First ISEW Cleanroom Workshop Summary

Graeme Smith

Q-Labs GmbH
Kaiserslautern, Germany

gs@q-labs.se

The first Cleanroom workshop to be held as part of the International Software Engineering Week in association with ICSE-18 took place in Berlin on 26 March 1996. Sadly, Harlan Mills, the principal founder of Cleanroom passed away earlier this year and it is fitting that the ICSE 18 proceedings should be dedicated to his memory. That this first Cleanroom workshop should take place in this year is a clear sign that his work will live on.

Background:

Cleanroom Software Engineering is a set of technical and management processes for engineering large software systems with high quality and certified reliability [1]. It was developed within the Federal Systems Division of IBM during the 1980s by a team led by Harlan Mills. Cleanroom has a long pedigree, being the culmination of 20 years of research and experience in Software Engineering by Mills and his colleagues, starting from the work on structured programming [2], progressing through Box Structure Development [3], reliability certification [4] and more recently the approach to certification based on Markov chains [5].

The Cleanroom process achieves quality and productivity gains relative to other development approaches by concentrating on *defect prevention* (hence the name, Cleanroom) rather than the currently more widespread defect detection practices. Defect prevention is achieved by developing software in many small invention steps followed by immediate verification, with the consequence that later invention steps are built on correct foundations. Defect prevention improves quality, but it also achieves increases in productivity and lead-time by reducing rework.

The principal parts of the Cleanroom process are:

- rigorous specification
- stepwise refinement and verification
- team-based verification in place of unit testing and debugging
- incremental development to scale up to development of large systems
- team-based development
- reliability certification based on statistical usage testing

For further reading about the Cleanroom process itself see [6], [7] and experience reports on its use [8].

Objectives of the Workshop:

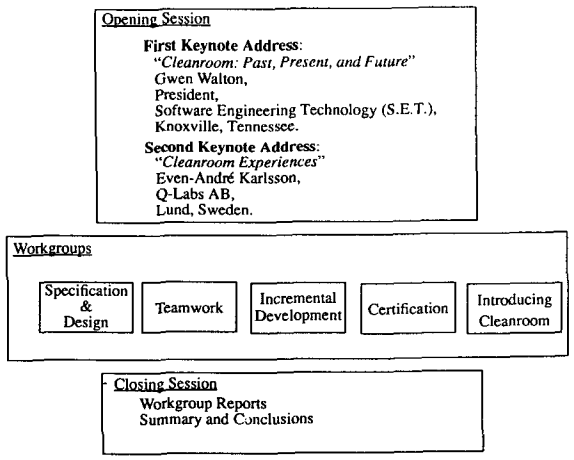
The purpose of the workshop was to bring together those currently active in Cleanroom, whether in research or in industry, to share experiences and to discuss common problems.

The theme of the workshop was "The Further Evolution of the Cleanroom Process." Cleanroom has now been in use for about 9 years so perhaps it is an appropriate time to look again at the original goals and principles which it embodies, in the light of practical experience. This led to the following specific objectives:

- to re-visit the original goals of the Cleanroom process and the principles that if embodies to verify that they are still appropriate to the needs of today's software developers,
- to see to what extent the original goals have been met in the first nine years of Cleanroom Software Engineering,
- to study the evolution of the process to date and to consider the need for future evolution.

Structure of the Workshop:

In trying to meet these goals, the workshop was structured into three parts as follows:



The idea behind this structure was that the keynote speakers should re-state the objectives and principles of Cleanroom and summarize the experience with its use. Then, the workshop should split into small working groups, each concentrating on a single aspect of the Cleanroom process with the intention of evaluating its contribution to the overall goals of the process and considering how that aspect should evolve in the future.

The First Keynote Address

The abstract of the first keynote address is reproduced below. A full copy of the paper is to be found in the proceedings of the workshop [9].

Title: "Cleanroom: Past, Present, and Future"

By: Gwendolyn H. Walton, President, Software Engineering Technology, Inc.

and

J.H. Poore, Professor, University of Tennessee, USA.

Abstract: Although the intellectual contributions of Dr. Harlan D. Mills were vast and varied, we will focus on the collection of his ideas that are now known as Cleanroom software engineering. The "Cleanroom" name was motivated by Harlan's point-of-view that the key to high quality software was to make it correct by design and to follow a rigorous design process. Given the complexity of software being developed today, Mills' original Cleanroom principles are at least as applicable to modern software projects as they were when he first defined them. In this paper we re-examine Mills' key Cleanroom principles, look at the present application and commercialization of Mills' ideas, and provide some comments on our vision of the future in software engineering research and performance.

