About James Noble and his research

Professor James Noble's research focuses primarily on two areas of software engineering: the design of the users' interface (the parts of software that users have to deal with every day), and the programmers' interface (the internal structures and organisations of software that programmers see only when they are designing, building, or modifying software). His research in both these areas is coloured by his longstanding interest in object-oriented approaches to design. Topics he has studied range from aliasing and object ownership, design patterns, agile methodology, visualisation and computer music, through to postmodernism and the semiotics of programming.

James Noble is Professor of Computer Science and Software Engineering at Victoria University of Wellington (VUW). After gaining both a BSc (Hons) and a PhD from VUW, he worked for the University of Technology in Sydney, and then at the Microsoft Research Institute, Macquarie University, before returning to VUW in 1999 as a lecturer.

He is the author of Small Memory Systems: Patterns for Systems with Limited Memory (with Charles Weir), and has published many papers on object-orientation design patterns, aspects, software visualisation and software engineering for international academic conferences and journals.

His current research projects include:

- ***Agile Development*—research focused on smarter software development (see story on reverse)**
- ***Design patterns and ownership types*—techniques for improving program designs and their reliability**
  Software is an integral part of everyday life; however, its shortcomings have never been more apparent. A word processor crashing when asked to save a document, or a telephone exchange refusing to connect emergency calls are both symptoms of our inability to engineer software that works correctly and reliably. Prof Noble and his team are directly addressing this problem by identifying important recurring design patterns in the specification and design of software systems, and then developing tools based on ownership types to verify the software.
- ***Postmodern programming in architecture*—a new way of thinking about programming**
  The long-held modernist viewpoint of many software developers is that software systems should be composed of neat, uniform components (like lego blocks), and that the two most important aims of a program are to be correct and efficient. The postmodern view being researched by Prof Noble and his colleague, Prof Robert Biddle, is that the very success of software engineering is fundamentally changing the way programmers will work in the future, and how software development projects will be carried out. The researchers' view is that successful computer systems will more likely be combinations of widely different kinds of components, built by pulling together pieces of pre-existing software – from within companies, from open-source repositories, or via Google. Correctness and efficiency are likely to become less important while the ability to repair program mistakes fast will increase in importance.
- ***"Grace"*—an open-source, object-oriented programming language for education**
  Object-oriented programming is widely taught in introductory computer science courses; however, without a single programming language as the obvious choice for teaching the subject, Professor Noble is collaborating with fellow language researchers from around the world to create a new language that will make teaching programming a simpler, easier process.

The Benefits of Being Agile

Research focused on smarter software development

Victoria University scientists are contributing their expertise to improving a new way of developing software that is faster, more flexible and more affordable.

The Agile approach, which is rapidly being adopted by many of the world's software giants, has been created to address some of the issues with the traditional software development process which dates back to the 1960s and has its roots in computerisation for the space race and weapons development.

Due to cost overruns, the United States Defence Department developed its own software, rather than buying it. However, it put in place a relatively bureaucratic process in which as much as half the effort can go into planning, detailed documentation and formal sign off by the parties involved before development work begins.

Computer Science Professor James Noble says Agile methods involve self-organising teams that develop software through iterative and incremental work cycles in close collaboration with their customers. Changes can be made at any stage and the product is tested along the way to ensure early detection of problems.

"Anecdotal evidence shows it speeds up software development and can show more quickly where problems lie or whether the project is doomed," says Professor Noble.

But, despite its burgeoning popularity, many aspects of the Agile process have not been researched until now.

Professor Noble's research group has investigated how teams should self-organise, studying 58 Agile practitioners from software organisations in New Zealand, India and North America to understand how they go about self-organising. The information has been used to develop a process others can follow.

Professor Noble says it's a challenging area as there is no boss telling people what to do and it's a high pressure, constantly changing environment with deadlines to meet.

The research has shown that support from senior management within an organisation is vital and that projects go more smoothly when customers play an active role.

To find out more about Professor Noble's research, please contact him directly by calling **+64-4-463 6736** or emailing him at **james.noble@ecs.vuw.ac.nz**

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON