

Rapid Application Development

Chapter 1 : Introduction to Rapid Application Development

Dr. Orawit Thinnukool

College of Arts, Media and Technology, Chiang Mai University

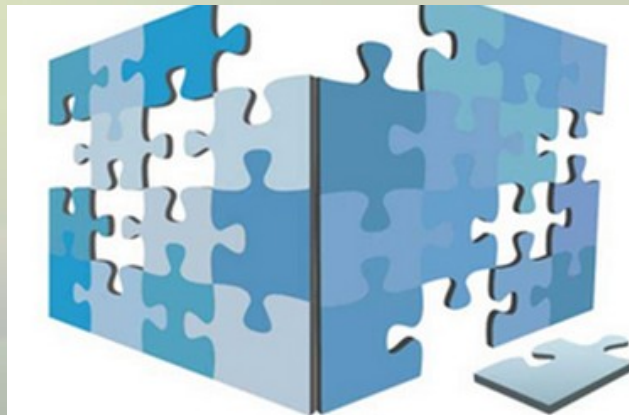
Contents

- The history of Rapid Application Development
- Essential Ingredients of RAD
- Introduction to Rapid Application Development
- The James Martin RAD Methodology
- Pros and Cons of Rapid Application Development
- Steps to develop RAD
- Points to remember before selecting the best software methodology
- How to analysis Methodology to develop RAD
- Tools for Development and Design RAD

Introduction to Rapid Application Development

What is Rapid Application Development ?

R A D is a development lifecycle designed to give much faster development and higher-quality results than those achieved with the traditional lifecycle. It is designed to take the maximum advantage of powerful development software that has evolved recently.



The history of Rapid Application Development

- Rapid application development is a response to processes developed in the 1970s and 1980s, such as the Structured Systems Analysis and Design Method and other Waterfall models.
- The waterfall solution to this was to try and rigidly define the requirements and the plan to implement them and have a process that discouraged changes to either.
- RAD approaches on the other hand recognized that software development was a knowledge intensive process and sought to develop flexible processes that could take advantage of knowledge gained over the life of the project and use that knowledge to reinvent the solution.

The history of Rapid Application Development (Cont.)

- The first such RAD alternative was developed by Barry Boehm and was known as the spiral model.
- Boehm and other subsequent RAD approaches emphasized developing prototypes as well as or instead of rigorous design specifications.
- Risk reduction
- Users are better at using and reacting than at creating specifications.
- Prototypes can be usable and can evolve into the completed product.

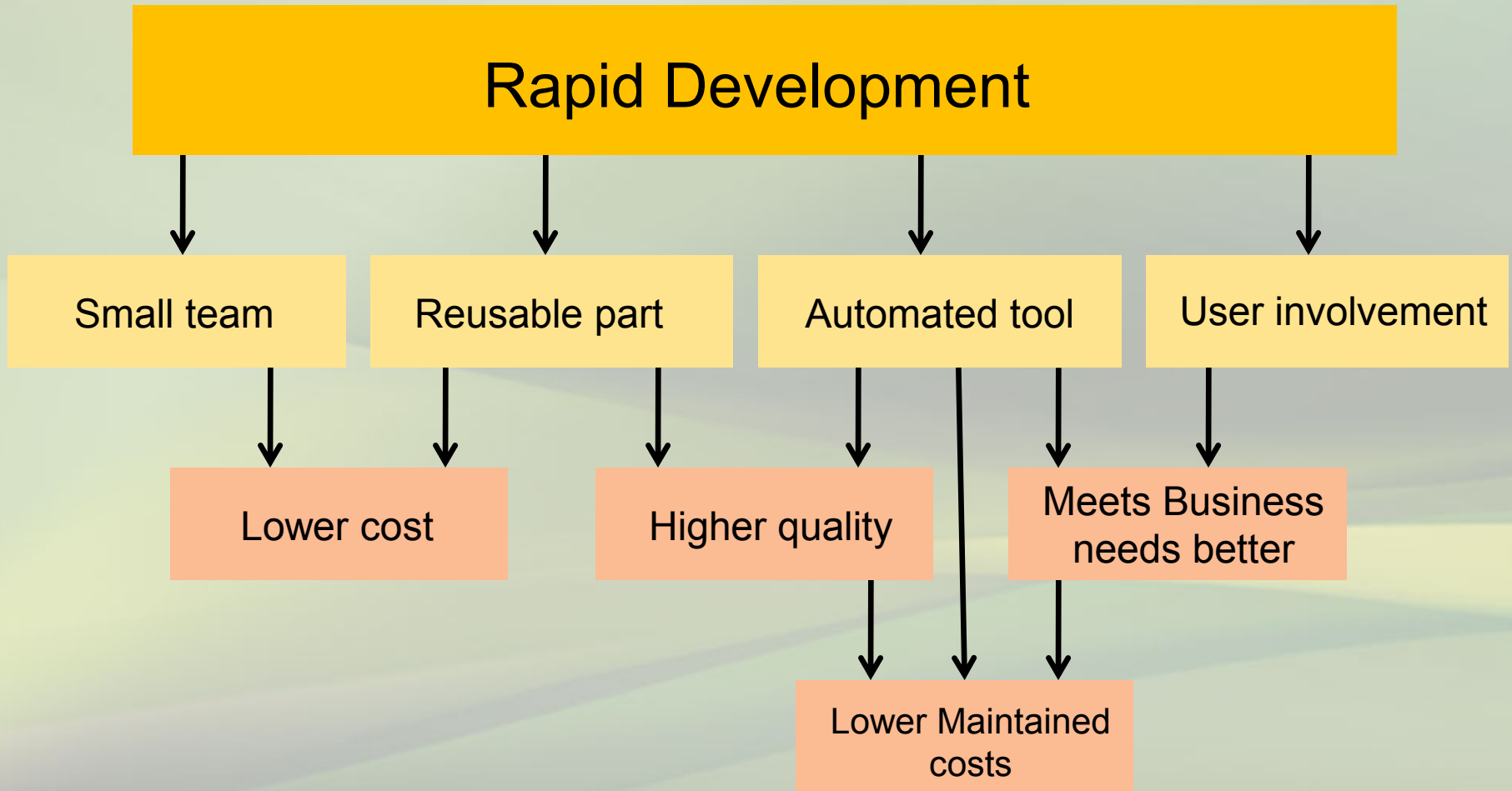


Dr. Barry Boehm

The history of Rapid Application Development (Cont.)

