The South African Institute of Computer Science and Information Technology

The 1997 National Research and Development Conference

Riverside Sun
Vanderbijlpark
13 & 14 November

Hosted by

Potchefstroomse Universiteit
vir Christelike Hoër Onderwys

The Department of Computer Science and Information Systems
Potchefstroom University for Christian Higher Education
Vaal Triangle Campus

PROCEEDINGS

Edited by L.M. Venter & R.R. Lombard
The South African Institute of Computer Science
and
Information Technology

Proceedings
of the
The 1997 National
Research and
Development
Conference
Towards 2000

Riverside Sun
Vanderbijlpark
13 & 14 November

Edited by
L.M. Venter
R.R. Lombard
©1997 Copyrights reside with the original authors who may be contacted directly

ISBN 1-86822-300-0

Printed and Binded by Xerox Printers, Potchefstroom

The views expressed in this book are those of the individual authors
Foreword

This book contains a collection of papers presented at a Research and Development conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT). The conference was held on 13 & 14 November 1997 at the Riverside Sun, Vanderbijlpark. Most of the organization for the conference was done by the Department of Computer Science and Information Technology of the Vaal Triangle Campus, Potchefstroom University for Christian Higher Education.

The programming committee accepted a wide selection of papers for the conference. The papers range from detailed technical research work to reports of work in progress. The papers originate mainly from Academia, but also describe work done in and for Industry. It is hoped that the papers give a true reflection of the current research scene in Computer Science and Information Technology in South Africa. Since one of the aims of the conference is Research development, the papers were not subjected to a refereeing process.

A number of people spent numerous hours helping with the organization of this conference. In this regard, we wish to thank the members of the Organizing committee, and the Programming committee who had very little time to screen the abstracts and compile the program. A special thanks goes to the secretary of the department, Mrs Helei Jooste, whose very able work was interrupted by the birth of her first child.
Organizing Committee

Conference General Chairs
Prof. J.M. Hattingh (PU for CHE)

Organizing Chair
Prof. Lucas Venter (PU for CHE)

Organizing Committee
Mrs. S. Gilliland
Mr. J.P. Jooste
Mr. R.R. Lombard
Mrs. M. Huisman

Secretariat
Mrs. H. Jooste

Program Chair
Prof. A de Waal (PU for CHE)

Program Committee
Prof. D. Kourie (UP)
Prof. C. Bornman (UNISA)
Prof. L.M. Venter (PU for CHE)
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreword</td>
<td>i</td>
</tr>
<tr>
<td>Organizing Committee</td>
<td>ii</td>
</tr>
<tr>
<td>List of Contributors</td>
<td>vii</td>
</tr>
<tr>
<td>Software Objects Change : Problems and Solution</td>
<td>1</td>
</tr>
<tr>
<td>S.A. Ajila</td>
<td></td>
</tr>
<tr>
<td>Liming-like Curve Constructions</td>
<td>26</td>
</tr>
<tr>
<td>M.L. Baart and R. McLeod</td>
<td></td>
</tr>
<tr>
<td>A Model for Evaluating Information Security</td>
<td>27</td>
</tr>
<tr>
<td>L. Barnard and R. von Solms</td>
<td></td>
</tr>
<tr>
<td>Integrating Spatial Data Management and Object Store Technology</td>
<td>31</td>
</tr>
<tr>
<td>S. Berman, S. Buffler and E. Voges</td>
<td></td>
</tr>
<tr>
<td>Metamodelling in Automated Software Engineering</td>
<td>32</td>
</tr>
<tr>
<td>S. Berman and R. Figueira</td>
<td></td>
</tr>
<tr>
<td>Using Multimedia Technology for Social Upliftment in Deprived Communities of Southern Africa</td>
<td>33</td>
</tr>
<tr>
<td>L. Bester and E. de Preez</td>
<td></td>
</tr>
<tr>
<td>Extending the Client-Server Model for Web-based Execution of Applications</td>
<td>36</td>
</tr>
<tr>
<td>L. Botha, J.M. Bishop and N.B. Serbedzija</td>
<td></td>
</tr>
<tr>
<td>Access Control Needs in an Electronic Workflow Environment</td>
<td>45</td>
</tr>
<tr>
<td>R.A. Botha</td>
<td></td>
</tr>
<tr>
<td>The Use of the Internet in an Academic Environment to Commercially Supply and Support Software Products</td>
<td>51</td>
</tr>
<tr>
<td>B. Braude and A.J. Walker</td>
<td></td>
</tr>
<tr>
<td>Explanation Facilities in Expert Systems Using Hypertext Technology</td>
<td>63</td>
</tr>
<tr>
<td>T. Breetzke and T. Thomas</td>
<td></td>
</tr>
<tr>
<td>Theoretical Computer Science: What is it all about, and is it of any relevance to us?</td>
<td>75</td>
</tr>
<tr>
<td>C. Brink</td>
<td></td>
</tr>
<tr>
<td>Representing Quadrics on a Computer</td>
<td>76</td>
</tr>
<tr>
<td>M.A. Coetzee and M.L. Baart</td>
<td></td>
</tr>
</tbody>
</table>
The Generation of Pre-Interpretations for Detecting Unsolvable Planning Problems 77
D.A. de Waal, M. Denecker, M. Bruynooghe and M. Thielscher

