

# **Component-Based Software Engineering: New approach in managing large-scale software**

**Seminar**

**August 31st, 9:00-12:00**

**Hörsalen ABB Corporate Research Kopparlunden**

## **Program**

**09:00-09:15 Introduction**

**09:15-10:00 Ivica Crnkovic:** Component-based development process

**10:00-11:00 Judith Stafford:** Architecture: The Key to Component-based Development

**11:00-12:00 Kurt Wallnau:** Managing Design Complexity Arising from Commercial Software Components.

There is a high trend of using components, especially COTS (commercial off the shelf) components, in software development. Both customers and producers share the enthusiasm for the CBSE approach because of the obvious advantages: The development time dramatically decreases, the usability of the products increases, the production costs usually decrease, and so on. Indeed, in many domains significant improvements in efficiency of development have been achieved. However, inclusion of components, over which the producer does not have complete control over, increases the risk of getting unexpected results. Even good components can corrupt a good product if they are managed in the wrong way. In some domains, such as industrial automation, this risk is unacceptable, and additional measures are required to minimize it. One of the problems is the lack of established engineering procedures and, in general, a lack of knowledge of CBSE.

This seminar gives an overview of some aspects of CBSE:

- An overview of Component-based development process
- A role of Software Architecture in Component-Based Software engineering
- Managing Design Complexity Arising from Commercial Software Components

The seminar is intended to software engineers, specialists and development managers.

## **Abstracts**

### **Ivica Crnkovic: Component-based software engineering – a new paradigm of software development**

Software development meets many challenges today: The development life cycle is getting shorter, the software application domains expand rapidly, pressure on the software development costs are enormous and the requirements for integration of different domains are high. At the same time the technology in both software and hardware is changing extremely fast. The key to the future success if not survival of software companies will be the ability to manage complexity and rapidly adapting to change. Without a possibility to reuse system components already developed it will be not possible to meet the requirements from the market. This implies that the development process must change focus – from programming-intensive activities to reuse, integration, standards, management of complex and flexible structures, finding

proper solutions, tradeoff analysis and marketing survey. These changes require changes in development procedures, tools, but also developers' skills and organizational changes. This presentation gives an overview of the demands on the software development companies, and of the development process of component-based systems.

### **Judith Stafford**

#### **Architecture: The Key to Component-Based Development**

In this talk I will describe software architecture and architectural style, and their use to help predictably design component-based systems that satisfy system requirements. The architecture of a software system is the collection of components and connectors that comprise the system along with the attachments among them. In component-based development the architectural components are embodied by implementation components and the connectors are embodied by the communication protocols of the framework into which the component implementations are deployed. A system's architecture can be used to reason about system quality in terms of abstract properties of components thereby providing

- an opportunity to analyze tradeoffs among various potential architectures,
- design for variability to support product lines, and to
- reason about the potential for an assembly of pre-existing components
- to work together as desired.

Further benefit can be achieved if architectural styles are used as a base for system development. Architectural styles are collections of component and connector types along with constraints on their interactions. Styles have been defined with focus on the both functional and extra-functional aspects of a system. Architectural styles support a prefabricated approach to component-based development, thereby providing support for streamlined architectural design and faster system development.

### **Kurt Wallnau:**

#### **Managing Design Complexity Arising from Commercial Software Components**

Use of commercial software components offers great promise, but also introduces complexities not adequately addressed by current design methods. For example, each commercial component in a system is a locus of independent evolution over which the designer has limited control. The need for vendors to sustain market differentiation requires innovation and frequent component releases, leading to instability of component features and mismatch of components across different vendors. We have developed the idea of component ensemble and their related design processes for exploring the unstable and obscure design spaces resulting from aggressive use of commercial components.

This presentation defines ensemble and describes their representation as blackboards, and illustrates their use with design problems drawn from our own experience.

**Kurt Wallnau** (<http://www.sei.cmu.edu/staff/kcw/>) has 15 years of software development experience in industry, defense and research. Wallnau's principal interests are COTS software integration and component-based software engineering.

Wallnau's most recent tour at the Carnegie Mellon University Software Engineering Institute began in 1994. Since that time Wallnau's primary areas of investigation at the SEI have been in the techniques and technologies of COTS software integration, and component-based software engineering (CBSE). In the COTS arena, Wallnau established the COTS-Based Systems "testbed" for conducting focused ("situated") product and technology evaluations. Wallnau has led the CBS effort to define a flexible COTS software evaluation methodology, one that is linked directly with an overall COTS-sensitive design process. In the CBSE arena, Wallnau co-organized the 1998 International Conference on Software Engineering (ICSE) workshop on CBSE; he is co-organizing the follow-on workshop, likewise affiliated with the 1999, 2000 and 2001 ICSE. Wallnau has several publications in the area of COTS- and Component-Based Software Engineering and co-author of several books. His book Building Systems from Commercial Components is recently published.

**Judith Stafford** (<http://www.sei.cmu.edu/staff/jas/>) is a Senior Member of the Technical Staff at Carnegie Mellon University Software Engineering Institute since 2000. Before that she worked at University of Colorado where she obtained a Ph.D. degree in Computer Science.

Her research interest is focused on Software Architecture. She is a member of research teams working with Predictable Assembly from Certifiable Components, Architecture Tradeoff Analysis Method, Architecture Representation, Attribute-Based Architectural Styles, etc. Stafford is a coorganiser of ICSE 2001 CBSE workshop. She has several publications in the area of Component-Based Software Engineering and Software Architecture and she is co-author of several books being published.

**Ivica Crnkovic** (<http://www.idt.mdh.se/personal/icc/>) is professor in Industrial Software Engineering at Mälardalen University, Department of Computer Engineering, and a lab-leader at Computer Science Laboratory. Previously he worked at ABB, where he was responsible for development environments. He was a project leader and manager of a group who developed Software Development Environment tools and methods for distributed development and maintenance of real-time systems. He leads the Industrial IT research group at Mälardalen University. He is member of the Configuration Management group at Verkstadsindustrier, Sweden. He is co-organizer and a member of program committee of several workshops related to Component-Based Software Engineering and Configuration Management. His main research interests are Component-based Software Engineering, Software Configuration Management, and in general Software Engineering.