- Starting with the ideas of Barry Boehm and others, James Martin developed the rapid application development approach during the 1980s at IBM.
- This has resulted in some confusion over the term RAD even among IT professionals.
- RAD was often an essential part of larger business re engineering programs. The rapid prototyping approach of RAD was a key tool to help users and analysts "think out of the box" about innovative ways that technology might radically reinvent a core business process

Introduction to Rapid Application Development

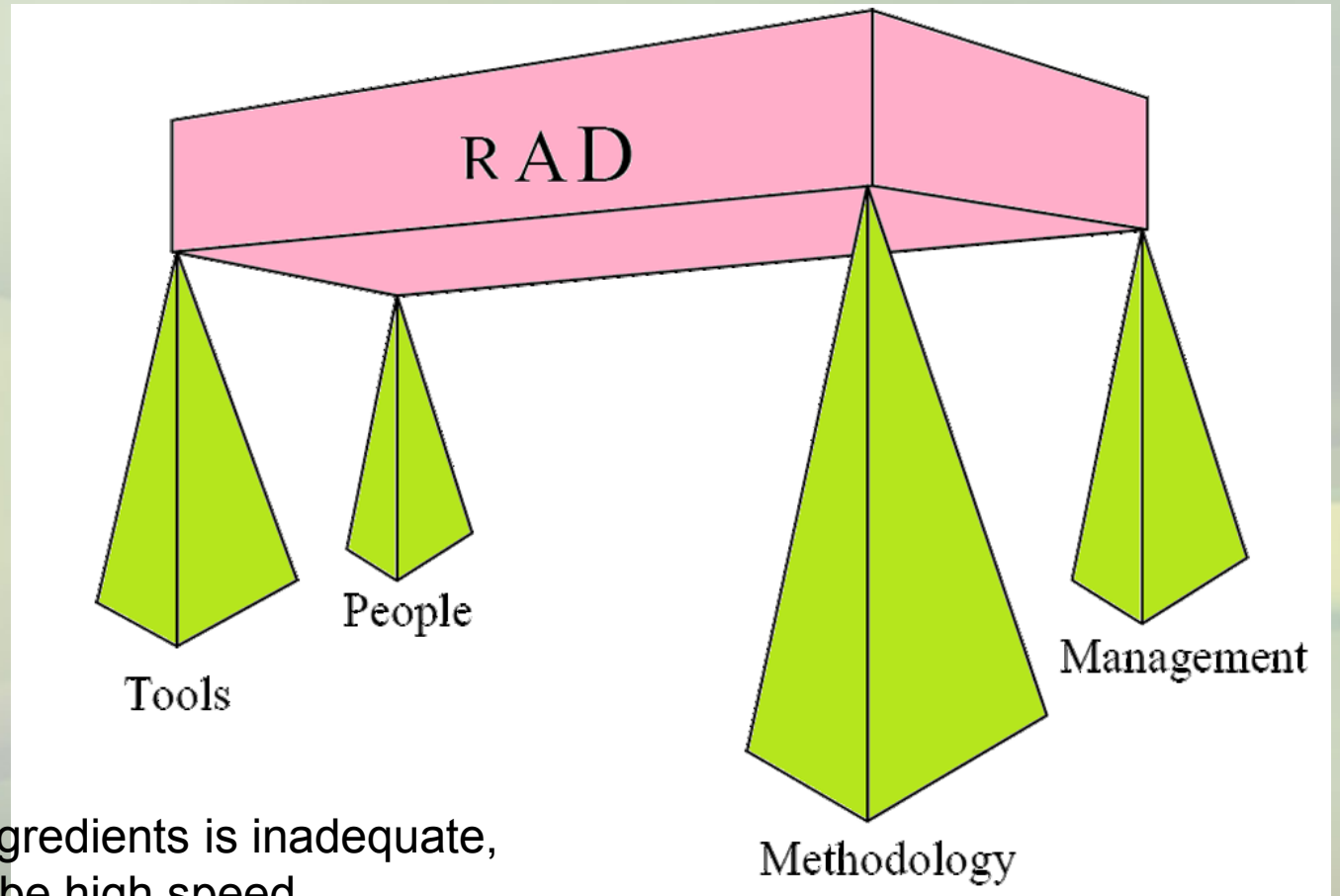


**Rapid development, high quality and lower costs go hand in hand if an appropriate development methodology is used

Essential Ingredients of RAD

RAD has four essential ingredients:

- 1.Tools,
- 2.Methodology,
- 3.People, and
4. Management.



If any one of these ingredients is inadequate, development will not be high speed

ESSENTIAL INGREDIENTS OF RAD

1. Tools

The use of powerful CASE tools will provide automation support for systems development through features such as code generation and automatic consistency checking. CASE tools that generate prototypes can be used to support an iterative development approach such as RAD, allowing end users to see the application evolves as it is being built.

2. Methodology

The most effective family of techniques must be formalized and used to deliver the systems. A complete list of tasks is provided to ensure that no essential activity is overlooked, while techniques are fully documented to ensure that a task is performed in the proper way.

ESSENTIAL INGREDIENTS OF RAD

3. People

Bright people must be well-trained in both the methodology and the tools. Small teams that work consistently well together should be grouped together on assignments.

4. Management

The project must be managed for speed through the use of technique such as facilitated Joint Requirements Planning (JRP) and Joint Application Design (JAD) workshops to extract users' requirements quickly. Timebox Management is used in Rapid Construction to iteratively deliver the system to the users.