The Emerging Role of the Chief Information Officer in South Africa 87
B. Dekenah

A Java-Implemented Remote Respiratory Disease Diagnosis System on a High Bandwidth Network 88
A. Foster

Early Results of a Comparative Evaluation of ISO 9001 and ISO/IEC 15504 Assessment Methods Applied to a Software Project 89
C. Gee and A.J. Walker

A Neural Network Model of a Fluidised Bed 99
M. Hajek

The Effects of Virtual Banking on the South African Banking Industry 100
M.L. Hart and M. Dunley-Owen

Linear Response Surface Analysis and Some Applications 118
J.M. Hattingh

Model Checking Software with Symbolic Trajectory Evaluation 120
A. Hazelhurst

A Risk Model to Allocate Resources to Different Computerized Systems 137
H.A. Kruger and J.M. Hattingh

Returns on the Stock Exchange 144
J.W. Kruger

Cardinality Constrained 0-1 Knapsack Problems 150
M.F. Kruger, J.M. Hattingh and T. Steyn

An Investigation in Software Process Improvement in the Software Development of a large Electricity Utility 151
M. Lang and A.J. Walker

Design and Implementation of a C++ Package for Two-Dimensional Numerical Integration 162
D.P. Laurie, L Pluym and Ronald Cools

Algebraic Factorization of Integers Using BDE's 169
H. Messerschmidt and J. Robertson
Global Optimization of Routes after the Process of Recovery
M. Mphahlele and J. Roos

Using a Lattice to Enhance Adaptation Guided Retrieval in Example Based Machine Translation
G.D. Oosthuizen and S.L. Serutla

Information Systems Development and Multi Criteria Decision Making / Systems Thinking
D. Petkov, O. Petkova

The Development of a Tutoring System to Assist Students to Develop Answering Techniques
N. Pillay

Combining Rule-Based Artificial Intelligence with Geographic Information Systems to Plan the Physical Layer of Wireless Networks in Greenfield Areas
K. Prag, P. Premjeeth and K. Sandrasegaran

A Distributed Approach to the Scheduling Problem
V. Ram and P. Warren

More readings than I thought: Quantifier Interaction in Analyzing the Temporal Structure of Repeated Eventualities
S. Rock

Ray Guarding Configuration of Adjacent Rectangles
I. Sanders, D. Lubinsky and M. Sears

Developing Soft Skills in Computer Students
C. Schröder, T. Thomas

Information Security Awareness, a Must for Every Organization
M. Thomson and R. von Solms

Pla Va: A Lightweight Persistent Java Virtual Machine
S. Tjasink and S. Berman

Beliefs on Resource-Bounded Agent
E. Viljoen

Object-Orientated Business Modelling and Re-engineering
M. Watzenboeck
On Indexing in Case Based Reasoning Applied to Pre-Transportation Decision Making for Hazardous Waste Handling
K.L. Wortmann, D. Petkov and E Senior

