

تحليل وتصميم النظم 2 -

شريحة مقتبسة من:

جيفري أ. هوفر ، جوي ف. جورج ، جوزيف س. فالاتسيش
(تحليل وتصميم النظم الحديثة ، الإصدار السابع ، بيرسون برنتيس هول)

الفصل 1

تطوير الأنظمة
بيئة

أهداف التعلم

تحديد تحليل وتصميم نظم المعلومات.
□ وصف دورة حياة تطوير نظم المعلومات

(SDLC).

□ شرح أدوات التطوير السريع للتطبيقات (RAD) وهندسة البرمجيات بمساعدة الكمبيوتر (CASE).

□ وصف المنهجيات الرشيقية والبرمجة المتطرفة.

□ شرح التحليل والتصميم الموجه للكائنات والعملية العقلانية الموحدة (RUP).

مقدمة

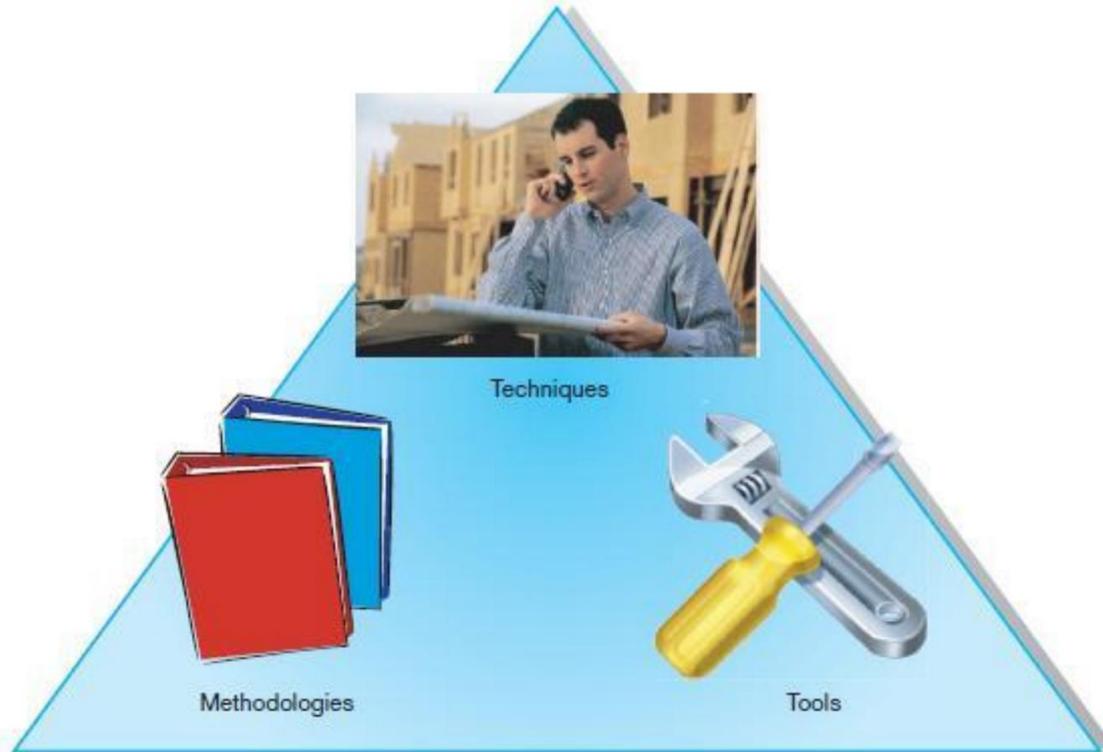
□ تحليل وتصميم نظم المعلومات

□ عملية تنظيمية معقدة

□ تستخدم لتطوير وصيانة نظم المعلومات الحاسوبية

□ يستخدمها فريق من المتخصصين في مجال الأعمال والأنظمة

مقدمة (تابع)



الشكل : 1-1 نهج تنظيمي لتحليل النظم و
التصميم مدفوع بالمنهجيات والتقنيات والأدوات

مقدمة (تابع)

□ برامج التطبيقات

□ برامج الكمبيوتر المصممة لدعم الوظائف أو العمليات التنظيمية

□ محلل نظم

الدور التنظيمي الأكثر مسؤولية عنه
تحليل وتصميم نظم المعلومات

نهج حديث للأنظمة التحليل والتصميم

□ الخمسينيات: التركيز على التشغيل الآلي الفعال
للعمليات الحالية

□ ظهور الثالث الإجمالي
لغات الجيل (3GL) أجهزة كمبيوتر أسرع وأكثر موثوقية

سبعينيات القرن الماضي: أصبح تطوير النظام أقرب إلى تخصص
هندسي

نهج حديث لتحليل وتصميم النظم (تابع)

