



**HY-351:**

**Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων**  
Information Systems Analysis and Design

Πανεπιστήμιο Κρήτης, Φθινόπωρο 2005

**Φροντιστήριο 1**

**Θέμα : CASE Tools**

**Ημερομηνία : 10 Οκτωβρίου 2005**



# Outline

- Software Development
  - Problem
  - Need for tools
- CASE Tools
  - Definition
  - Classification
  - Standard functionality
  - Components
  - Benefits and Difficulties
  - CASE Tools & UML
- Tool demonstration
  - Rational Modeler
  - ArgoUML

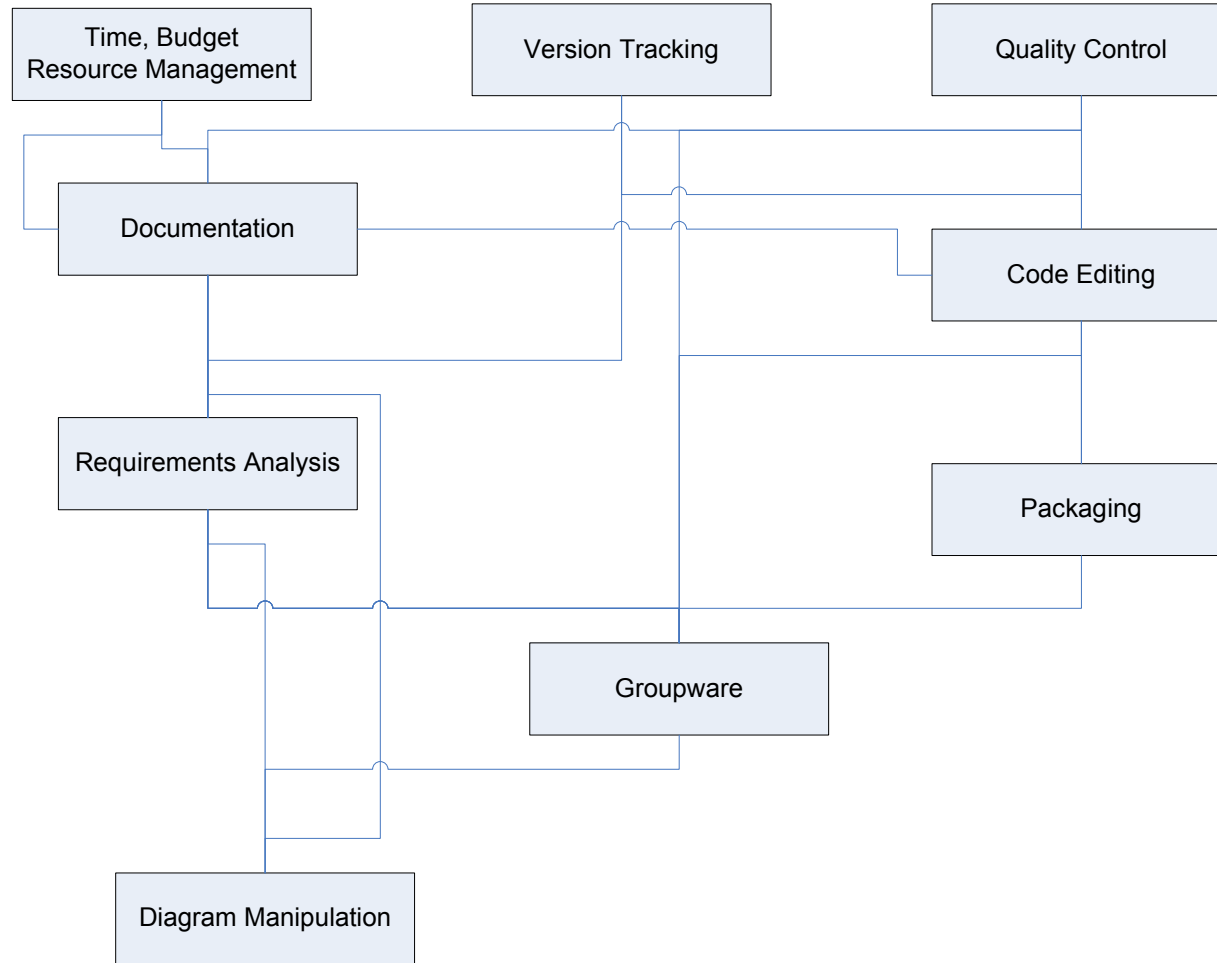


# Software development : Problem

- Software development requires all the support it can get because:
  - It is a highly systematic process due to its complexity
  - It is carried out by teams that need to cooperate on a subtask level
  - Requires reusability of components and services