Author Index
<table>
<thead>
<tr>
<th>Name</th>
<th>Institution and Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>S.A. Ajila</td>
<td>Department of Mathematics and Computer Science, National University of Lesotho, Roma, 180 Lesotho</td>
</tr>
<tr>
<td>L. Baart</td>
<td>Department of Mathematics, Vaal Triangle Campus of the PU for CHE, PO Box 1174, Vanderbijlpark, 1900</td>
</tr>
<tr>
<td>L. Barnard</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>S. Berman</td>
<td>University of Cape Town, Rondebosch, 7701</td>
</tr>
<tr>
<td>L. Bester</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>J.M. Bishop</td>
<td>Computer Science Department, University of Pretoria, Pretoria, 0002</td>
</tr>
<tr>
<td>L. Botha</td>
<td>Computer Science Department, University of Pretoria, Pretoria, 0002</td>
</tr>
<tr>
<td>R.A. Botha</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>B. Braude</td>
<td>Software Engineering Applications Laboratory, Electrical Engineering, University of the Witwatersrand, Private Bag 3, Wits, 2050</td>
</tr>
<tr>
<td>T. Breetzke</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>C. Brink</td>
<td>University of Cape Town, Rondebosch, 7700</td>
</tr>
<tr>
<td>M. Bruynooghe</td>
<td>Departement Computerwetenschappen, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium</td>
</tr>
<tr>
<td>S. Buffler</td>
<td>University of Capetown, Rondebosch, 7701</td>
</tr>
<tr>
<td>M.A. Coetzee</td>
<td>Department of Mathematics, PU for CHE, Private Bag X6001, Potchefstroom, 2531</td>
</tr>
<tr>
<td>R. Cools</td>
<td>Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium</td>
</tr>
<tr>
<td>E. de Preez</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>D.A. De Waal</td>
<td>Department of Computer Science and Information Systems, PU for CHE, Private Bag X6001, Potchefstroom, 2531</td>
</tr>
<tr>
<td>B. Dekenhah</td>
<td>The Board of Executors</td>
</tr>
<tr>
<td>M. Denecker</td>
<td>Departement Computerwetenschappen, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium</td>
</tr>
<tr>
<td>M. Dunley-Owen</td>
<td>Department of Information Systems, University of Cape Town, Rondebosch, 7700</td>
</tr>
<tr>
<td>R. Fiqueira</td>
<td>University of Cape Town, Rondebosch, 7701</td>
</tr>
<tr>
<td>A. Foster</td>
<td>Department of Computer Science, University of Cape Town, Rondebosch, 7701</td>
</tr>
<tr>
<td>C. Gee</td>
<td>Software Engineering Applications Laboratory, Electrical Engineering, University of the Witwatersrand, Private Bag 3, Wits 2050</td>
</tr>
</tbody>
</table>
K. Sandrasegaran
Department of electrical Engineering
University of Durban-Westville
Private Bag X54001
Durban, 4000

C. Schoder
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

M. Sears
Department of Mathematics
University of the Witwatersrand
Private Bag 3
Wits, 2050

E. Senior
International Center for Waste Technology
University of Natal, Pietermaritzburg
Private Bag X01
Scotsville, 3209

N.B. Serbedzija
GMD FIRST
Rudower Chausee 5
D-12489 Berlin
Germany

S.L. Serutla
Department of Computer Science
The University of Pretoria
Pretoria, 0002

T. Steyn
PU for CHE
Private Bag X6001
Potchefstroom, 2520

M. Thielscher
Fachgebiet Intellektik, Fachgebiet Informatik
Technische Hochschule Darmstadt
Alexanderstrasse 10
D-64283 Darmstadt
Germany

T. Thomas
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

M. Thom-
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

S. Tjasink
University of Cape Town
Rondebosch, 7700

E. Viljoen
Department of Computer Science and Information Systems
University of South Africa
PO Box 392
Pretoria, 0001

E. Voges
University of Cape Town
Rondebosch, 7701

R. Von Solms
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

A.J. Walker
Software Engineering Applications Laboratory,
Electrical Engineering
University of the Witwatersrand
Private Bag 3
Wits, 2050

P. Warren
Department of Computer Science
University of Natal
Private Bag X01
Scotsville, 3209

M. Watzenboeck
University of Botswana
Private Bag 022
Gaborone
Botswana

K.L. Wortmann
Department of Computer Science
University of Natal, Pietermaritzburg
Private Bag X01
Scotsville, 3209
An investigation in software process improvement in the software development of a large electricity utility

MT Lang *, AJ Walker**

* Eskom Information Technology Department
** Software Engineering Applications Laboratory
Department of Electrical Engineering
University of the Witwatersrand

Abstract

Process improvement techniques as well as metrics have just recently been implemented in the software development department in the Eskom Information Technology Department. The techniques implemented are Function Point Analysis (FPA) and the ISO/IEC 15504 emerging international standard for software process improvement and capability determination (SPICE). Subscription to the Computer Sciences Corporation (CSC) Research and Advisory Services Performance Enhancement Programme (PEP) has facilitated the collection of baselining and measurement data, of which FPA is part. This paper seeks to provide background of these techniques, details of the four projects which were chosen for the comparative case study, and the approach taken for the study. Preliminary results are reviewed.

Keywords: ISO/IEC 15504, function point methods, metrics, software measurements, software quality management, software engineering standards, capability determination, software process assessment.

