Software Quality Management XXV

Achieving Software Quality in Development and in Use

Editors:

P Marchbank
Southampton Solent University
M Ross
Southampton Solent University & BCS Quality SG
G Staples
BCS Quality SG Chairman
Twenty Fifth International Conference 
on 
Achieving Software Quality 
in 
Development and in Use 

SQM 2017

CONFERENCE CHAIRS
P Marchbank, G Staples

CONFERENCE DIRECTOR
M Ross

INTERNATIONAL ADVISORY COMMITTEE
G Abeyesinghe (Sri Lanka) M Ahonen (Finland)
S Ahonen (Finland) E Bacon (UK)
S Barikzai (UK) E Berki (Finland)
M Biro (Austria) P Burgess (USA)
R Dawson (UK) FJ Dominguez Mayo (Spain)
D Edgar-Nevill (UK) V Edgar-Nevill (UK)
I Fernandez (Spain) R Gevorgyan (Armenia)
E Georgiagou (UK) T Jackson (UK)
P Kampylis (Spain) M King (UK)
P Lineca (UK) L Li (China)
R Lock (UK) C Long (UK)
J Markkula (Spain) F Nilsson (Norway)
P Oriogun (Nigeria) N Paltalidis (UK)
A Savva (Cyprus) M Sheriff (Sierra Leone)
K Siakas (Greece) I Stamelos (Greece)
J Valtanen (Finland) J van Vuren (Finland)
PREFACE

This volume contains the edited proceedings of the twenty fifth international conference on Quality Management held at Southampton Solent University organised by the Quality Specialist Group of the BCS, The Chartered Institute for IT.

The objective of this series of annual conferences is to promote international co-operation among those concerned with software quality and process improvement by creating a greater understanding of software quality issues and by sharing current research and industrial experience.

The papers cover a broad spectrum of practical experience and research. The topic areas include quality standards, quality management, security issues, and systems engineering.

We would like to thank the many people who have brought this international conference into being: the Organising Committee, the International Advisory Committee, particularly for all their hard work in reviewing both the abstracts and the final papers, and the committee members of the BCS's Quality Specialist Group.

The organisers would like to thank Southampton Solent University for their sponsorship.

The Editors
CONTENTS

Keynote
Is it Time to Rethink Project Success?
Darren Dalcher (University of Hertfordshire) 13

Session 1: Software Quality Issues
Using Process Mapping Software to Redesign a Management System
Sue Turner (Consultant, UK) 27

Determining the Quality of Mathematical Software
Using Reference Data Sets
Keith J. Lines, Ian M. Smith (National Physical Laboratory, UK) 39

A Methodological Proposal and Tool Support for the HL7 Standards Compliance in the Development of Health Information Systems
A. Martínez-García, C.L. Parra-Calderón (Fundación Pública Andaluza para la Gestión de la Investigación en Salud de Sevilla, Virgen del Rocio University Hospital, Seville, Spain), M. A. Olivero, A. Suárez-Bote, J.M. Sánchez-Bejines, F.J. Domínguez-Mayo, M.J. Escalona, M. Mejías (Web Engineering and Early Testing Group, Computer Languages and Systems Department, University of Seville, Spain) 53

Products and Prototypes: What’s the difference?
John Estdale (IT Architecture Specialists Ltd, UK) 65

CoFeD: A Visualisation Framework for Comparative Quality Evaluation
Elli Georgiadou, Anthony White, Richard Comley (Middlesex University, UK) 77
Session 2: Risk Issues
Evaluating the Impact of Networking Security Issues on Business Productivity and Efficiency
Nickos Paltalidis (Queen Mary, University of London, UK), Alexandru Cristina (National Express, UK)

A Socio-Cognitive and Computational Model for Decision Making and User Modelling in Social Phishing
Sunil Chaudhary (Faculty of Natural Sciences, University of Tampere, Finland), Eleni Berki (Faculty of Natural Sciences, University of Tampere, Finland) and (Department of Computer Science and Information Systems, University of Jyväskylä, Finland), Linfeng Li (Beijing Institute of Petrochemical Technology, Information Engineering College, China), Juri Valtanen (School of Education, University of Tampere, Finland), Marko Helenius (Department of Pervasive Computing, Tampere University of Technology, Finland)

Session 3: Quality Issues
The Challenges Facing Road-Side ‘m-Preneurs’ in Leap-Frogging the Constraints of ICTs in DCs: A Nigerian Case Study
Dili Ojukwu, (Federal University Ndufu-Alike Ikwo, Abakiliki, Ebonyi State, Nigeria)

Gamified Budgeting for Managing Household Finances
Michael Sievenpiper, Prins Butt (Southampton Solent University, UK)
Proposing a Weighting Function for Adjusting the Global Information Technology Report Networked Readiness Index Framework 147
Peter K. Oriogun (Lead City University, Ibadan, Oyo State, Nigeria)