# Software development : Activities





# Software development : Methodologies

- **Forward Engineering**
  - Analysis – Design – Implementation
- **Reverse Engineering**
  - Given the implementation create the specifications
- **Roundtrip Engineering**
  - Start anywhere, end anywhere (or nowhere, ever)
- **Re-engineering**
  - Restructure and rebuild (partially) an existing system to fit new requirements



# Software development : Need for tools

- **Development time: CASE tools**
  - Support software development and project management
  - Such tools exist only on the developers machine
- **Runtime: “Libraries, toolkits, frameworks, etc”**
  - Enhance functionality by pre-developed and re-used software
  - Such tools (software artifacts) are used at runtime



# CASE Tools: Definition

- CASE: Computer-Aided Software Engineering
- A suite of tools (toolbox) to support all aspects of the software development process, e.g.
  - Analysis and design diagrams
  - Source code creation
  - Data management repository
- The tools inside the toolbox must be able to cooperate

*CASE Tools are computerized applications supporting and partially automating software production activities [Fugetta]*



## CASE Tools : Goal

- Productivity enhancement
- Software quality
- Project management

*The tools should do the routine work.*

*Good developers write good software faster with CASE Tools.*

*Bad developers write more bad software in the same time.*





# CASE Tools : General requirements

- Support of the software development process and methodology
  - Create diagrams
- Supply basic functionality, do routine tasks automatically
  - Be able to support editing of code in the particular programming language, supply refactoring methods
- Features to enhance efficiency
  - Automatic code generation
- Features to enhance quality
  - Support of design patterns
- Intuitive use
- Integration with other tools
  - Code editor works with code repository



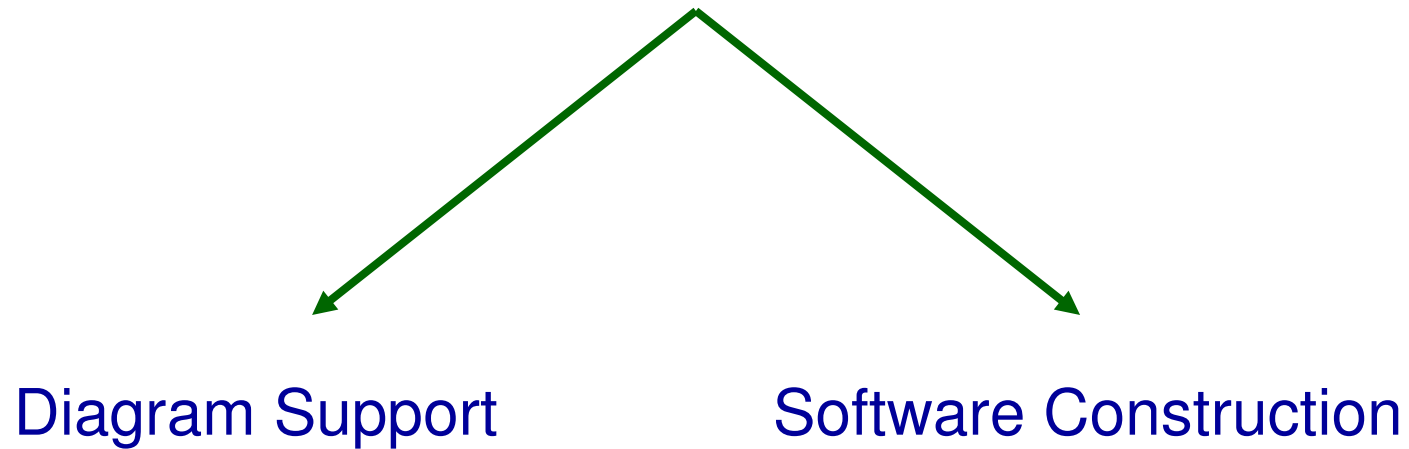
# CASE Tools: Classification

Planning	Analysis	Design	Implementation
<u>Upper Case</u> Support the analysis & design		<u>Lower Case</u> Support the construction & maintenance	
Integrated CASE (I-CASE)			

- *Currently there are tools for the entire cycle*



# CASE Tools : Standard functionality





# CASE Tools : Diagram Support

- Checks for syntactic correctness
  - the correct symbols are used
- repository support
  - storing diagrams, descriptions of diagrams and specifications
- checks for consistency and completeness
  - the same “object” is modelled by different aspects and diagrams
    - (the manual consistency and completeness check is time consuming)
- navigation to linked diagrams
- layering
- traceability
  - the way from requirements to code, so if a req is changed ...
- report generation
- system simulation
- performance analysis



# CASE Tools : Software Construction

- **Code generators**
  - they save time
  - the code is consistent with the design
  - change in reqs = > change in code
  - => database schemata
- **maintenance tools**
  - reverse engineering tools: from code to design models
  - analyze of program code and identify those parts that are most likely to be subject to change