1 Introduction

There are many techniques for both measurement and process improvement in the industry - some freely available, while others are proprietary, some de facto while others are recognised by standards bodies. Three such techniques have been implemented in this software development department. This study looks at these techniques in the context of a number of typical projects supporting the core business of electricity supply.

Core business consists of three groups:

a. Generation which deals with the generation of electricity by the power plants,
b. Transmission, which deals with the lines which transport the electricity from the power plants to local areas, and
c. Distribution which deals with the supply of electricity to individuals and organisations, as well as the related customer care process.

Non-core business is services which support the core business such as finances and human resources.

Human and technological factors, which are considered outside the scope of process are excluded.

2 Historical Context

In this section, some background is provided on the techniques available culminating with those used by the department.

2.1 Metrics

Function points originated in 1979 when Allan Albrecht [1], then working for IBM was approached to develop a method to monitor and estimate productivity independent of the programming language and technology used. Albrecht felt that size was the primary driver of a project, and centred the development of his measure around software size.

The basis of any software system is that it has a number of inputs, some processing logic, a number of outputs and accesses some files. User perspective dictates the approach to the count and the "size" of a function or file is based on standard values, which were originally statistically calculated. These concepts are demonstrated in a function point context in Figure 1.
Software Process Improvement in a Large Electricity Utility

As specified by Garnus and Herron [3], function points have also been used to:

a. Measure staff productivity
b. Estimate project effort and schedule
c. Measure productivity against other organisations
d. Measure support requirements.
e. Monitor outsourcing agreements
f. Drive IS related business decisions
g. Normalise other measures.

Function points have some problems:

a. they are perceived to apply to more mainframe type applications.
b. they tend to favour applications which are not overly technically complex.
c. they are easier to calculate based on a newly developed system, or maintenance counts based on already existing function point information.
d. they do not appear to cover real-time process control, mathematical optimisation, embedded systems and scientific systems.

In order to address some of these issues, a number of alternative proprietary methods have been derived such as feature points, MKII function points and 3D function points. Workarounds and adaptations to the interpretation of the function point methodology have been developed and democratically accepted by members of an organisation called the International Function Point User's Group(IFPUG).

Although lines of code has been a controversial area due to its dependence on language and programmer, some companies have used this as a productivity metric, taking into account the above influencing factors when using it for estimations.

Using information provided on project size, estimate and actual work effort and cost, defects and errors, platforms, technologies, packages, methods, tools and techniques, staff and user experience and project life cycle or methodology, CSC have developed a method of comparing projects within an organisation. The output of this is a “footprint” (see Figure 2) which isolates and statistically measures every part of the system development lifecycle. Each spoke of the “footprint” represents a measure with the PEP baseline displaying the normalised results of those activities in the rest of the world, and the project measure the relation to the baseline. When the project line is outside the PEP baseline for a particular measure, it implies that it is better on average. [16] A workshop following this assessment by CSC and facilitated by South African PEP representative, Quantimetrics produces a report with suggestions for improvements.

As described in IT Review [14], the descriptions of the measures displayed in Figure 2 are as follows:

a. Function Delivery Index - Overall efficiency of project team which is a measure of the relationship between project size, time and effort.

b. Staffing Index - Whether a project appears to be overstaffed, which is often an indicator of poor planning or unrealistic deadlines. This appears to have a strong correlation to Brook’s Mythical Man
Month [2], which states that adding people to an already late project makes it even later.

c. Function Points/staff month - How much is delivered for resource invested. This incorporates size and effort.

d. Function Points/month - Speed of delivery.

e. Technical Quality Index - Errors detected during testing.

f. Mean Time To Failure - Average time between failures once a system is released into production.

g. Schedule Conformance - On-time delivery.

h. Budget Conformance - Delivery of project within budget.

An important point to note is that the footprint may behave "like a rubber band" i.e. increasing speed may decrease quality.

The footprint provides management with helpful statistics for future planning, resource allocation and quality improvement, particularly as baselining should provide an impetus for performance improvement initiatives, and gives supervisors a tool for effective management. It provides a language which is more suited to discussions with business. [14]

Although there are general metrics which may be captured by most companies, there are some very company specific measures which should ultimately be based on company goals.

Benchmarking is a very risky area. If it is not implemented in the correct manner, it can provide incorrect information and be fairly damaging both to a company's reputation and to employee motivation. Conversely, it can play a role in identifying key areas for improvement in relation to other industry players. It is important to compare apples with apples. It is also important that the benchmarking company provide clear definitions in terms of inclusions of effort, scope and life cycle stages to enable accurate and useful comparison.