The Second Keynote Address

A summary of the second keynote address is provided below. A full copy of the paper is to be found in the proceedings of the workshop [9].

Title: "Cleanroom Experiences"

By: Even-André Karlsson, Q-Labs AB, Lund, Sweden.

Summary:

The address covered the following points:

- Cleanroom objectives: what do we try to achieve with the Cleanroom techniques
- Current practices: degree of implementation (how have we used these things) and successes (some examples)
- Open issues: difficulties, theoretical and practical

Each of these points was considered in relation to the following specific topics:

- Black box specifications
- Usage modelling
- Stepwise refinement and functional verification
- Statistical usage testing
- Incremental development
- Teamwork and frequent reviews

As a general conclusion,

- the objectives are sound
- Interesting and challenging technical areas
- Organizational areas picked up extensively
- Technical areas harder, requires more support, e.g. text book

The Workgroups:

Specification & Design

The workgroup on specification and design suggested the following solutions to some familiar problems:

- How detailed should a black box specification be?
  - a black box spec can be developed iteratively
  - the greater the level of detail in the black box the harder the task of validating completeness and correctness
  - the criterion for the level of detail in a black box specification depends on the purpose and function
- How to check the correctness of a black box specification?
  - tracing to requirements statements
  - formal inspection
  - prototyping
  - executable specifications
- Why is it so important to have an 'implementation-free' specification?
  - no design bias
  - transitions local, while in the state box they are not local
  - understandable by customer
- How can timing considerations be included in a black box specification?
  - timing requires special notation
  - the SCR method includes timing definitions and an approach to combining this into Cleanroom was presented to the workgroup [10].

Teamwork

The workgroup on teamwork identified the following major issues in connection with the use of teamwork on Cleanroom projects:

1. Team/Individual performance evaluation
   - composition? (e.g. 40/60)
   - mechanism
2. Does the team/organization structure impact the use of Cleanroom principles?
   - line/project roles
   - cross-functional vs function-specific teams
3. How is large-scale implementation of teamwork done effectively?
   - pilot to large-scale
   - project-specific
   - ongoing support — training, coaching, etc
4. How is a common culture developed for the effective use of Cleanroom in a team-based organization?
   - geographic dispersion
   - cultural differences (communication)

The following main conclusions were presented by the workgroup:

1. It is widely recognized that high-powered teams are required for Cleanroom but there is little guidance on team/organizational issues.
2. That is changing with current work on Team Assignment Model [11] (i.e., team empowerment, negotiation, work schedule, work routines).
3. Cleanroom reviews require effective communication, timely feedback, etc. — review guidelines needed to support reviews at all levels of design.

Incremental Development

The following main findings were reported by the workgroup on incremental development:

1. The ideal size parameters for optimum use of incremental development are:
   - development time: 6 - 8 weeks per increment
   - team size: 4 people
2. A major constraint on the successful development of an incremental construction plan is a good understanding of the product architecture
3. Configuration Management becomes more critical when using incremental development.
4. Cross-dependencies to other Cleanroom practices: teamwork only.
5. Increments on project level put constraints on the team level.

Certification

The certification workgroup identified a number of issues to be overcome in order to advance in this area and gave general indications as to whether the solutions are to be found in further research, better education and training or further experience.

	Issues	Further Research	More Experience	Better Education
1	Random data generation	X	X	
2	Abstraction in usage models	X	X	X
3	Many test cases required for 'good statistics'		X	
4	Need for reference values for reliability data		X	X
5	Usage model complexity measures	X		
6	Help to assign probabilities in usage models	X	X	
7	Need for consistent data reporting			X
8	Stratification of different types of usage		X	
9	Modularization of usage models	X	X	
10	Multi-user models	X		
11	The oracle problem	X	X	
12	Support for automatic testing	X		

Introducing Cleanroom

The workgroup identified the following milestones on the way to successful introduction of Cleanroom into an organization:

1. Establish a team mentality
2. Secure management "buy-in"
3. Bring about an attitude change amongst project staff
   - Cleanroom requires a different focus
4. Establish the right training
5. Plan a strategy of incremental introduction
   - process maturity is very important

Conclusions

In the final session of the workshop, a synthesis of the conclusions of all of the workgroups was sought. The overall conclusion was that the Cleanroom process and the engineering principles which it embodies are just as relevant today as when they were first proposed.

