



# SERIOUS

## Software Evolution, Refactoring of Operational Usable Systems

.....



### PROBLEM:

- Software intensive systems get larger and more complex
- Development costs and time increase

### STATE OF THE ART:

- European industry has done much research into new software engineering methods and technology, application is usually restricted to new systems
- Software intensive systems evolve during their lifetime
- During creation the quality of a product increases, but when the software has to be extended the quality will degrade



### THE SERIOUS APPROACH: SOFTWARE EVOLUTION



- Improve integration of software analysis and refactoring techniques in the day-to-day software process
- Define quality aspects in such detail that it can be applied and used in the early phases of software development and continued to be useable in the evolution phase

#### CONTACT

Project Leader: **John Pleunis**

Philips Medical Systems – The Netherlands ~ Tel: +31 40 27 62499, Fax: +31 40 27 62379

E-mail: [John.Pleunis@philips.com](mailto:John.Pleunis@philips.com) ~ Website: [/www.hitech-projects.com/euprojects/serious/index.htm](http://www.hitech-projects.com/euprojects/serious/index.htm)





# SERIOUS

## Software Evolution, Refactoring of Operational Usable Systems

.....

### ORGANISATION:

- Technical Evolution: provide methods and tools to support the technical evolution of software systems
- Evolutionary Software Engineering: Develop processes and methods to improve the software development models in industry
- Quality Driven Evolution: provide methods, tools, models and platforms for the quality control during system evolution

### METRICS:

Coupling

*“Coupling quantifies dependencies between software entities”*

Variability

*“Variability is the ability of a software system to be changed or customized”*

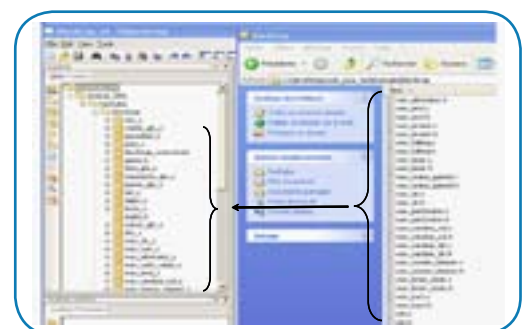
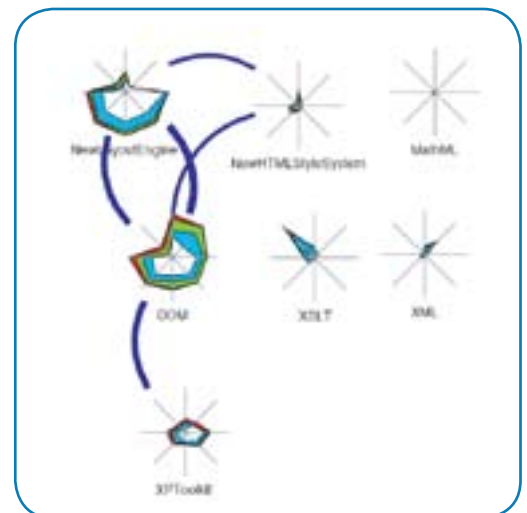
### TOOLS AND METHODS:

Visualisation

- Use Kiviat diagrams to visualise system properties and dependencies that would otherwise remain hidden

Reverse Engineering

- Use system reverse engineering and model annotation for safety and security analysis



### CONTACT

Project Leader: **John Pleunis**

Philips Medical Systems – The Netherlands ~ Tel: +31 40 27 62499, Fax: +31 40 27 62379

E-mail: [John.Pleunis@philips.com](mailto:John.Pleunis@philips.com) ~ Website: [www.hitech-projects.com/euprojects/serious/index.htm](http://www.hitech-projects.com/euprojects/serious/index.htm)



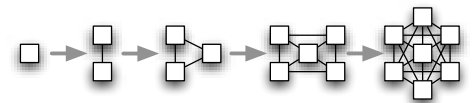


# SERIOUS

## Reengineering large systems - Tool support for static analysis

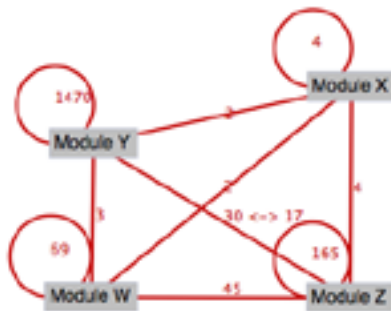
### PROBLEM: SOFTWARE DESIGN ERODES

- Structure impedes efficient changes
- How to *diagnose* and *cure* architecture/design weaknesses?