The James Martin RAD Methodology

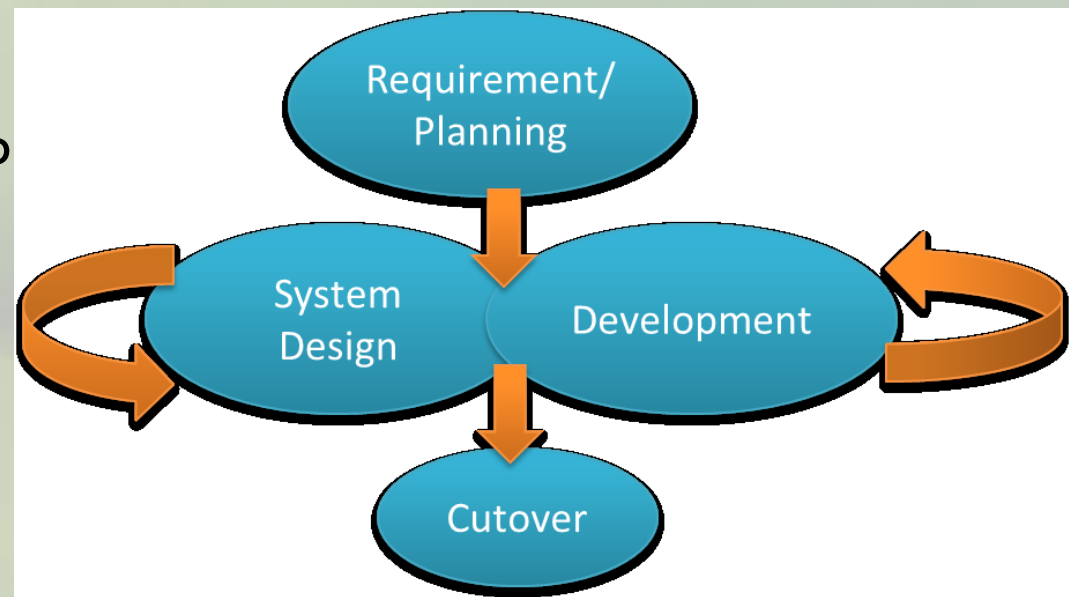
- The James Martin approach to RAD divides the process into four distinct phases.

1. Requirements Planning p

2. System design phase

3. Development phase

4. Cutover phase



Phases in the James Martin approach to RAD

Pros and Cons of Rapid Application Development

The advantages

- Better Quality
- Risk Control
- More projects completed on time and within budget
- Flexible

The disadvantages

- The risk of a new approach
- Requires time of scarce resources
- Less control
- Poor design
- Very large systems

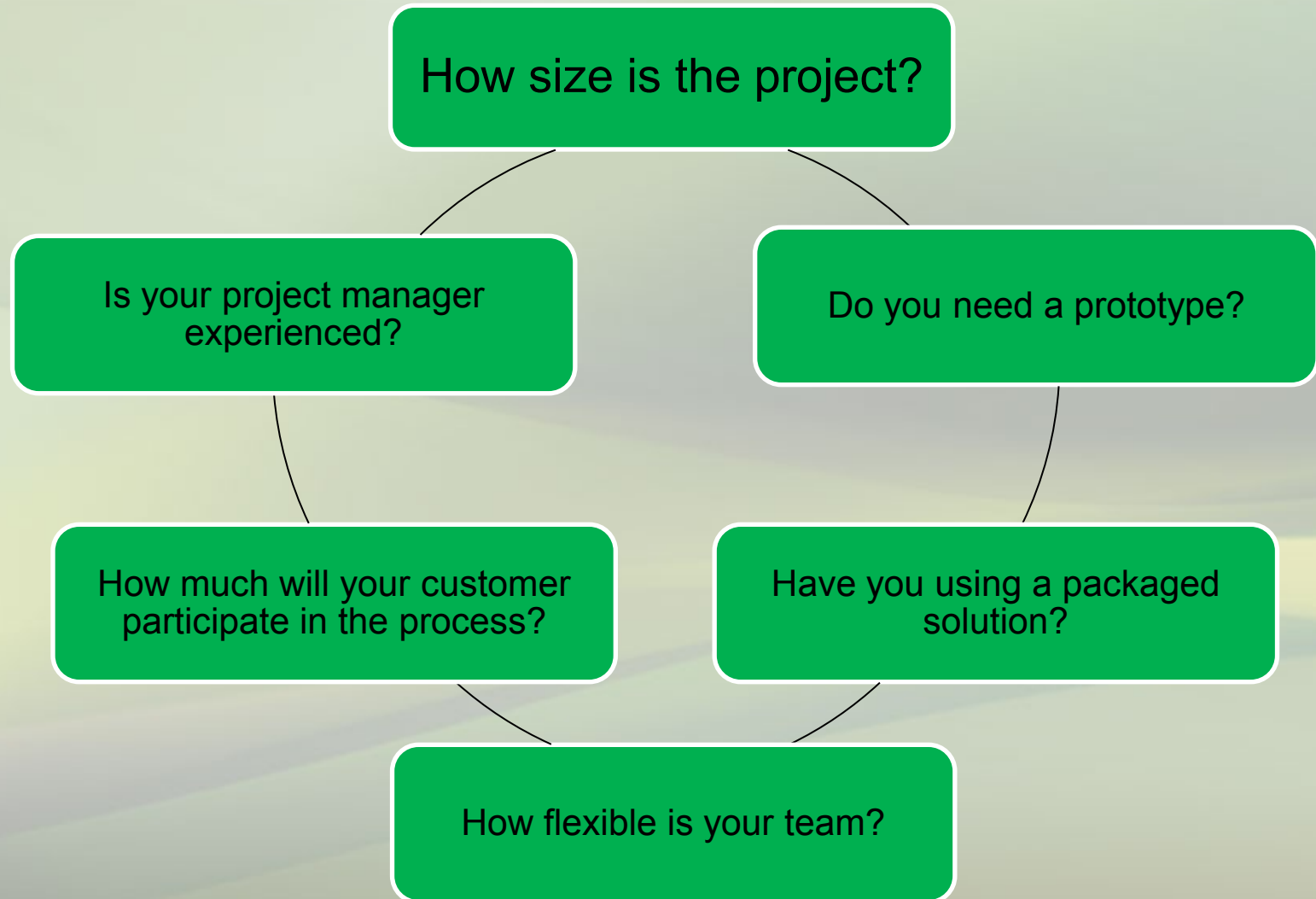
Steps to develop RAD

- Selection of target application
- Considered several criteria as we evaluate potential of RAD
 - Internal or external process
 - Large enterprise
 - Value to organization
- Survey of potential users
 - What management issues are not adequately addressed by existing tools or other resource ?
 - How do they solve this problem now?
 - What do they envision as a desirable solution?
 - Who are the potential users of the new application?
 - Is there a need to exchange data with other application?

Steps to develop RAD (cont.)