الثمانينيات: طفرة كبيرة مع 4GL وأدوات CASE والأشياء الموجهة للكائنات

□ التسعينيات: التركيز على تكامل النظام وتطبيقات واجهة المستخدم الرسومية ومنصات العميل / الخادم والإنترنت

□ القرن الجديد: تطبيقات الويب

التطوير ، وأجهزة المساعد الرقمي الشخصي والهواتف الذكية اللاسلكية ، والتطبيقات القائمة على المكونات ، ومقدمي خدمات التطبيقات (ASP)

تطوير نظم المعلومات

□ منهجية تطوير النظام هي عملية قياسية يتم اتباعها في منظمة لإجراء جميع الخطوات اللازمة لتحليل نظم المعلومات وتصميمها وتنفيذها وصيانتها.

حياة تطوير النظم دورة (SDLC)

المنهجية التقليدية المستخدمة لتطوير وصيانة واستبدال
نظم المعلومات

□ مراحل في SDLC:

□ التخطيط

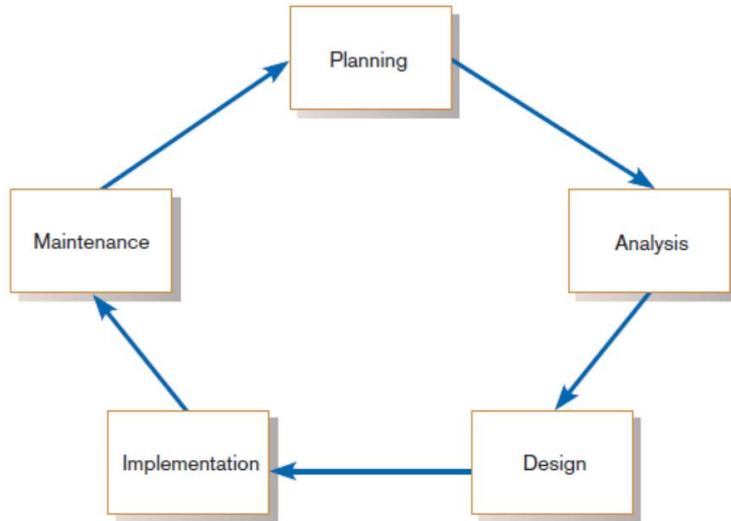
□ التحليل

□ التصميم

□ التنفيذ

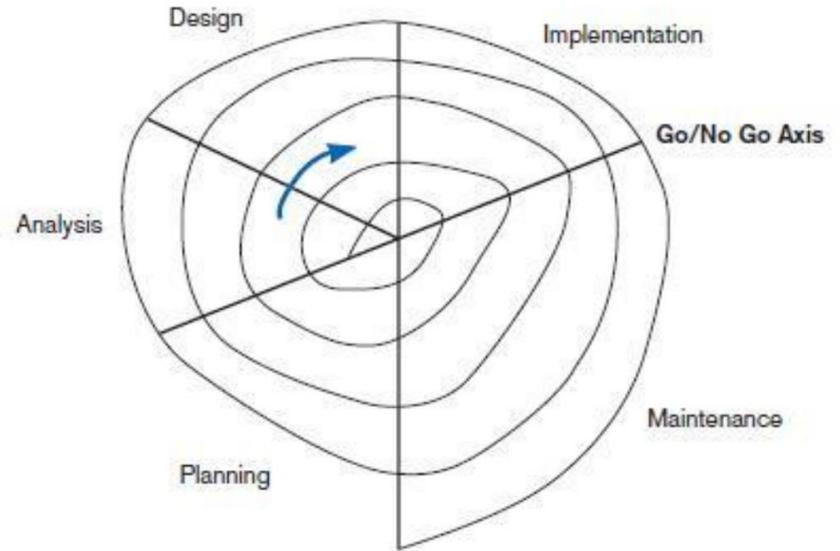
□ الصيانة

وجهات النظر القياسية والتطويرية من SDLC



الشكل 1-2

دورة حياة تطوير النظم



الشكل 1-3 النموذج التطوري

دورة حياة تطوير الأنظمة (SDLC)(تابع)

التخطيط - إجمالي المنظمة

يتم تحديد احتياجات نظام المعلومات وتحليلها وترتيبها
حسب الأولوية وترتيبها

□ التحليل - يتم دراسة وتنظيم متطلبات النظام

□ التصميم - يتم تحويل وصف الحل الموصى به إلى
مواصفات نظام منطقي ثم فيزيائي

دورة حياة تطوير الأنظمة (SDLC)(تابع)