### SOLUTION: REENGINEERING SCENARIOS

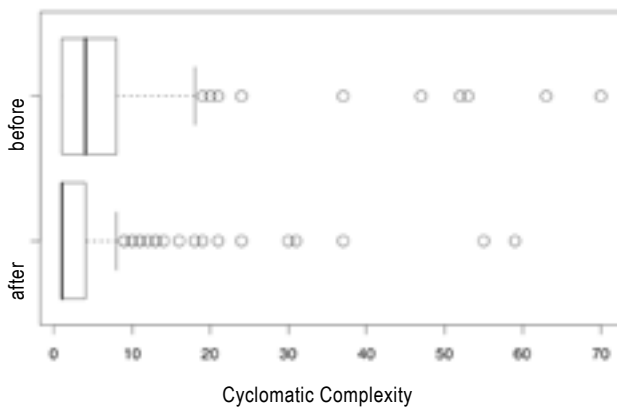
Fetch<sup>1</sup> is a robust static analysis tool chain, used to:



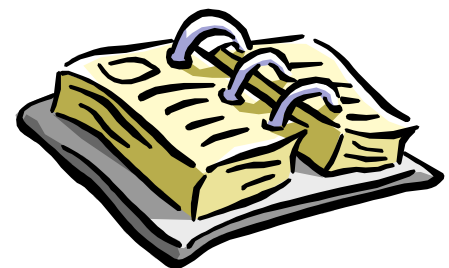
Visualize architecture/design to contrast actual with conceptual organization.



Identify exceptional entities as maintenance-prone components.



Quantify structural degradation or improvement across multiple versions (e.g., during refactoring).



The **Refactoring Handbook** documents techniques and methods supporting these scenarios.

(1) <http://www.lore.ua.ac.be/Research/Artefacts/fetch>

#### CONTACT

Project Leader: **John Pleunis**

Philips Medical Systems – The Netherlands ~ Tel: +31 40 27 62499, Fax: +31 40 27 62379

E-mail: [John.Pleunis@philips.com](mailto:John.Pleunis@philips.com) ~ Website: [www.hitech-projects.com/euprojects/serious/index.htm](http://www.hitech-projects.com/euprojects/serious/index.htm)





# SERIOUS

## Concern-based Software Comprehension

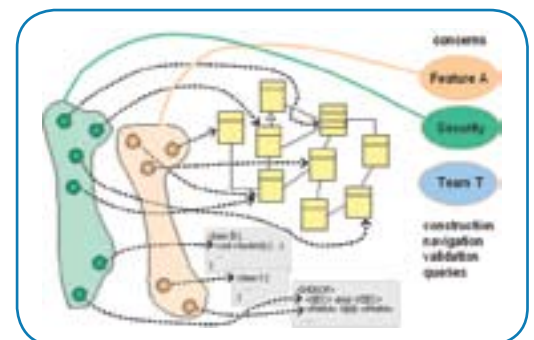
.....

### PROBLEM:

- Software is inherently multidimensional: there are always cross-cutting entities that are interesting for different stakeholders (*so-called concerns*)
- Software analysis needs are often focused on specific parts and aspects only

### STRUCTURAL ANALYSIS

- Stakeholders' interests are represented as concerns
- Concern representations are linked to software artifacts (e.g. model elements, source code snippets or documentation)
- CMT: Tool support for constructing, navigating, validating and querying the concerns
- Case study: Analysis of an OS used in Nokia's mobile phones



### BEHAVIORAL ANALYSIS

- Behavioral rules related to specific concerns are represented as *Behavioral profiles*
- Software runtime behavior is analyzed and validated against the defined rules
- Bebop: Tool support for Behavioral profiles
- Case study: Validation and analysis of an Eclipse extension