- Specific requirements
 - we sought to develop an automated management tool that can be used to project, reserve, and track resources related to task;
 - People : tracking professional development skill
 - : projecting staffing need for adding personal
 - Tasks : facilitating task planning by looking the project plan to people
 - : tracking task process, comparing actual values to projections
 - : conducting trend analysis of skill required, personal, and funding
 - : conducting risk analysis as an integral part of task planning

Points to remember before selecting the best software methodology



How to analysis Methodology to develop RAD

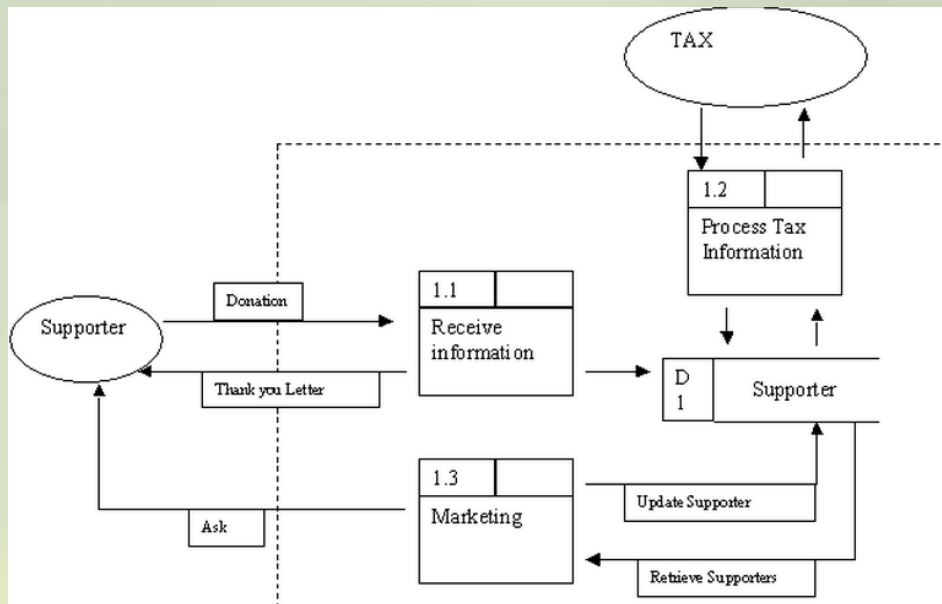
The object-oriented analysis methodology select for RAD was use case requirements Model ;

- High-level use cases are develop to identify functional area of the system
- A use case diagram is used to portray the system
- Expanded use case are developed to show detail and order of events
- Sequence diagram are used to display process flow
- Class diagram is developed to show identified objects and their relationship

How to analysis Methodology to develop RAD

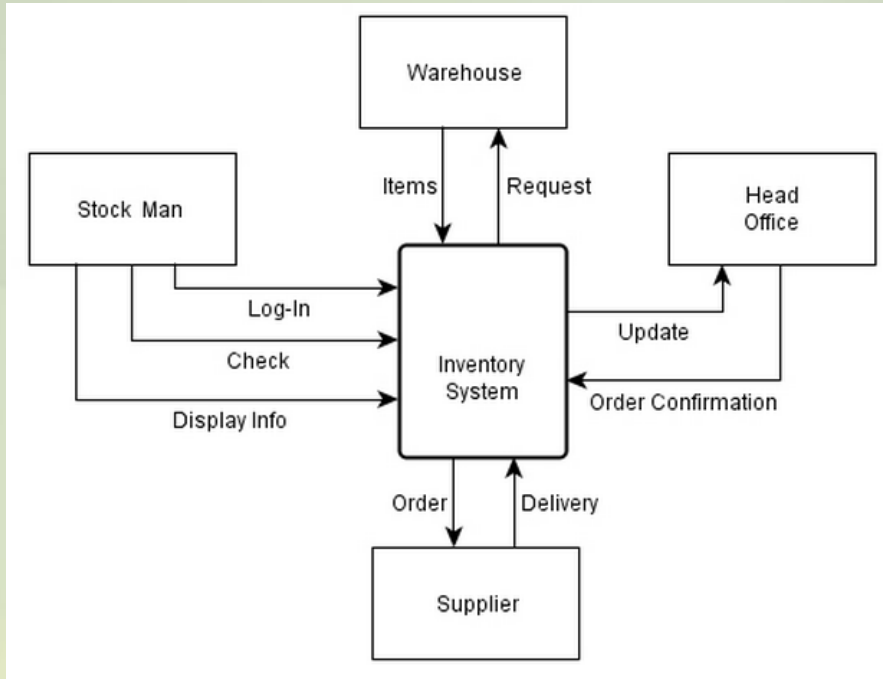
- **Context diagram**

- is a graphical representation of the "flow" of data through an information system.
- **A DFD** shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).



Data flow diagram example for the tax payment system

How to analysis Methodology to develop RAD

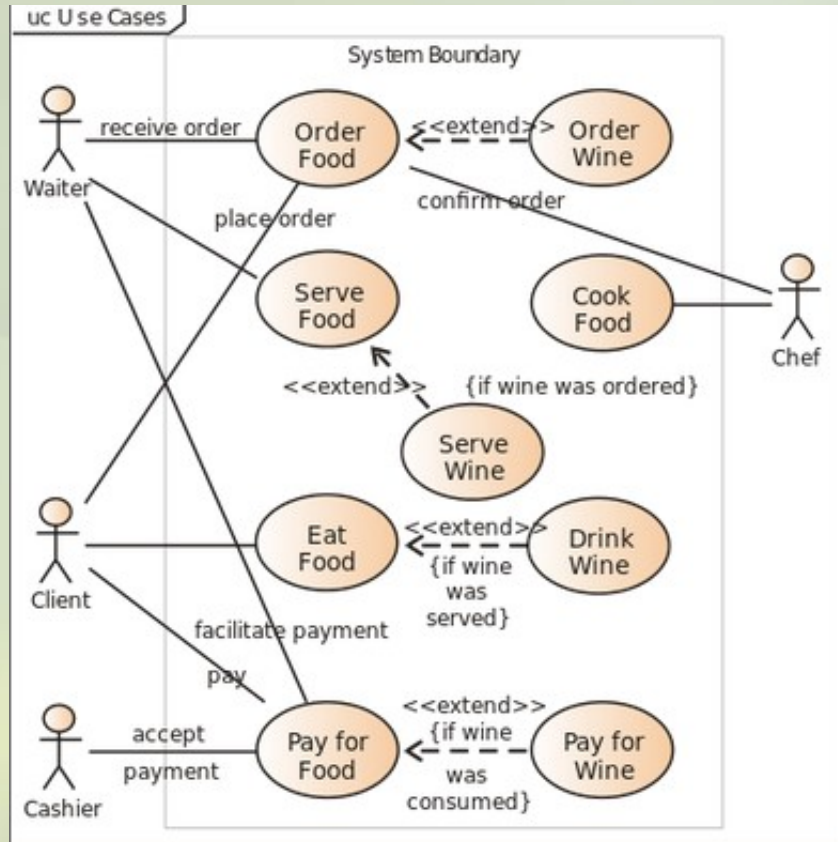


Context diagram example for designing the inventory system

- **Context diagram**

- context diagrams show a system, often software-based, as a whole and its inputs and outputs from/to external factors
- it defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it
- context diagrams are typically included in a requirements document.

