	خادم الويب	

Web site maintenance	صيانة موقع الويب	
Well-structured relation	علاقة حسنة الهيكلة	
Word processor	معالج نصوص	معالج كلمات، منسق كلمات
Work breakdown structure	هيكل تقسيم العمل	
Workflow	انسياب العمل	دفق الأعمال، انسياب العمليات
Working version	نسخة عاملة	

كشاف الموضوعات
Index

أ

الاتساقية (Consistency) ٢٠٨

الاجتزائية (Modularity) ١٢

الإحتوائية (Encapsulation) ٥٠٩

إختبار التكامل (Integration Testing) ٤٥٦

اختبار الجذل (Stub Testing) ٤٥٧

اختبار ألفا (Alpha Testing) ٤٥٩

اختبار القبول (Acceptance Testing) ٤٥٩

اختبار النظام (System Testing) ٤٥٦

اختبار الوحدة (Unit Testing) ٤٥٦

اختبار بيتا (Beta Testing) ٤٥٩

اختلاف الشروط (Indifferent Condition) ٢١٨

إدارة التشكيل (Configuration Management) ٤٨٢

الأدوات (Tools) ٧

أدوات هندسة البرمجيات بمساعدة الحاسوب (CASE Tools) ٣٤

الأدوار (Roles) ٦

إزالة التطبيع (Denormalization) ٤١٥

الأسئلة المغلقة (Closed-Ended Questions) ١٥٦

الأسئلة المفتوحة (Open-Ended Questions) ١٥٥

الأساليب (Techniques) ٧

إستراتيجية للتصميم (Design Strategy) ٢٩٤

الأسلوب التجريبي (Prototyping) ٣٣

اعادة التقسيم (Refactoring) ٥٣٧

الاعتماد الوظيفي (Functional Dependence) ٣٩٤

الأفراد (Users) ٦

الإقتران (Coupling) ١٢

الاكتمالية (Completeness) ٢٠٧

إكتمالية مخطط DFD Completeness (DFD) ٢٠٧

الالتزام التزايدي (Incremental commitment) ١٠٠

الامكانية التشغيلية (Operational Feasibility) ١١٥

الامكانية التقنية (Technical Feasibility) ١١٦

الامكانية الجدولية (Schedule Feasibility) ١١٦

الامكانية السياسية (Political Feasibility) ١١٦

الامكانية القانونية والتعاقدية (Legal and Contractul Feasibility) ١١٦

أمين مكتبة النظام (System librarian) ٤٨٢

الإنجليزية المهيكلة (Structured English) ٢١٤

أنظمة مساندة الأداء الالكتروني (Electronic Performance Support Systems) ٤٦٩
إنهاء المشروع (Project Close Down) ٧٠
أنواع البيانات (Data type) ٤١٠

ب

البرمجه القصوى (eXtreme Programming) ٥٢٩
برمجيات تطبيقية (Application Software) ٥
برمجيات حلول المنشآت (ERP) ٢٠
برمجيات مفتوحة المصدر (Open-Source Software) ٢١
بنية تقسيم العمل (Work Breakdown Structure) ٦١
بيئة (Environment) ٩

ت

تبادل البيانات الكترونيا (EDI) ١١٦
التجارة الالكترونية (Electronic Commerce) ١٢٦
التجميع (Aggregation) ٥١٣
التحكمات (Controls) ٦
تحليل الفجوه (Gap Analysis) ٢١٠
التحليل المتعادل (Break-Even Analysis) ١١٣
تحليل النظم (Systems Analysis) ٢٩
تحليل وتصميم نظم المعلومات (Information Systems Analysis and Design) ٥
تخطيط الحوار (Dialogue Diagramming) ٣٦٦
التخطيط للمشروع (Project Planning) ٥٩
تخطيط واختيار النظم (Systems Planning and Selection) ٢٨
التدريب والمساندة (Support) ٤٦٧
التصميم التشاركي (Participatory Design) ٣٧
تصميم التطبيقات المشترك (JAD) ٣٥
التصميم المبسط (Simple design) ٥٣٦
تصميم النظم (Systems Design) ٢٩
التطبيع (Normalization) ٣٩٢
تطوير التطبيقات السريع (RAD) ٣٥
التطوير الداخلي (In-House Development) ٢٢
التعدادية في العلاقة (Cardinalities in Relationships) ٢٦٥
تعدد (Multiplicity) ٥٠٩
تعديلات خط قاعدي (Baseline Modules) ٤٨٢
التعميم (Generalization) ٥١١
التعهيد الخارجي (Outsourcing) ١٦
تغذية راجعة (Feedback) ٣٦٠
التفعيل (Activation) ٥١٦
التفكيك (Decomposition) ١١
التفكيك الوظيفي (Functional Decomposition) ٢٠١
التقرير (Report) ٣٣٦
التقنيات القالبة (Disruptive Technologies) ١٧٤
تقنية مراجعة تقويم برنامج (PERT) ٧٥
التكاليف المتكررة (Recurring Cost) ١١٠
تكاليف المرة الواحدة (One-Time Cost) ١١٠
التكاليف الملموسة (Tangible Cost) ١٠٩
التكاليف غير الملموسة (Intangible Cost) ١٠٩
التلاحم (Cohesion) ١٣
تنجيز النظم وتشغيلها (Systems Implementation and Operation) ٣١
التنصيب (Installation) ٤٦٠
التنصيب المباشر (Direct Installation) ٤٦٠
التنصيب المتوازي (Parallel Installation) ٤٦٠
التنصيب المرحلي (Phased Installation) ٤٦٠