□ التصميم المنطقي - يتم وصف جميع الميزات الوظيفية للنظام المختار للتطوير في التحليل بشكل مستقل عن أي نظام أساسي للكمبيوتر التصميم المادي - يتم تحويل المواصفات المنطقية للنظام من التصميم المنطقي إلى التفاصيل الخاصة بالتكنولوجيا التي يتم من خلالها إنشاء جميع البرمجة وبناء النظام يمكن إنجازه

دورة حياة تطوير الأنظمة (SDLC)(تابع)

□ التنفيذ - يتم ترميز نظام المعلومات واختباره وتثبيته ودعمه في المنظمة

□ الصيانة - يتم إصلاح نظام المعلومات وتحسينه بشكل منهجي

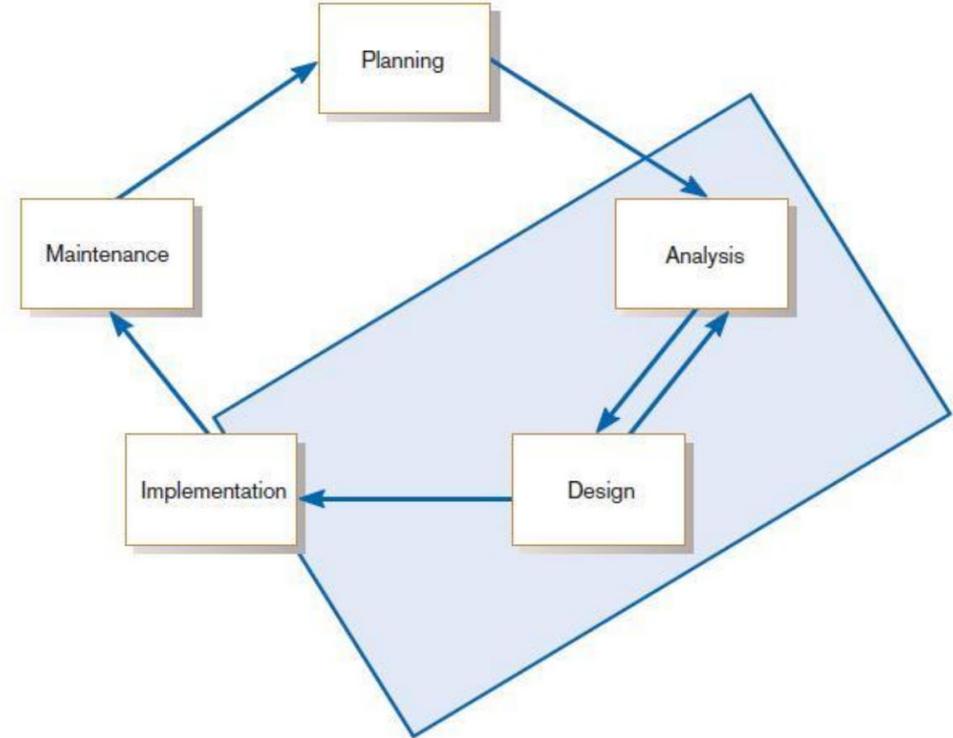
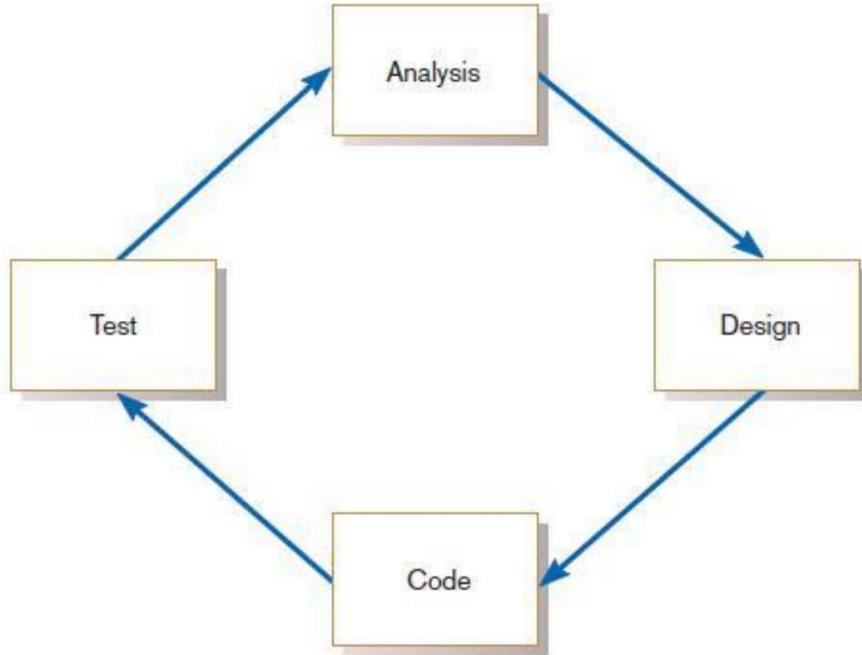
TABLE 1-1 Products of SDLC Phases