2.2 Software Process Improvement

In June 1991, the fourth plenary meeting of ISO/IEC JTC1/SC7 approved a study period to investigate the needs and requirements for a standard for software process assessment. One of the conclusions reached was that there was international consensus on the needs and requirements for a standard for software process assessment.

The standard will provide a structured approach for the assessment of software processes for the following purposes:

a. by or on behalf of an organisation with the objective of understanding the state of its own processes for process improvement;

b. by or on behalf of an organisation with the objective of determining the suitability of its own processes for a particular requirement or class of requirements;

c. by or on behalf of one organisation with the objective of determining the suitability of another organisation's processes for a particular contract or class of contracts.

The SC7/WG10 Workgroup completed its task of producing the set of working drafts in June 1995. These working drafts (Rev 1.0) have formed the basis for a Technical Report Type 2, which has recently become identified as the ISO/IEC 15504 Standard for Software Process Assessment.

2.3 Process assessment, process improvement and capability determination

The model adopted by ISO/IEC 15504 is shown in Figure 3 [8]. Fundamental to the model is the concept of a process, which in Part 9 [13] is described as a set of interrelated activities, which transform inputs into outputs.

Processes are examined using the mechanism of process assessment, defined as a disciplined evaluation of an organisation's software processes against a model compatible with the reference model described Part 2 of ISO/IEC 15504. The output of the assessment exercise is a profile, can be used variously for process improvement (Action taken to change an organisation's processes so that they meet the organisation's business needs and achieve its business goals more effectively) and/or for determination of process capability - a systematic assessment and analysis of selected software processes within an organisation against a target capability, carried out with the aim of identifying the strengths, weaknesses and risks associated with deploying the processes to meet a particular specified requirement.
Clearly, the relationships presented in this model are sufficiently generic to be widely useful outside the field of software.

2.4 The ISO/IEC 15504 architecture

The Standard comprises a set of nine documents. The two key components of the standard are a process model (Part 2) [9] and guidance on conducting assessments (Part 3) [10]. The process model against which an organisation is assessed includes a top-level normative reference model (Part 2) as well as an informative embedded model (Part 5) [11] which contains lower-level detail.

The process model is made up of a framework of attributes in two dimensions: The Process dimension contains process categories, processes and base practices, and the Capability dimension contains capability levels, process attributes and management practices. Input and output work products are associated to products to be used as indicators.

The normative part of the guidance to conducting assessments (Part 3) includes a set of requirements for the collection of data. The informative part to conducting assessments (Part 4) [11] allows for other existing compatible methodologies to be implemented, providing that a clear mapping exists between the attributes of that methodology to the reference model (Part 2).

2.5 The ISO/IEC 15504 Process categories

The process categories described in ISO/IEC 15504 Part 2 are sufficiently generic to find wide application outside the software domain. The process categories are listed in concept Table 1.

<table>
<thead>
<tr>
<th>Process category</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer-Supplier</td>
<td>Processes that directly impact the customer, support development and transition of the product to the customer, and provide for its correct operation and use.</td>
</tr>
<tr>
<td>Engineering</td>
<td>Processes that directly specify, implement, or maintain a system and product and its user documentation.</td>
</tr>
<tr>
<td>Support</td>
<td>Processes that may be employed by any of the other processes (including other supporting processes) at various points in the product development life cycle.</td>
</tr>
<tr>
<td>Management</td>
<td>Processes that contain practices of a generic nature which may be used by anyone who manages any sort of project.</td>
</tr>
<tr>
<td>Organisation</td>
<td>Processes that establish the business goals of the organisation and develop process, product, and resource assets which, when used by the projects in the organisation, will help the organisation achieve its business goals.</td>
</tr>
</tbody>
</table>

While ISO/IEC 15504 has software specific references in all five categories, references to ‘software’ can be comfortably replaced by the term ‘product’ - allowing the model to be used widely within an enterprise.

If a non-software company wishes to use the ISO/IEC 15504 model for process assessment, the embedded model (Part 5) will have to be adapted for the process domain of interest.

2.6 The ISO/IEC 15504 Capability levels

As indicated in Part 2, a capability level is a set of process and management attribute(s) that work together to provide a major enhancement in the capability of a Supplier to perform a process.

Each level provides a major enhancement of capability in the performance of a process. The levels constitute a rational way of progressing through improvement of the capability of any process.

There are six capability levels in the reference model:

- **Level 0 Incomplete:** There is general failure to attain the purpose of the process. There are no easily
identifiable work products or outputs of the process.