How to analysis Methodology to develop RAD

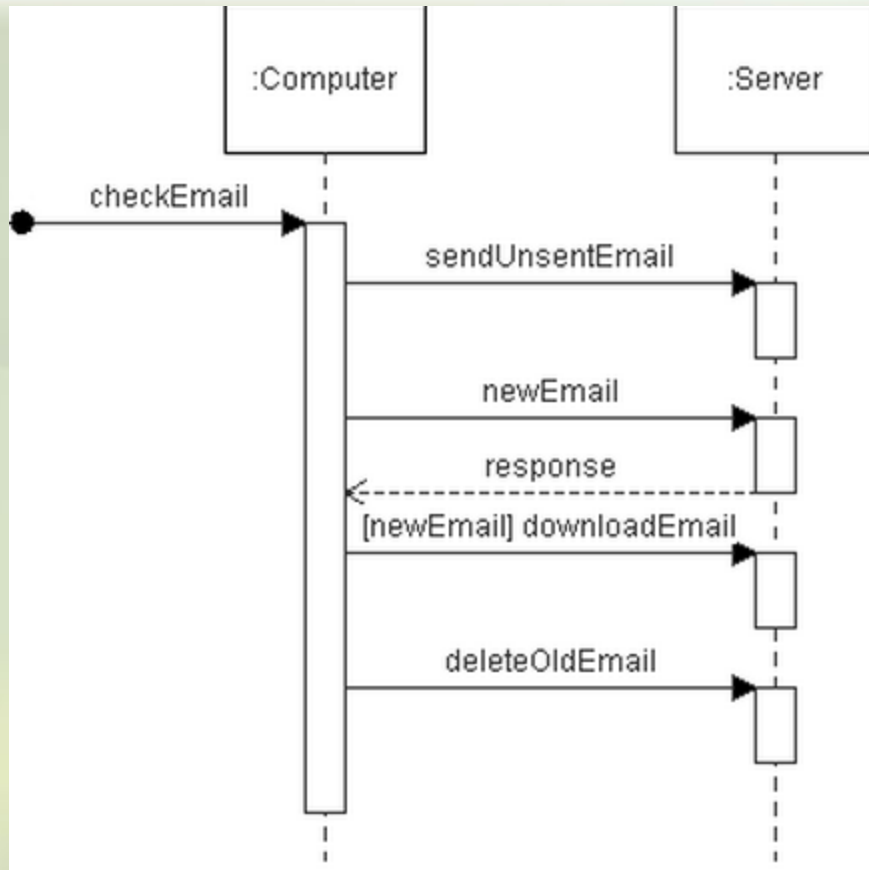


- **Use case diagram**

Use case is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human, an external system, or time.

A UML Use Case Diagram for the interaction of a client (the actor) and a restaurant (the system)

How to analysis Methodology to develop RAD



Sequence diagram
for sending an e-mail, the process
between computer and server

- **Sequence diagram**

- is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.
- A sequence diagram shows object interactions arranged in time sequence.
- It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

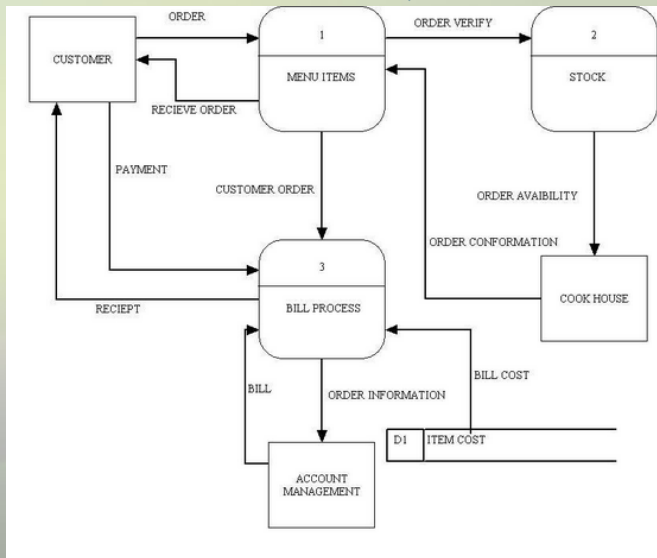
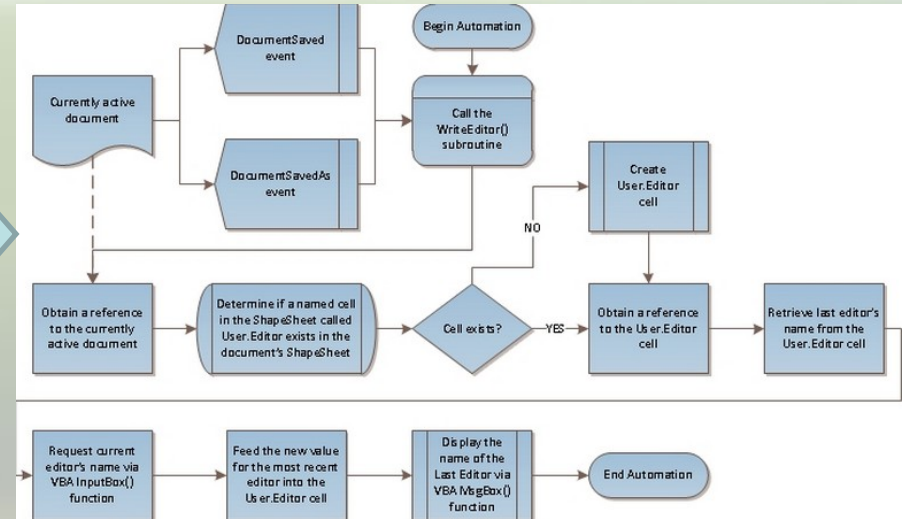
CASE- tools

- Computer-aided software engineering (CASE) is the use of software tools to assist in the development and maintenance of software. Tools used to assist in this way are known as CASE Tools

For example:

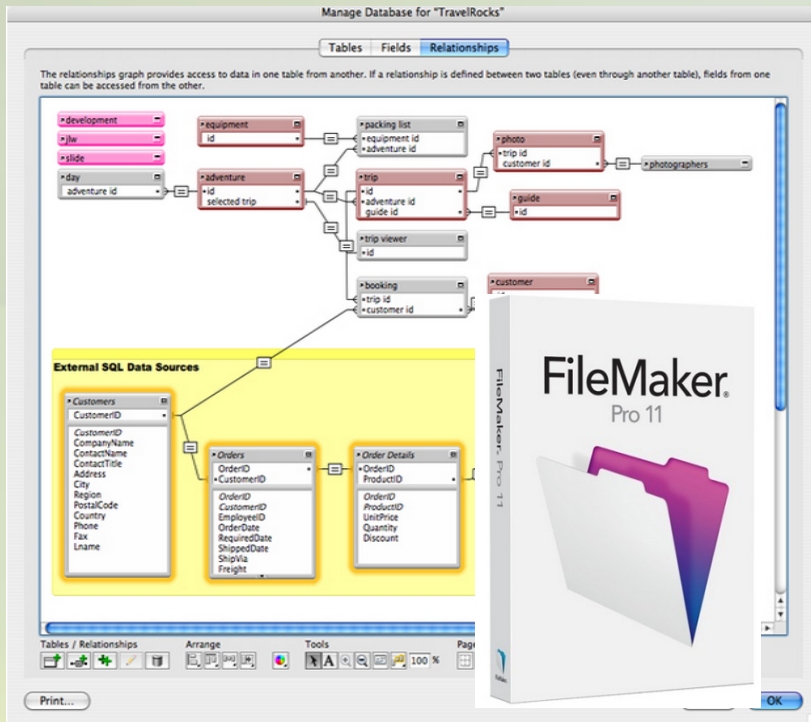
1. Software Testing Tools
 2. Software Maintenance Tools
 3. Software Engineering Management Tools
1. Software Testing Tools
 2. Software Maintenance Tools
 3. Software Engineering Management Tools

Tools for Development and Design RAD



Visio is a software, which provides many features that make your diagrams more meaningful, flexible, and responsive to your needs as well as tools for design program such as DFD diagram, Flow-chart , grant-chart Use case-diagram, class diagram.

Tools for Development and Design RAD



FileMaker software is normally positioned as an alternative to other desktop databases, but the product's "instant web publishing" feature makes it a good product for servicing alternate, web-enabled devices that need access to databases or spreadsheets.

pros:

- Develop applications with no coding needed fastest way to get a database on the web simple to learn and use

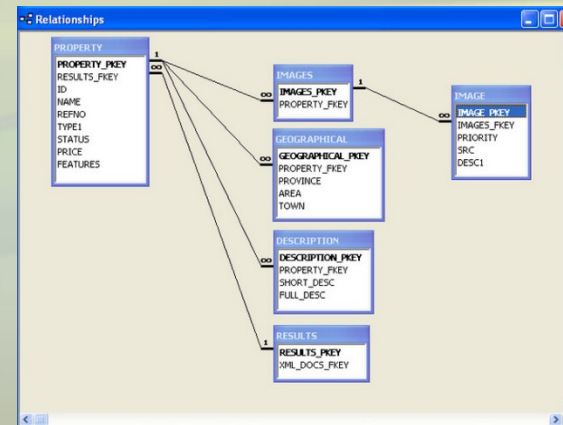
cons:

- Need to buy server versions to support additional users
- Limited Web application scripts

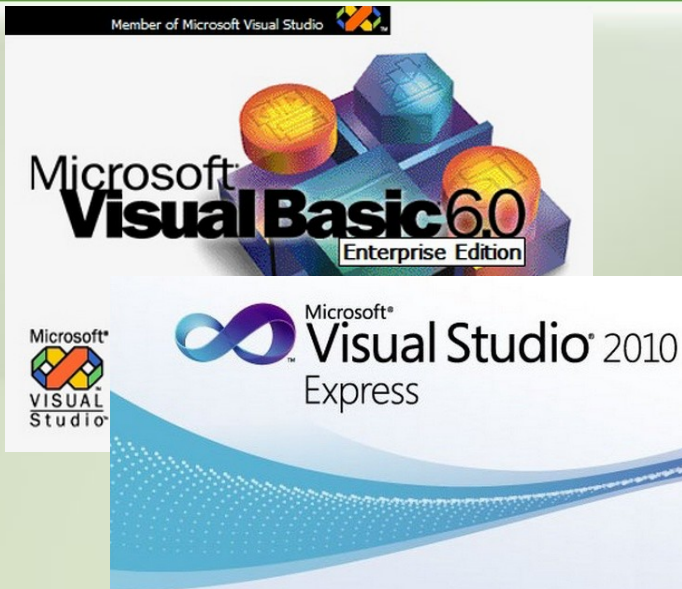
Tools for Development and Design RAD



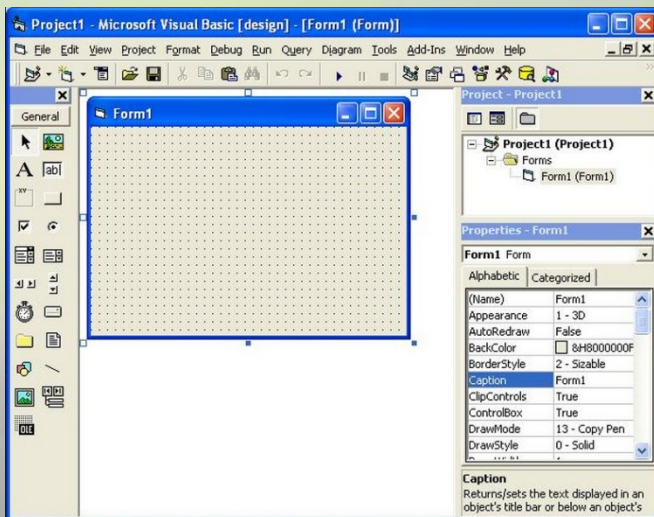
Microsoft Access also may be used as the 'front-end' of a program. Access tables support a variety of standard field types, indices, and referential integrity including cascading updates and deletes. Access also includes a query interface, forms to display and enter data, and reports for printing.