Phase	Products, Outputs, or Deliverables
Planning	<p>Priorities for systems and projects; an architecture for data, networks, and selection hardware, and information systems management are the result of associated systems</p> <p>Detailed steps, or work plan, for project</p> <p>Specification of system scope and planning and high-level system requirements or features</p> <p>Assignment of team members and other resources</p> <p>System justification or business case</p>
Analysis	<p>Description of current system and where problems or opportunities are with a general recommendation on how to fix, enhance, or replace current system</p> <p>Explanation of alternative systems and justification for chosen alternative</p>
Design	<p>Functional, detailed specifications of all system elements (data, processes, inputs, and outputs)</p> <p>Technical, detailed specifications of all system elements (programs, files, network, system software, etc.)</p> <p>Acquisition plan for new technology</p>
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

قلب عملية تطوير الأنظمة

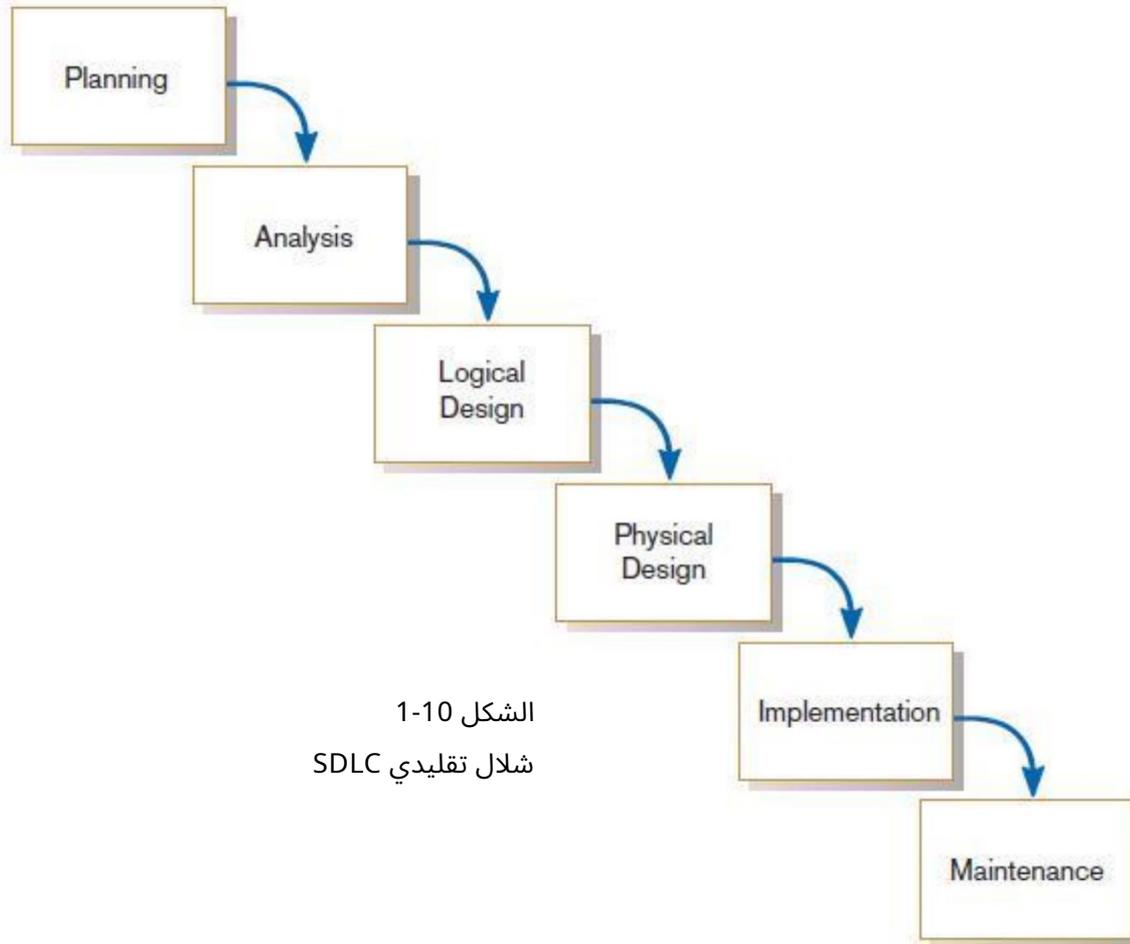
الشكل 1-8
الشكل 1-9

التحليل والتصميم - الكود - حلقة الاختبار



تجمع الممارسة الحالية بين التحليل والتصميم والتنفيذ في عملية واحدة تكرارية ومتوازية من الأنشطة.