**Level 1 Performed:** The purpose of the process is generally achieved. There may not be evidence of rigorous project planning and tracking. Individuals within the organisation recognise that an action should be performed, and there is general agreement that this action is performed as and when required. There are identifiable work products for the process, and these provide evidence to the achievement of the purpose.

**Level 2 Managed:** The process delivers work products of acceptable quality within defined timescales. Performance according to specified procedures is planned and tracked. Work products conform to specified standards and requirements.

**Level 3 Established:** The process is performed and managed using a defined process based upon good product engineering principles. Individual implementations of the process use approved, tailored versions of standard, documented processes. The resources necessary to establish the process definition are also in place.

**Level 4 Predictable:** The defined process is performed consistently in practice within defined control limits, to achieve its goals. Detailed measures of performance are collected and analysed. This leads to a quantitative understanding of process capability and an improved ability to predict performance. Performance is objectively managed. The quality of work products is quantitatively known.

**Level 5 Optimising:** Performance of the process is optimised to meet current and future business needs, and the process achieves repeatability in meeting its defined business goals. Quantitative process effectiveness and efficiency goals (targets) for performance are established, based on the business goals of the organisation. Continuous process monitoring against these goals is enabled by obtaining quantitative feedback and improvement is achieved by analysis of the results.

The capability level of each process is independently assessed, leading to a collection of profiles which represent the assessment output.

Assessments can be conducted without automated tool support, but experience has shown the amount of data to be captured during an assessment and the need to maintain detailed records for process improvement purposes, leads to a definite need for automated tool support for process assessments.

**Table 1: Description of process categories**

<table>
<thead>
<tr>
<th>Process category</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer-Supplier</td>
<td>Processes that directly impact the customer, support development and transition of the software to the customer, and provide for its correct operation and use.</td>
</tr>
<tr>
<td>Engineering</td>
<td>Processes that directly specify, implement, or maintain a system and software product and its user documentation.</td>
</tr>
<tr>
<td>Support</td>
<td>Processes that may be employed by any of the other processes (including other supporting processes) at various points in the software life cycle.</td>
</tr>
<tr>
<td>Management</td>
<td>Processes that contain practices of a generic nature which may be used by anyone who manages any sort of project in a software life cycle.</td>
</tr>
<tr>
<td>Organization</td>
<td>Processes that establish the business goals of the organization and develop process, product, and resource assets which, when used by the projects in the organization, will help the organization achieve its business goals.</td>
</tr>
</tbody>
</table>

The processes in Figure 1 are enumerated for illustrative purposes, but they are not the same across all the process categories see Table 2 for examples.
Table 2: Processes for Engineering Process Category (ENG) and Management Process Category (MAN) [Error! Reference source not found.]

<table>
<thead>
<tr>
<th>Process</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENG.1</td>
<td>Develop system requirements and design</td>
</tr>
<tr>
<td>ENG.2</td>
<td>Develop software requirements</td>
</tr>
<tr>
<td>ENG.3</td>
<td>Develop software design</td>
</tr>
<tr>
<td>ENG.4</td>
<td>Implement software design</td>
</tr>
<tr>
<td>ENG.5</td>
<td>Integrate and test software</td>
</tr>
<tr>
<td>ENG.6</td>
<td>Integrate and test system</td>
</tr>
<tr>
<td>ENG.7</td>
<td>Maintain system and software</td>
</tr>
<tr>
<td>MAN.1</td>
<td>Manage the project</td>
</tr>
<tr>
<td>MAN.2</td>
<td>Manage quality</td>
</tr>
<tr>
<td>MAN.3</td>
<td>Manage risks</td>
</tr>
<tr>
<td>MAN.4</td>
<td>Manage subcontractors</td>
</tr>
</tbody>
</table>

Each process has a capability rating. This rating scale is consistent across all the processes and process categories. As you progress up the capability maturity scale, you transform from not performing the process (level 0) to solely performing the process (level 1) to managing it at a team-based level (level 2) to using a defined organisation-wide process (level 3) to measuring it, and using it for effective prediction and process control (level 4) and then finally to optimising it (level 5).

Each column in the graph in Figure 4 represents a capability level rating for a particular process.

To establish this rating, an assessor makes a judgement supported by documented objective evidence. This may be based on practice attributes (by interviewing process practitioners and managers) or work product characteristics (by examining documents, plans etc.) provided by the SPICE embedded model.

A first assessment can form part of a baseline and then using the Guidelines for Process Improvement [Error! Reference source not found.], improvements can be prioritised and implemented in accordance with organisational goals.

SPICE is a relatively new standard and is still undergoing development and phase 2 trials.