# CASE Tools : Components

## Test Tools

- Fault injectors
- Bug Tracker
- Quality Control

## Project Management Tools

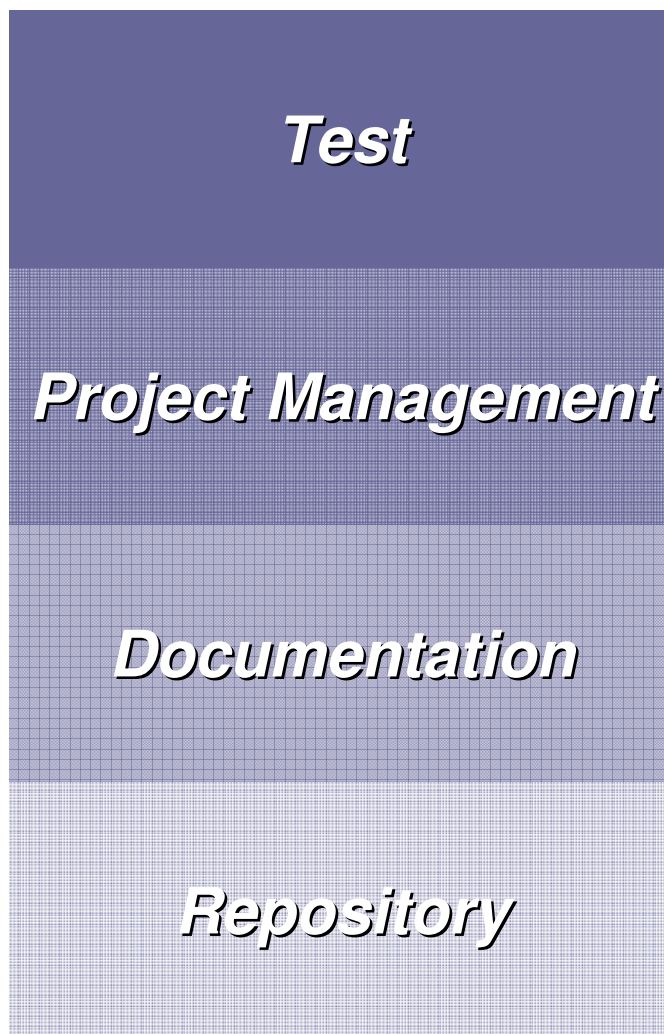
- Tasks and dependencies
- Resource planning
- Monitoring

## Documentation Tools

- Word processors
- Graphics tools
- Report generators

## Repository System

- Team development
- Integrity of artifacts
- Management of variants





# CASE Tools : Benefits and Difficulties

- **Benefits**

- The standardization of notations aids the communication within the team
- Automatic checks
- Reuse of design/code
- Code generation saves time
- Increase product quality

- **Difficulties**

- sometimes tools restrict you on how you should work
- the validation of correctness/completeness may create the illusion that the design indeed meets the requirements
- installation / training costs



# CASE Tools and UML : What is XMI?

- XMI = XML Metadata Interchange
- An XML schema or document type definition (DTD) for object modeling.
- Purpose:
  - enable easy interchange of metadata between modeling tools – based on UML
- It is extremely verbose as UML can handle almost any data modeling requirement.





# References

- **CASE Tools**
  - A Classification of CASE technology, Fuggeta, IEEE
  - Fundamentals of Software Engineering, Ghezzi et al, Prentise Hall International
  - Software engineering tools and environments : A roadmap, William Harrison, Harold Ossher & Peri Tarr
  - Environments to Support Collaborative Software Engineering, Cornelia Boldyreff, Mike Smith, Dawid Weiss, David Nutter, Pauline Wilcox, Stephen Rank, Rick Dewar
- **XML Metadata Interchange**
  - <http://www.oasis-open.org/cover/xmi.html>
  - <http://www.omg.org/technology/documents/formal/xmi.htm>
- **Tutorials**
  - UML
    - <http://www.agilemodeling.com/>
  - Rational Modeler
    - Online courses (non-free)
    - Build in tutorials
    - Build in user-guide
  - ArgoUML
    - <http://argouml.tigris.org/documentation/defaulthtml/manual/pt01.html>
    - [http://www.cee.hw.ac.uk/ophelial/tutorial/argoUser/argo\\_main.html](http://www.cee.hw.ac.uk/ophelial/tutorial/argoUser/argo_main.html)



## List of CASE tools (UML 2.0)

- Rational (IDE integration)
  - [XDE](#)
  - [Modeler](#)
  - [Software Architect](#)
- [ArgoUML](#)
- [MonoUML](#)
- [Visual Paradigm](#)
- [Microsoft Visio](#) (extra stencils for UML 2.0)
- .... More at [OMG Tool List](#)



## Tool Demonstration

- Rational Modeler
- ArgoUML