الشلال التقليدي SDLC



الشكل 1-10
شلال تقليدي SDLC

تبدأ إحدى المراحل عندما
تتكمّل مرحلة أخرى ، مع
القليل من التراجع والتكرار.

مشاكل مع نهج الشلال

□ تجاهل الملاحظات ، وتثبت المعالم في مواصفات التصميم حتى عندما تتغير الظروف

□ مشاركة محدودة للمستخدم (فقط في مرحلة المتطلبات)

□ التركيز بشكل كبير على المواعيد النهائية الهامة

مراحل SDLC على حساب ممارسات التنمية
السليمة

مناهج مختلفة للتحسين تطوير

□ أدوات CASE

□ التطوير السريع للتطبيقات
(راد)

□ منهجيات رشيقة

□ برمجة إكستريم

البرمجيات بمساعدة الحاسوب أدوات الهندسة (CASE)

□ تتيح أدوات الرسم التخطيطي التمثيل الرسومي.

□ تساعد شاشات الكمبيوتر ومولدات التقارير في وضع نموذج أولي لكيفية
"الشكل والمظهر" للأنظمة.

تعتبر منتجات IBM's Rational هي الأفضل
أدوات CASE المعروفة.

أدوات هندسة البرمجيات بمساعدة الحاسوب (CASE) (تابع)

□ تتحقق أدوات التحليل تلقائيًا من الاتساق في الرسوم البيانية والنماذج والتقارير.

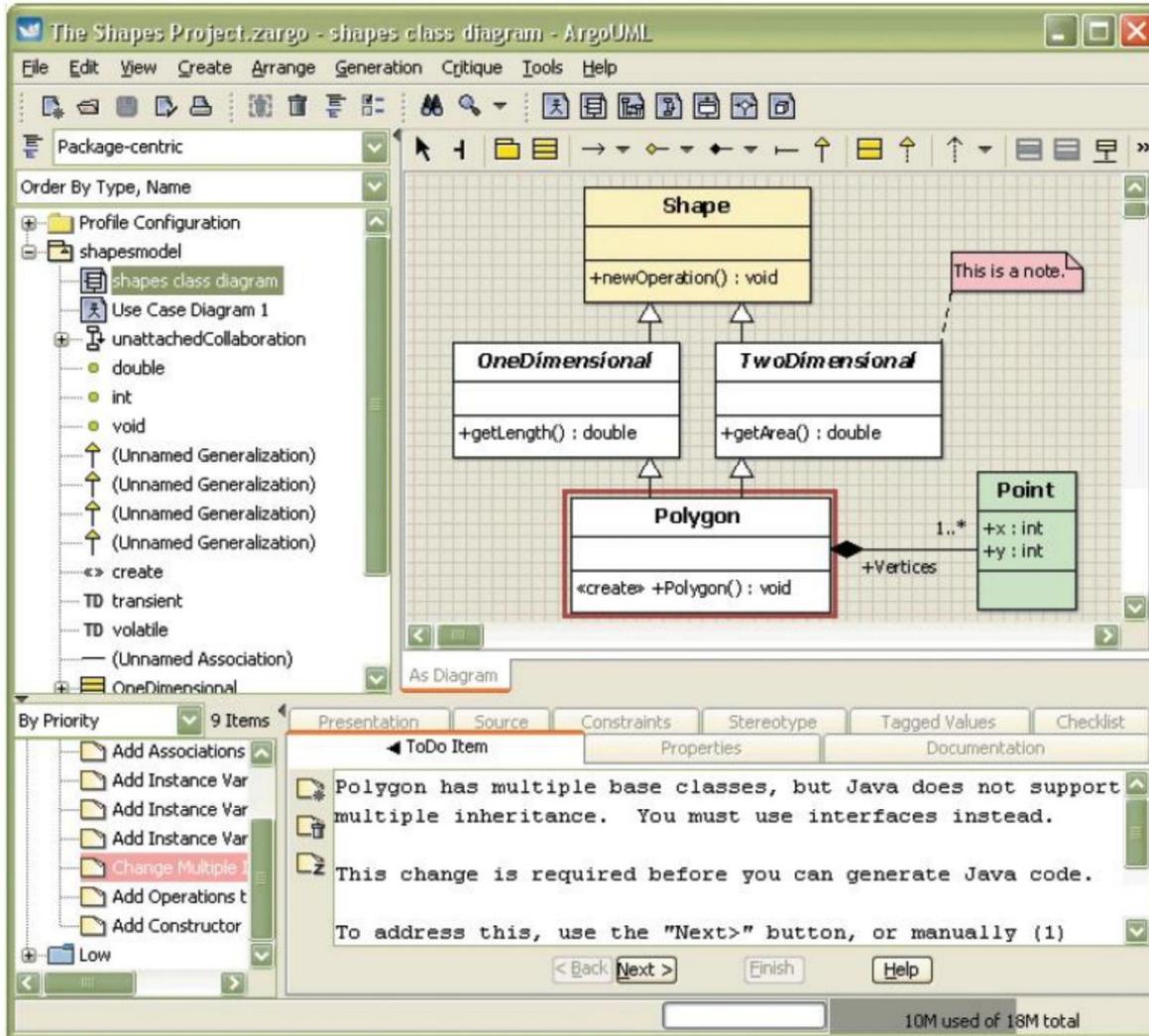
□ يوفر المستودع المركزي تخزينًا متكاملًا للرسومات التخطيطية والتقارير ومواصفات إدارة المشروع.

أدوات هندسة البرمجيات بمساعدة الحاسوب (CASE) (تابع)

□ تعمل أدوات التوثيق على توحيد التوثيق الفني وتوثيق المستخدم.

□ مولدات الكود تمكّن تلقائيًا إنشاء البرامج وكود قاعدة البيانات مباشرة من وثائق التصميم والرسوم البيانية والنماذج والتقارير.

أدوات CASE (تابع)



الشكل 1-11
لقطة شاشة من
أداة ArgouML مفتوحة ،
المصدر CASE

(المصدر: <http://argouml.tigris.org/>)

أدوات CASE (تابع)

TABLE 1-2 Examples of CASE Usage within the SDLC

SDLC Phase	Key Activities	CASE Tool Usage
Project identification and selection	Display and structure high-level organizational information	Diagramming and matrix tools to create and structure information
Project initiation and planning	Develop project scope and feasibility	Repository and documentation generators to develop project plans
Analysis	Determine and structure system requirements	Diagramming to create process, logic, and data models
Logical and physical design	Create new system designs	Form and report generators to prototype designs; analysis and documentation generators to define specifications
Implementation	Translate designs into an information system	Code generators and analysis, form and report generators to develop system; documentation generators to develop system and user documentation
Maintenance	Evolve information system	All tools are used (repeat life cycle)

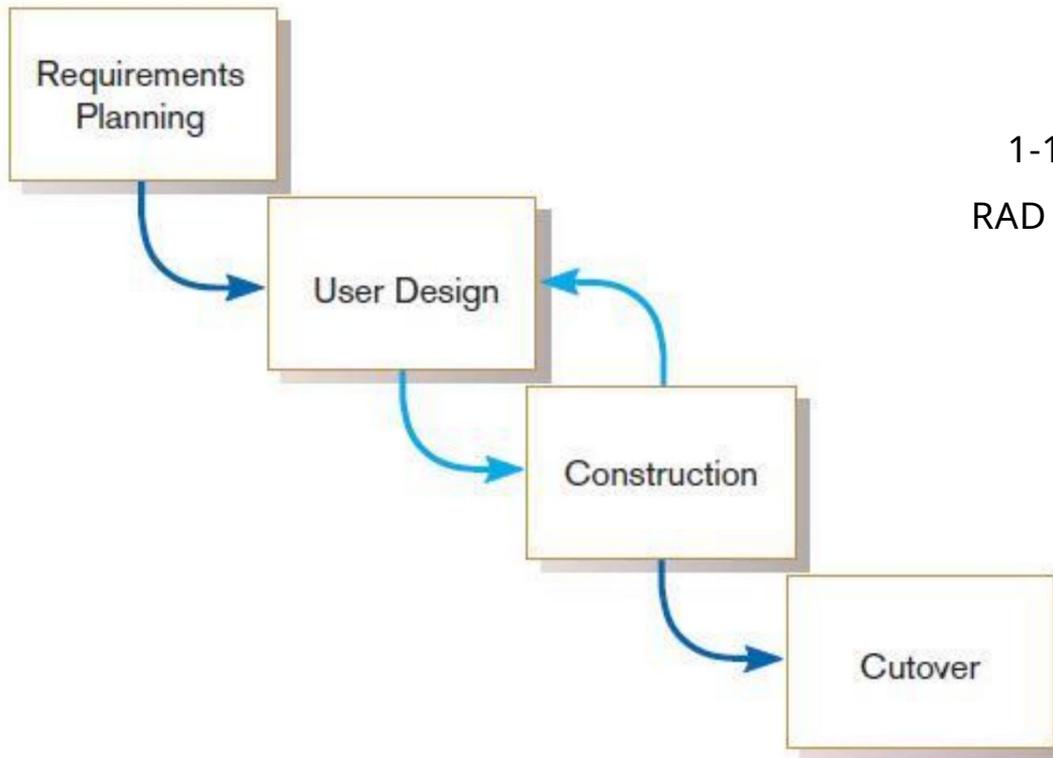
التطوير السريع للتطبيق (راد)