No studies have been done on the specific relationships of the SPICE processes on each other, although there are many software engineering sources which refer to best practices and provide a general idea of possible influences.

IFPUG feels that [5]: “The levels of maturity are additive. The levels also provide one path for achieving an environment where continual process improvement is achievable. Measurement is an integral component of the CMM, starting at the Repeatable level.”

One important aspect of higher organizational maturity is the increase in the level of detail, or granularity, of information. At low levels of maturity, information requirements are relatively low and the objects being measured are large, such as project phase. As the organization matures, the detail of information required becomes finer, such as activity, task, or deliverable.”

2.7 Departmental Process Improvement Initiatives

The metrics programme was initiated 3 years ago by a line manager and Quantimetrics. A number of systems analysts and developers were sent on function point training. The project was originally lead by a project manager with the trained counters providing information. He also performed the data collection for the PEP Baselining programme. Despite the fact that the Counting Practices Manual was a standard, there were a number of discrepancies and problems due to differences in interpretation of standard definitions. A metrics advisor was appointed to standardise the practice and take over from the project manager.

The original programme was difficult, as it was based on needs from the metrics advisor rather than needs from someone within the department. Slowly, with the move from charging on a time and materials basis to charging on a fixed cost basis, some project managers came to realise the benefits of using function points for estimating and cost justification. They proactively trained and used the technique, gathering their own history base which was then used in future estimations. PEP is now in it's third year - there does not appear to be any direct actions as a result of the workshops that have taken place. These benefits were not yet realised at the time of the projects chosen for the case study.

ISO 9000 was felt to be inappropriate for the department's needs due to the controversy surrounding the fact that an ISO 9000 certified company may not necessarily provide either good quality product or good process. The departmental manager elected that we participate in the SPICE trials with the hope of establishing whether it would be usable by the department. The trials are still in progress and the case studies below are the major contributions for the trials.

Siacsit '97

156
3 Selection of Case Studies

Projects were selected by line managers for submission to the PEP cycle and data collection and function point counts were done according to the schedule provided by Quantimetrics. Reviews were performed on the data submitted to CSC. After assessment of results and formulating the PEP baseline, a workshop including managers and software engineers was held to present results and discuss possible courses of action.

It was then decided to select three projects from within the participating projects for the corresponding SPICE assessments. This would ensure that a function point count for the project existed, as well as metrics and benchmarking data which would have been quality assured externally through CSC's review process.

The three projects were required to be IT systems which supported core business.

A number of projects were shortlisted, and some were rejected when problems with resource availability were discovered. When interviews for SPICE began, one of the project teams had had two projects as part of the PEP Baselining and requested that their second project be included as well. The SPICE assessments were performed and are in the process of being reviewed and the SPICE results presented.

4 Presentation of Results

The tables and figures below provide the outputs of the various techniques.

| Table 3: Function point counts as specified in [18] |
|---|---|---|
| Project | Description | Analysis Count | Implementation Count |
| A | New standalone development on predominantly 4GL platform | 146 | 172 |
| B | New standalone development on predominantly 4GL platform | None, since requirements were gathered through prototyping | 307 |
| C | New integrated development on 4GL platform | None, since metrics were gathered after implementation | 89 |
| D | New integrated development on 4GL platform | None, since metrics were gathered after implementation | 128 |

The project functional sizes are relatively small.

Refer to section 2.1 for descriptions of the elements of the PEP footprints.

![Figure 5: PEP Footprint of Project A courtesy of Quantimetrics [17]](image-url)
every attempt has been made to ensure that data collected is accurate, high values may be an indication that the physical data collection is based on recollection rather than accurate records being logged and maintained. The footprints above show that productivity signified by the function delivery index is fairly high. The error data submitted indicated ineffective systems testing. Project A had a fixed cost contract, while projects C and D had a heavy time constraint with milestones in the Christmas season. Project B had no real constraints.

Refer to section 2.2 on descriptions of the elements of the graphs below.

Figure 6: PEP Footprint of Project B courtesy of Quantimetrics [17]

Figure 7: PEP Footprint of Project C courtesy of Quantimetrics [17]

Figure 8: PEP Footprint of Project D courtesy of Quantimetrics [17]

Missing points indicate that information submitted was insufficient to generate the required information. While
5 Analysis and Significance

No analysis has been performed on the results as yet, but early hypotheses are:

a. No obvious correlation between functional size and processes, although a system delivering no business value, which then could be size zero, should have a level 0 Customer-Supplier process.

b. No clear correlation between footprint and capability level or process characteristics. It may be true that:

i. at a higher capability level, your function delivery rate would increase, thereby decreasing cost, schedule, budget, and making your footprint appear better.

ii. having CUS, SUP, ENG.1, ENG.2, ENG.5 and ENG6 processes at a higher capability level would improve the technical quality index and decrease MTTF, thereby making the footprint appear better.

iii. once the relationship between the footprint performance indicators is stabilised, it may then be possible to "stretch the rubber band" evenly across all performance indicators. This may relate directly to an organisation's capability level.

