Workshop Abstracts

W1a Reverse Engineering and Legacy Systems
Cornelia Boldyreff, University of Durham, UK
Rob Davies, BT Laboratories, UK

Often legacy systems are mission critical to an enterprise, but lack of systemic evolution has resulted in systems which are difficult to change. Reverse engineering offers the possibility of recovering higher level views of the legacy systems and thus establishing the vital system understanding needed to support maintenance. Through application of reverse engineering techniques and tools to legacy systems, it is also possible to recover assets exploitable in evolving the legacy systems.

These recovered assets also can be used as the basis for organizing a component based approach to system development and more generally as a basis for a reuse program underlying future development. Rather than continuing to maintain and evolve existing legacy systems in isolation, through these approaches, it may be possible to identify product lines across a number of legacy systems or elements of those systems leading to a more unified form of support for related legacy systems. This workshop will consider these ideas, as well as pick up on issues raised in the related paper sessions.

W1b Configuration Management
Shirley Williams and Rachel McCrindle,
University of Reading, UK

Software configuration management is now a recognized discipline for controlling the evolution of large software systems and is acknowledged as adding considerable value to the control and visibility of the software development and maintenance processes. This workshop will focus on, but will not be restricted to, two main areas of interest. Firstly it will explore whether current configuration management techniques and practices are adequate for controlling the evolution of today's computer systems such as those of a distributed, embedded or multimedia nature. Secondly, it will consider what features and facilities should be incorporated into future tools and environments for facilitating the configuration management of both legacy and new software systems. Issues will cover both technical and managerial aspect of software configuration management.

W1c Object-Oriented Middleware, Frameworks for Systems Integration
Sri Raghavan, Digital Equipment Corporation, USA
Rudolf K. Keller, University of Montreal, Canada

The objective of this workshop is to share an understanding and synthesize the characteristic strengths and limitations of popular frameworks such as CORBA, Java, ActiveX/DCOM, and WWW for OO system integration efforts; and issues and approaches for successfully practicing OOAD in these contexts. Key challenges in object technology lie in inventing the right set of object abstractions and organizing them so that the objects can interoperate, collaborate, and evolve efficiently. In build-from-scratch projects there is considerable freedom in tackling these challenges. However it is different in systems integration projects which are driven and constrained by the existence of legacy systems and a strong a priori commitment to the use of off-the-shelf components. Since these projects are typically executed in tumultuous organizational context there are significant implications for the practice of OOAD methods.
**W2a  What Makes for Project Success or Failure?**

*John Jenkins, City University, UK  
Rob Kusters, TUE, Netherlands*

Most commentators argue that the reason for IS project failure lies not in Technology but in People. A particular cause being the innate optimism of Systems people which results in the adoption of unachievable targets which are passed off as estimates. This workshop will explore the evidence for this thesis and hopefully identify ways in which the problem can be alleviated. The issue of risk management will be explored.

**W2b  Net-Centric Computing**

*Scott Tilley, Software Engineering Institute, USA  
Margaret-Anne Storey, University of Victoria, Canada*

The explosion of interest in the World Wide Web has given rise to many new developments. One of them is Net-Centric Computing (NCC), the underlying principle of which is a distributed environment where applications and data are downloaded from network servers on an as-needed basis. The adoption of NCC will have significant impact on users, developers, and administrators. Some users may chaff at the limitations of NCC brings with it: the removal of "personal" in PC means users will no longer be able to significantly alter their desktop environment. However, for administrators this means a potential reduction in the cost and complexity of managing a corporate network. For developers, NCC offers an opportunity to greatly increase their customer base: applications written in a NCC-aware programming language, such as Java, means writing code once and having it immediately accessible on multiple platforms. It also means a different development environment, a new deployment model (renting applications versus buying), and new concerns about security. Through focused presentations and open discussion, this workshop will explore NCC and its potential impact on software users, application developers, and system administrators.

**W2c  Component Based Development**

*Ben Whittle, British Telecom plc, UK  
Andy Ormsby, University of Sussex, UK*

Component based development is the latest buzz phrase in the reuse world, but what is a component and how does it differ from the modules and classes that we already deal with everyday? In this workshop we will explore the idea of component based development, what difference will it make in terms of development methods, tools and techniques. We will pick up on themes from previous sessions in STEP'97 (reuse, COTS, OO, legacy systems) as well as bringing in specific ideas related to component technology. The workshop will covers issues such as:

- Is component based development really a paradigm shift?  
- How big is a component - and who cares?  
- What makes a component any different to a traditional software module?  
- Is a component a technical entity or a business entity?  
- What is the relationship between COTS and component based development?

**W2d  Structured Testing**

*Martin Pol, IP/Software Control Testen, Netherlands  
Erik van Veenendaal, KEMA, Netherlands*

This workshop will focus on the testing of complete information systems, not just the software, but also the related products and system support features. After a short introduction, the main testing pitfalls will be identified. Next a generic testing approach, applied in over 200 companies of different types will be presented covering the four pillars of structured testing: the testing lifecycle, testing techniques, infrastructure and tools and finally organizational issues. Two case study presentations will show the application of structured testing. Participants will be encouraged to interrupt the presentations and to start discussion which will provide feedback on the earlier collected pitfalls.
W3a  Choosing Techniques and Tools for Requirements Specification
Sara Jones, University of Hertfordshire, UK
Mehmet Ozcan, Sheffield Hallam University, UK

The aim of this workshop will be to promote discussion of factors which may significantly influence the choice of methods, tools and notations for requirements specification in a particular project. Discussion will focus on tools and techniques which support the generation of a requirements specification document, but other activities, such as the elicitation and validation of requirements, will also be considered in-so-much-as they affect the choice of tools for specification.

The outcome from this workshop will be the production of an initial categorization of 'project types' and some preliminary notes regarding experiences and lessons learned as to the kinds of methods, tools and techniques which can be, and have been, successfully applied in projects of these types. Where possible, the emphasis will be on real world and industrial projects.

W3b  Effectiveness of Software Process Improvement
Erik Rodenbach and Rini van Solingen
Schlumberger Retail Petroleum Systems, Netherlands

In order to manage and improve software development, industry uses many methods and techniques that focus on the development process of software, the so called "Software Process Improvement" (SPI). SPI aims at improving time-to-market, cost and quality of software, in particular by focusing on the process by which it is developed. However, most industrial organizations sell products, not processes. Therefore the effectiveness of process improvements must be visualized in order to decide whether 'changes' are really 'improvements'. This workshop will focus on the effectiveness of software process improvement. Beside a small overview presentation on SPI-methods, some examples from practice will be presented that showed beneficial results of SPI.

W3c  Distributed Cooperative Working
Linda Macaulay, UMIST, UK
Pearl Brereton, University of Keele, UK

New opportunities arise from the field of computer supported co-operative working where the merging of computing and telecommunications enables teams to function with individual members residing in different physical locations. The aim of this workshop is to explore the potential of new technologies to support distributed software engineering teams. The workshop will focus on two issues, firstly what technology is available and ready for use today, and secondly whether technology is in itself sufficient for success.

The outcome of the workshop will be the identification of those factors which affect the success of distributed co-operative working for software engineering teams.