يقلل من التصميم والتنفيذ
زمن

يشمل : مشاركة واسعة من المستخدمين ،
النماذج الأولية وأدوات CASE المتكاملة ومولدات الأكواد

□ مزيد من التركيز على واجهة المستخدم ووظيفة النظام ، وبدرجة أقل
على تحليل الأعمال التفصيلي وأداء النظام

تطوير التطبيقات السريعة (RAD) (تابع)



الشكل 1-12
دورة حياة RAD

منهجيات رشيقة

بدافع الاعتراف بالبرمجيات
التنمية على أنها سلسلة وغير متوقعة وديناميكية

- ثلاثة مبادئ رئيسية
- تكيفية وليست تنبؤية
- ركز على الأشخاص بدلاً من الأدوار
- عمليات التكيف الذاتي

TABLE 1-3 The Agile Manifesto

The Manifesto for Agile Software Development

Seventeen anarchists agree:

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

- *Individuals and interactions* over processes and tools.
- *Working software* over comprehensive documentation.
- *Customer collaboration* over contract negotiation.
- *Responding to change* over following a plan.

تجادل مجموعة منهجيات
Agile بأن منهجيات تطوير
البرمجيات

مقتبس من الهندسة بشكل
عام لا يتناسب مع تطوير
برمجيات العالم الحقيقي .

That is, while we value the items on the right, we value the items on the left more.

We follow the following principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Businesspeople and developers work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Continuous attention to technical excellence and good design enhances agility.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

—Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas (www.agileAlliance.org)

(Source: <http://agilemanifesto.org/> © 2001, the above authors this declaration may be freely copied in any form, but only in its entirety through this notice.)

متى تستخدم منهجيات Agile

- إذا كان مشروعك يتضمن:
- متطلبات ديناميكية أو غير متوقعة
- المطورون المسؤولون والمتحمسون
- العملاء الذين يفهمون العملية و

سوف تتدخل

TABLE 1-4 Five Critical Factors That Distinguish Agile and Traditional Approaches to Systems Development

Factor	Agile Methods	Traditional Methods
Size	Well matched to small products and teams. Reliance on tacit knowledge limits scalability.	Methods evolved to handle large products and teams. Hard to tailor down to small projects.
Criticality	Untested on safety-critical products. Potential difficulties with simple design and lack of documentation.	Methods evolved to handle highly critical products. Hard to tailor down to products that are not critical.
Dynamism	Simple design and continuous refactoring are excellent for highly dynamic environments but a source of potentially expensive rework for highly stable environments.	Detailed plans and Big Design Up Front, excellent for highly stable environment but a source of expensive rework for highly dynamic environments.
Personnel	Requires continuous presence of a critical mass of scarce experts. Risky to use no-agile people.	Needs a critical mass of scarce experts during project definition but can work with fewer later in the project, unless the environment is highly dynamic.
Culture	Thrives in a culture where people feel comfortable and empowered by having many degrees of freedom (thriving on chaos).	Thrives in a culture where people feel comfortable and empowered by having their roles defined by clear practices and procedures (thriving on order).

(Source: Boehm, Barry; Turner, Richard, *Balancing Agility and Discipline: A Guide for the Perplexed*, 1st Ed., (c)2004. Reprinted and electronically reproduced by permission of Pearson Education, Inc. Upper Saddle River, New Jersey.

البرمجة المتطرفة

- دورات تطوير تزايدى قصيرة
- الاختبارات الآلية
- فرق برمجة مكونة من شخصين
- البرمجة والاختبار والاستماع والتصميم

برمجة eXtreme (تابع)

□ يعمل التشفير والاختبار معًا

□ المزايا:

□ التواصل بين المطورين

□ ارتفاع مستوى الإنتاجية

□ كود عالي الجودة

التحليل الكينوني و التصميم (OOAD)

على أساس الأشياء وليس البيانات أو العمليات

□ الكائن: هيكل يحتوي على سمات وسلوكيات كيان
في العالم الحقيقي

التحليل والتصميم الكينوني (OOAD)(تابع)