Partnering of the techniques may necessitate the following:

a. PEP providing the facility to factor out the benchmarking results according to software process capability maturity to enable comparison within league or level.

b. alignment of the metrics programme with both organisational goals and capability level to allow tracking of achievement of goals. [5]

c. definition of a number of facilitating base practices for measurement.

Used correctly, all these techniques should work together in assisting with the development of a balance of better quality, faster and cheaper software. It is not the data collection (independent of technique) which should be the focus, but rather the analysis and ultimate application of the results.

6 Discussion

As stated by Pfleeger and Rombach[15], the two techniques should be able to complement each other, as "measurement helps you draw an objective process model", while "application of improvement techniques improves your ability to measure quality", by confirming improvements in process and product, albeit in small increments. Initial results display a high level relationship on the basis of the capability level of an organisational unit.

7 Future work

Analysis still to be done in this study:

a. insight provided by the footprint performance indicators into the capability level of an organisational unit.

b. insight provided by the footprint performance indicators into the capability level of a process.

c. insight provided by the footprint performance indicators into the capability level of a process category.

d. insight provided by the capability level of an organisational unit into the footprint performance indicators.

e. insight provided by the capability level of a process into the footprint performance indicators.

f. insight provided by the capability level of a process category into the footprint performance indicators.

g. limitations of SPICE covered by PEP.

h. limitations of PEP covered by SPICE.

i. limitations of both PEP and SPICE.

Extensions of this research may include variations on:

a. organisational units.

b. capability levels.

c. use of capability determination rather than process improvement.

d. industry sectors.

e. benchmarking organisations.

f. metrics.

8 Acknowledgements

Thanks to all the people who participated in the assessments, as well as the reviewers of the paper,
Software Process Improvement in a Large Electricity Utility

IFPUG, Quantimetrics and CSC for giving permission to publish proprietary information.

References


[2] Brooks F.P., Mythical Man Month, Publisher, City of Publication

[3] Garmus D. and Herron D., Measuring the Software Process, Publisher, City of Publication


[16] Quantimetrics, How we shape up, Information Technology Review, Vol , Issue #, Pg #s Computer Society of South Africa, Month, 1997

[17] SEAL, MLA Function Point Counts for projects A, B, C and D


9 Author contact details

Melanie Lang has obtained 6 out of 8 distinctions in Matric, a B.Sc. (Computer Science and Mathematics) and a B.Sc.Hons (Computer Science) culminating in a research project on Human Computer Interaction. She is currently doing research in software process improvement for a M.Sc Eng (Electrical). She is a member of the CSSA and SASQ.

She is an Analyst-Programmer and part of the Metrics and Quality team of the Eskom Information Technology Department. Projects have included SQL Windows, Oracle, C++, Robohelp, Pascal and Visual Basic et al development for systems for metering and billing; algorithm animation; a Distribution call-in centre; a data model view tool; a technology simulator; Transmission energy trading and a Distribution application interface to Groupwise.

Other work has been in customer support; journal publication; test and change control procedures training; function point counting; metrics and ISO/IEC 15504 (SPICE) assessments, establishing configuration management procedures, design of user interface standards; training material and project management templates.

Contact details: M Lang, ESKOM Information Technology Department, ESKOM, P O Box 1091, Johannesburg, 2001, South Africa. Internet E-mail: langmt@mwp.eskom.co.za.

Alastair Walker is an associate professor in the Department of Electrical Engineering, University of the Witwatersrand. He was responsible for establishing the Software Engineering Applications Laboratory in 1988. The SEAL received an ISO 9001 certification for software development in
July 1995. He is a Certified Quality Analyst, and a Certified Software Quality Systems Auditor.

Contact details: AJ Walker, Software Engineering Applications Laboratory, Department of Electrical Engineering, University of the Witwatersrand, Private Bag 3, P O Wits, 2050 Johannesburg, South Africa. Office phone: +27-11-716-5379, Fax: +27-11-403-1929, Mobile: +27-82-452-0933, Internet E-mail: walker@odie.ee.wits.ac.za; WWW: http://seal.ee.wits.ac.za.