A screenshot of a Microsoft Access form titled "Property Details for: 121 Any Road, Anytown". The form is divided into several sections. On the left, there are input fields for "Property ID" (value: 2), "Property Type" (dropdown: HOUSE - TERRACED), "Property Number" (value: 121), "Address" (value: Any Road), "Town" (value: Anytown), "Postal Code" (value: AB10 1CD), "County" (empty), "Free Hold" (checked), "Lease Hold" (unchecked), and "Asking Price" (value: £200,000.00). On the right, there is a "Photo" field with a file path "C:\Documents and Settings\G..." and "Add Image" and "Delete Image" buttons. Below that is a "Notes" field containing two paragraphs of text: "Situated on one of the most s... locations in town, this conserv... offers easy access to town an... local amenities." and "This house, built circa 1885, c... many of the original features... been kept to a very high stan... decor." At the bottom, there are navigation buttons and a page indicator "Property 1 of 2".

Tools for Development and Design RAD

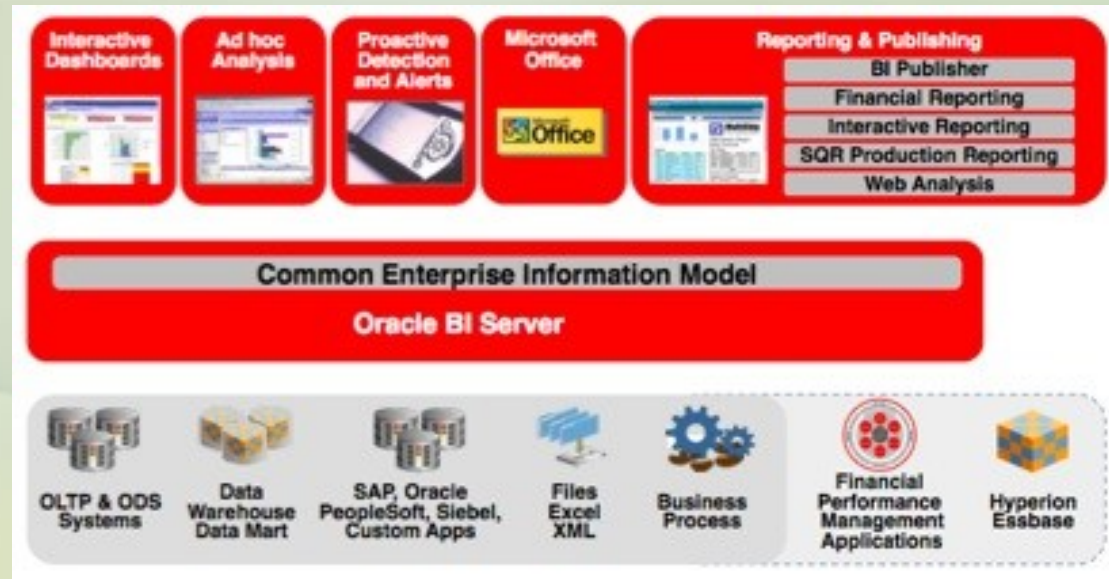


Visual Basic software was derived from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using Data Access Objects, Remote Data Objects, or ActiveX Data Objects, and creation of ActiveX controls and objects.



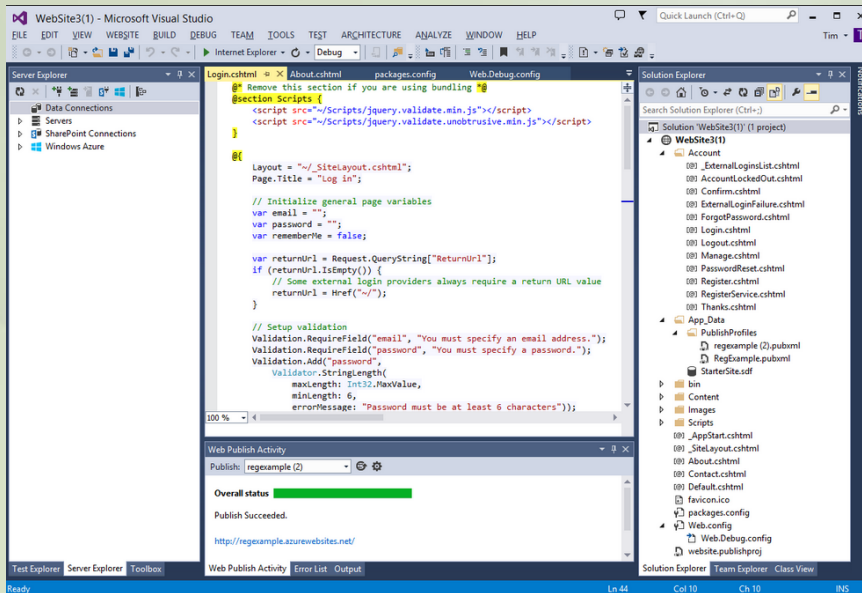
Forms are created using drag-and-drop techniques. A tool is used to place controls (e.g., text boxes, buttons, etc.) on the form (window). Controls have attributes and event handlers associated with them.

Tools for Development and Design RAD

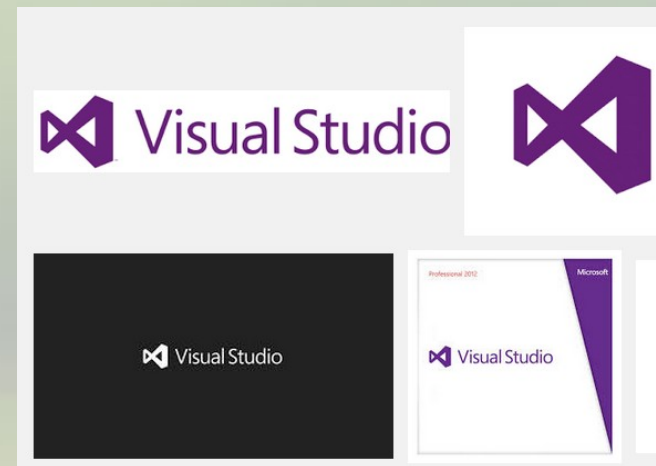


- Oracle Enterprise Development Suite
 - Oracle Developer Suite is the most complete and integrated development environment, combining the power of application development and business intelligence tools in a single suite that is based on the latest industry standard