□ فئة الكائن: مجموعة منطقية للكائنات تشترك في نفس السمات والسلوكيات

الميراث : هرمي

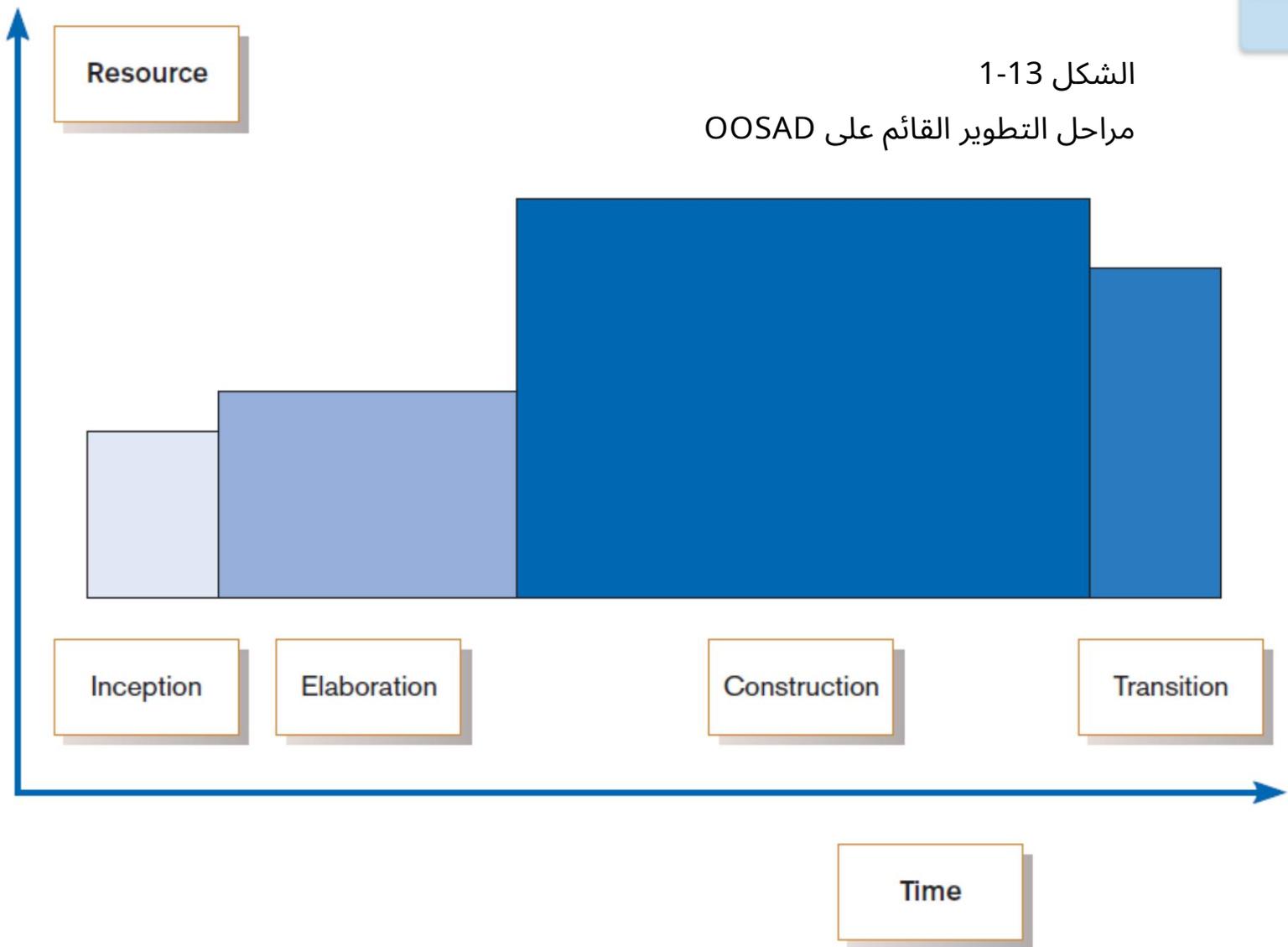
يمكن ترتيب الفئات الفرعية من وراثته خصائص الطبقات الفائقة

العملية العقلانية الموحدة (RUP)

□ منهجية تطوير الأنظمة الشيئية

يؤسس أربع مراحل من التطوير: البداية ، والتوضيح ، والبناء ، والانتقال

يتم تنظيم كل مرحلة في عدد من التكرارات المنفصلة.



ملخص

في هذا الفصل تعلمت كيفية:

تحديد تحليل وتصميم نظم المعلومات.

□ وصف دورة حياة تطوير نظم المعلومات (SDLC).

□ شرح التطوير السريع للتطبيق ، (RAD)

النماذج الأولية ، وهندسة البرمجيات بمساعدة الحاسوب ، (CASE) والهندسة المعمارية الموجهة للخدمة (SOA).

□ وصف منهجيات رشيقة وبرمجة إكستريم.

□ شرح التحليل والتصميم الكينوني

(OOAD) والعملية العقلانية الموحدة (RUP).