Looking at the experience of the years of application of Cleanroom,

- the anticipated project performance gains are realizable with "full adherence" to the Cleanroom process
- many projects have benefitted from more limited performance gains with "partial adherence" to Cleanroom, i.e., through use of a selected subset of the Cleanroom practices:
  - teamwork
  - incremental development
  - frequent reviews
- Adoption of the Cleanroom practices has been steady but not spectacular, many of them have found their way into other development methods e.g., incremental development is widely practised.
- The path to greater uptake of Cleanroom lies in improved technology transfer activities rather than in extension or evolution of the process itself.

References:

1. "Cleanroom Software Engineering", H.D. Mills, M. Dyer and R.C. Linger; IEEE Software Sept 1987.
2. "Structured Programming: Theory and Practice", R.C. Linger, H.D. Mills, B.I. Witt. Addison-Wesley 1979.
3. "Principles of Information Systems Analysis and Design", H.D. Mills, R.C. Linger, A.R. Hevner. Academic Press 1986.
4. "Certifying the Reliability of Software", P.A. Currit, M. Dyer, H.D. Mills, IEEE Trans. S.E., vol 12 No 1, Jan 1986.
5. "Markov Analysis of Software Specifications", J.A. Whittaker and J.H. Poore, ACM Trans on Software Engineering Methodology, vol. 2 No 1, Jan 1993.

6. "Engineering Software under Statistical Quality Control", R.H. Cobb and H.D. Mills, IEEE Software, Nov 1990.
7. "Cleanroom Process Model", R.C. Linger, IEEE Software, Mar 1994.
8. "Cleanroom Software Engineering for Zero-Defect Software", R.C. Linger, Proc, 15th Int'l Conf. on S.E., IEEE-CS press 1993
9. "Proceedings of the First ISEW Cleanroom Workshop, Berlin, 1996", IEEE. (to be published).
10. "Integrating SCR Requirements into Cleanroom Software Engineering", C. Bunse and E. Kamsties, in [9].
11. "Establishing Effective Team Routines for Cleanroom Support", S. Becker, T.M. Janzon and B. Nilsson, in [9].

Editor's Filler

Pass the soap please ... wasn't that refreshing?

It reminds me of a joke ... you know you should always carry a bar of soap with you when you go on a boat, that way if you fall overboard you can wash yourself ashore..

Cough, cough.. now for something entirely different.

Did your check out the book reviews yet?

Aren't there a lot of them!!!

As you can see, when we skip an issue to give you the readers a conference proceedings, the book reviews pile up.

The next proceedings you will receive is **SIGSOFT'96** — as the November issue ... just in case you were wondering.

That means that the September issue is the last issue this year that I have to edit .... Wow!

Current Research Directions in Software Engineering for Parallel and Distributed Systems

***Innes Jelly and **Ian Gorton**

Sheffield Hallam University, UK
i.jelly@shu.ac.uk
*** CSIRO Division of Information Technology**
Australia
iango@syd.dit.csiro.au

Introduction

In March 1996, the First IFIP Workshop on Software Engineering for Parallel and Distributed Systems was held in Berlin, Germany. The two day workshop was co-sponsored by the German Computer Society GI (Gesellschaft fur Informatik), and organised in association with the International Software Engineering Conference– ICSE 18.

The aim of the workshop was to provide a forum for exchange of information and publication of the latest technological and theoretical advances in software engineering for parallel and distributed systems. Our previous experience of running short workshop on this topic in Aachen, Germany (1993), Como, Italy (1994) and Hawaii, USA (1995) had indicated that there was a growing need for this specialized event [1], [2], [3]. The International Programme Committee was formed from a group of experts in different countries and application areas, all of whom were enthusiastic to explore and publicize contemporary research in parallel and distributed software engineering.

In this report we discuss the rationale for the workshop, detail its outcomes and look at some of the research issues that it highlighted. The proceedings of the workshop are published by Chapman and Hall [4], and full information is available on the PDSE Web page: http://www.dcs.shef.ac.uk/~prc/PDSE.html

Rationale for Workshop

Many software applications require the use of explicit parallel programming techniques in order to meet their specification. Parallelism is needed to exploit the processing power of multi-processor systems in order to achieve high performance, to provide fault-tolerance and reliability in safety-critical and real-time systems, and to deal with physically distributed computing resources. While the range of existing software and hardware technology that can be employed in parallel and distributed systems development is massive, a set of underlying problems concerned solely with the use of parallelism can be identified.

These include:

- identification of problem-domain and solution-domain parallelism