التنصيب على موقع واحد (Phased Installation) ٤٦٠
تنظيم الملف (File Organization) ٤١٨
تنظيم الملف التسلسلي (Sequential File Organizations) ٤٢٠
تنظيم الملف الخليط (Hashed File Organizations) ٤٢٢
تنظيم الملف المفهرس (Indexed File Organizations) ٤٢٠
تنفيذ المشروع (Project Execution) ٦٦
تهيئة المشروع (Project Initiation) ٥٦
التوثيق (Documentation) ٦
توثيق المستخدم (User Documentation) ٤٦٤
توثيق النظام (System Documentation) ٤٦٤

ج

جدول القرار (Decision Table) ٢١٧
الجدول المادي (Physical Table) ٤١٥
جدوى اقتصادية (Economic Feasibility) ١٠٧

ح

حالة الاستخدام (Use Case) ٥٠٤
الحدث (Event) ٥١٤
حدود (Boundaries) ٩
الحقل (Field) ٤١٠
الحوار (Dialogue) ٣٦٣

خ

خطة المشروع القاعدية (Baseline Project Plan) ١٠٢

د

الدرجة (Degree) ٢٦٣
دور الارتباط (Association Role) ٥٠٩
دورة حياة تطوير النظم (Systems Development Life Cycle) ٢٦

ر

الرسالة البسيطة (Simple Message) ٥١٧
الرسالة المتزامنة (Synchronous Message) ٥١٧
رسم المستوى-٠ (Level-0 Diagram) ١٩٨

س

سلوك (Behavior) ٥٠٧

ص

الصفة (Attribute) ٢٠٢
الصفة المتعددة القيم (Multivalued Attrributes) ٢٦١
الصنف التجريدي (Abtract Class) ٥١٢
صنف الكائن (Object Class) ٥٠٧
الصيانة (Maintenance) ٤٧٧
الصيانةُ الإصلاحية (Corrective Maintenance) ٤٧٧
الصيانةُ التكيفية (Adaptive Maintenance) ٤٧٧
الصيانةُ المتممة (Perfective Maintenance) ٤٧٧
الصيانةُ الوقائية (Preventive Maintenance) ٤٧٨

ض

ضبط القيمة المجهولة (Null Value Control) ٤١٤

ط

الطرق (Methodologies) ٧
طريقة تطوير النظم (SystemsDevelopmentMethodology) ٢٦

ع

العتاد (Hardware) ٦

العلاقات (Relationships) ٢٦٢

العلاقة (Relational) ٣٩١

العلاقة الثلاثية (Ternary Relationship) ٢٦٥

العلاقة الثنائية (Binary Relationship) ٢٦٤

علاقة الدرجة الأولى (Unary Relationship) ٢٦٣

العلاقة جيدة الهيكلة (Well-Structured Relations) ٣٩٢

العمليات (Operation) ٥٠٨

عمليات الأعمال الأساسية (Key Business Processes) ١٧٢

عملية تحديد المتطلبات (Process of Determining Requirements) ١٤٩

عمود الشرط (Condition Stubs) ٢١٧

عمود الفعل (Action Stubs) ٢١٧

عمود القواعد (Rules Stubs) ٢٠٣

غ

غرض (Purpose) ٩

ف

الفائدة الملموسة (Tangible Benefit) ١٠٧

فتات كوكي (Cookie Crumbs) ٣٧٢

الفحص (Inspection) ٤٥٤

الفحص المكتبي (Desk Checking) ٤٥٥

الفوائد غير الملموسة (Intangible Benefits) ١٠٨

ق

قائد الجلسات (JAD Session Leader) ١٦٦

قاعدة البيانات العلاقية (Relational Database) ٣٠٣

قناع المدخل (Input Mask) ٤١٣

قوالب HTML Template-based (HTML) ٣٧٤

القيمة الافتراضية (Default Value) ٤١٣

القيمة الزمنية للنقود (Time Value of Money) ١١١

قيود (Constraints) ١٠

ك

كائن (Object) ٥٠٧

الكينونات الرابطة (Associative Entities) ٢٦٦

الكينونة (Entity) ٢٥٦

ل

لغة النمذجة الموحدة (UML) ٥٠٢

م

المؤشر (Pointer) ٤١٩

المتجانسات (Homonyms) ٤٠٥

مترابطة (Interrelated) ٩

متوسط الوقتِ بين كل فشل (MTBF) ٤٨٠

محلل النظم (System Analyst) ١٤

مخزن البيانات (Data Store) ١٩٤

المخطط الشبكي (Network Diagram) ٦٤

مخطط الصنف (Class Diagram) ٥٠٧

المخطط المتسلسل (Sequence Diagram) ٥١٦

مخطط المثال (Instance Diagram) ٥٠٨

مخطط المكونات (Component Diagram) ٥٢٠

مخطط النشر (Deployment Diagram) ٥٢٠

مخطط تدفق البيانات (Data-Flow Diagram) ١٩١

مخطط تدفق البيانات التأطيري (Context Data Flow Diagram) ١٩٦
مخطط حالة الاستخدام (Use – Case Diagram) ٥٠٤
مخططات DFDs الأولية (Primitive DFD) ٢١٠
مدير الأنظمة (Systems Manager) ٥١
المرادفات (Synonyms) ٤٠٤
المسار الحرج (Critical Path) ٧٩
المشروع (Project) ٥١
مصادر البرمجيات (Sources of Software) ١٧
المصدر/ مصب (Source/Sink) ١٩٤
معدل الخصم (Discount Rate) ١١٢
المعرف (Identifiers) ٢٥٩
المفتاح الأساسي (Primary Key) ٣٨٨
المفتاح الخارجي (Foreign key) ٣٩٧
المفتاح المرشح (Candidate Key) ٢٥٩
المقابلة والإنصات (Interviewing and Listening) ١٥٣
مكتب المساعدة (Help Desk) ٤٧٢
مكونة (Component) ٩
الملف المادي (Physical File) ٤١٨
الممثل (Actor) ٥٠٤
منتجو البرامج المحزمة (Packaged Software Producers) ١٩
المنهجية الرشيقة (Agile Methodologies) ٥٢٦
الموازنة (Balancing) ٢٠٤

ن

الناسخ (Scribe) ١٦٦
النظام (System) ٨
النظام الرسمي (Formal System) ١٦٢
النظام غير الرسمي (Informal System) ١٦٢
نظم المعلومات الإدارية (MIS) ٢٤
نظم دعم القرار (DSS) ٢٤
نظم معالجة العمليات (TPS) ٢٣
نمذجة العمليات (Process Modeling) ١٩١
النموذج (Form) ٣٣٥
نموذج (Gantt (Gantt Chart ٦٢
نموذج التطبيع القياسي الثالث (3NF) ٣٩٦
نموذج التطبيعي الثاني (2NF) ٣٩٥

هـ

هيكلة المتطلبات (Requirements Structuring) ١٥٢

و

واجهات (Interfaces) ١٠
الوحدة المرجعية (Referential Integrity) ٣٩٧

نبذة عن المترجم

تخصص أ.د. عبدالله الغامدي في مجال هندسة البرمجيات وتحديدا «تحليل وتصميم النظم» لأكثر من عشرين عاما حيث تركزت دراساته العليا من الماجستير والدكتوراه ودراسات ما بعد الدكتوراه في نفس المجال، كذلك قام بنشر العديد من الأبحاث في مجلات ومؤتمرات عالمية تخصصت جلها في مجال «تحليل وتصميم النظم»، كذلك قام بإدارة العديد من المشاريع الاستشارية لتطوير نظم المعلومات لجهات حكومية وخاصة مما أثرى خلفيته التطبيقية في مجال «تحليل وتصميم النظم».

أمضى الأستاذ الدكتور عبدالله الغامدي في تدريس مادة «تحليل وتصميم النظم» لطلاب كلية علوم الحاسب والمعلومات ما يربو عن ثلاثة عشر عاما ولا زال مشرفاً على تقديم نفس المقرر وما يشمل ذلك من توصيف وتطوير محتوى المادة المقدمة، استخدم خلال تلك الفترة العديد من الكتب الأجنبية واستقر خلال الست سنوات الأخيرة على هذا الكتاب «أساسيات تحليل وتصميم النظم» لما له من مزايا معرفية وتعليمية.

يشغل الأستاذ الدكتور عبدالله الغامدي حاليا منصب رئيس قسم هندسة البرمجيات بكلية علوم الحاسب والمعلومات بجامعة الملك سعود.