Tools for Development and Design RAD



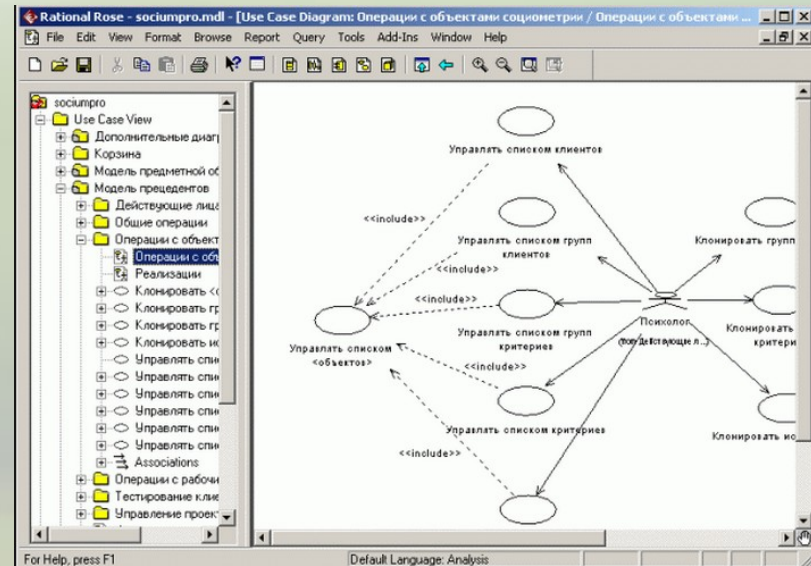
Microsoft Visual Studio



- It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Tools for Development and Design RAD

Rational Rose XDE

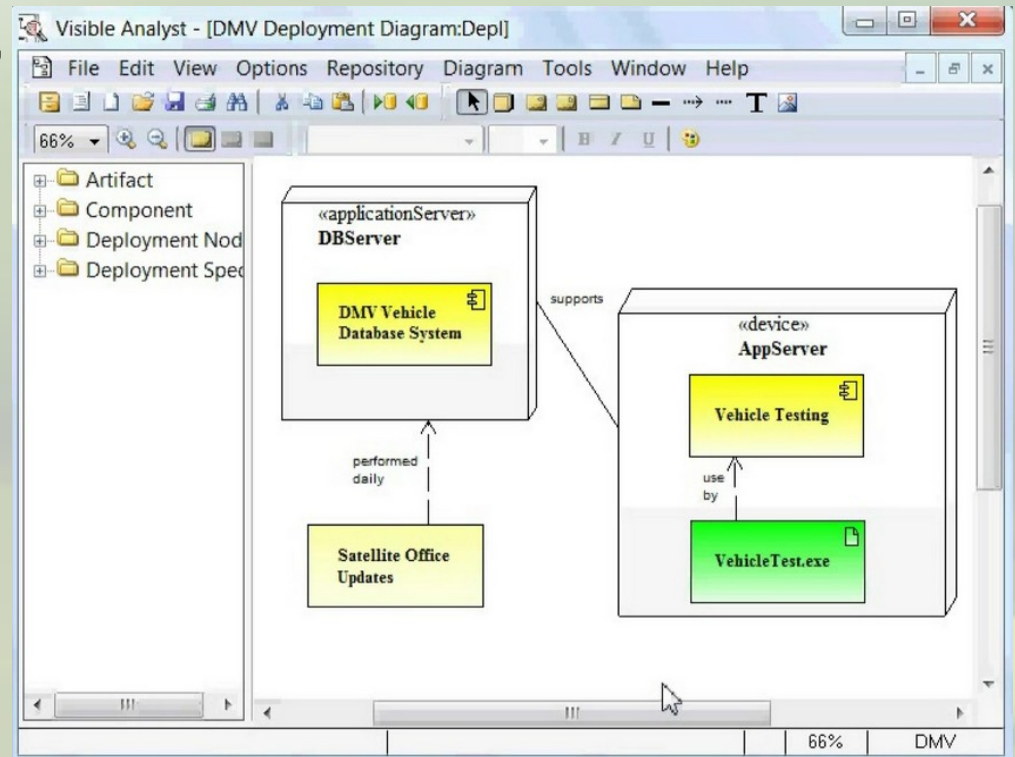


The Rational Rose program is a set of UML modeling tools for software design. The Unified Modeling Language (UML) is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It simplifies the complex process of software design, creating a "blueprint" for construction of software systems.

Rational Rose could also do source-based reverse engineering; the combination of this capability with source generation from diagrams was dubbed roundtrip engineering

Tools for Development and Design RAD

This program is designed for use by consultants, business analysts, data base administrators, enterprise architects, data warehouse architects and others performing structured analysis and design, BPMN, Data Warehouse or Enterprise Architecture activities. The Visible Analyst supports Structured Systems Analysis, Structured Systems Design and Object Oriented Development and BPMN. Our new custom modeling capability allows users to design and create custom diagram types and custom symbols with full repository support.



Visible®

Answer The questions

- 1: What is Rapid Application Development (Definition) ?
- 2: What is used for Rapid Application Development ?
- 3: What tools used to design DFD, and context diagram?
- 4: What is point to develop best software methodology?

References

- Boehm, Barry (May 1988). "A Spiral Model of Software Development". *IEEE Computer*. Retrieved 1 July 2014.
- Drucker, Peter (November 3, 2009). *Post-Capitalist Society*. Harper Collins e-books. ISBN 0887306209.
- Martin, James (1991). *Rapid Application Development*. Macmillan. ISBN 0-02-376775-8.
- Martin, James (1991). *Rapid Application Development*. Macmillan. pp. 81–90. ISBN 0-02-376775-8.
- Hotle, Matt (April 13–14, 2010). "The Disintegration of AD: Putting it Back Together Again". <http://www.gartner.com.br>. Enterprise Integration Summit, Sao Palo, Brazil: Gartner Group. Retrieved 1 July 2014.
- Beck, Kent (2000). *Extreme Programming Explained*. Addison Wesley. pp. 37. ISBN 0201616416.
- Gerber, Aurlon; Van der Merwe, Alta; Alberts, Ronell; (2007), Implications of Rapid Development Methodologies, CSITEd 2007, Mauritius, November 2007
- Andrew Begel, Nachiappan Nagappan. "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study, *Microsoft Research*". Retrieved 2008-11-15.
- E. M. Maximilian and L. Williams. (2003). "Assessing Test-driven Development at IBM". Proceedings of International Conference of Software Engineering, Portland, OR, pp. 564-569, 2003.
- M. Stephens, Rosenberg, D. (2003). "Extreme Programming Re factored: The Case Against XP". Apress, 2003.