Software Systems Engineering: A Journey to Contemporary Agile and Beyond, Do People Matter? 159
Raid AlQaisi, Eddie Gray, Bonnie Steves (Glasgow Caledonian University, UK)

A Comparative Study of Cloud Services Use by Prospective IT Professionals in Five Countries 175
Chetan Sharma Kandel, Eleni Berki, Yan Zhao, Sunil Chaudhary (Faculty of Natural Sciences, University of Tampere, Finland), Margaret Ross (School of Media, Art and Technology, Southampton Solent University, UK), Geoff Staples (BCS Quality SG, UK)
Keynote
Is it Time to Rethink Project Success?

Darren Dalcher

Hertfordshire Business School, University of Hertfordshire
De Havilland Campus, Hatfield, Hertfordshire, AL10 9AB
d.dalcher2@herts.ac.uk

Abstract

The notion of success and failure in software projects are confusing. Failure is often considered in the context of the iron triangle as the inability to meet time, cost and performance constraints. Yet, while there is a consensus around the prevalence of project failure, new projects seem destined to repeat past mistakes. This paper tries to advance the discussion by offering a new perspective for reasoning about the meaning of success and the different types of software project failures. The paper advocates rising beyond the fixation with internal parameters of efficiency. It begins by discussing the limited insights from existing project failure surveys, before offering a four level model addressing the essence of successful delivery and operation in software projects and considering the different measures required in order to utilise richer measurements of success.

Keywords: project success, project failure, output, outcome, software project management.

1.0 The Extent of Software Project Failures

The popular computing literature is awash with stories of software development failures and their adverse impacts on individuals, organisations, and societal infrastructure. Indeed, contemporary software development practice is regularly characterised by runaway projects, late delivery, exceeded budgets, reduced functionality, and questionable quality that often translate into cancellations, reduced scope, and significant re-work cycles [1]. The net result is an accumulation of waste typically measured in financial terms. For example, in 1995 failed US projects cost $81 billion, with an additional $59 billion of overspend, totalling $140 billion [2]. Capers Jones contended that the average US cancelled project was a year late, having consumed 200% of its expected budget at the point of cancellation [3]. In 1996, failed projects alone totalled an estimated $100 billion
In 1998, 28% of projects failed, at a cost of $75 billion, while in 2000, 65,000 US projects were reported to be failing [2]. McManus and Wood-Harper [5] reported that the cost of software project failure across the European Union in 2004 was €142 billion. More recently, a McKinsey-Oxford survey of more than 5,400 software projects revealed that half of all projects significantly fail on budgetary assessment, while 17 per cent of projects actually threaten the very existence of the company, with the average project running 45 per cent over budget and seven per cent behind schedule, while delivering 56 per cent less functionality than predicted [6]. According to the report achieving $15 million in benefits now requires an average spending in excess of $59 million.

Consultancies and polling organisations have attempted to collect market data about the prevalence of failure. The Standish Group, for example, has been compiling an annual failure survey since 1994. In 1995, 31.1% of US software projects were cancelled, while 52.7% were completed late, over budget (cost 189% of their original budget), and lacked essential functionality [2]. Only 16.2% of projects were completed on time and within budget; only 9% were in larger companies, where completed projects had an average of 42% of desired functionality (ibid.). The 1996 cancellation figure rose to 40% (ibid.) before improving to around 15% in 2002 (see Figure 1). However, the most recent figures reveal that the current failure rate is 21% [7] with 63% of overall projects labelled as not successful.

![Figure 1 Standish Figures 1994-2012](image)

While the research approach used by the Standish Group has been challenged over the methodology adopted and its rigour [8, 9, 10, 11], the figures provide a well-referenced baseline related to the extent of software project failures. Other studies appear to confirm the high failure rates. For example, Taylor [12] reported that
only 130 projects out of 1,027 were considered successful, while a 2004 PriceWaterhouseCoopers study surveyed 10,640 projects and revealed that only 2.5% of companies achieve budget, scope and schedule targets on all their projects. Sauer and Cuthbertson [13] reported that 16% of IT projects (with a major emphasis on software development) were considered successful, however Sauer, Gemino and Reich [14] noted that 67% of the projects were nonetheless delivered close to budget, schedule and scope expectations. More recently, McManus and Wood-Harper [5] discovered that only one in eight IT projects can be considered truly successful, with almost a quarter (23.8%) cancelled due to issues related to requirements, change, communication, business process alignment and overspend. Using similar definitions IBM [15] reported that only 40% of projects experienced by 1,500 change management executives met their schedule, budget and quality targets, while KPMG [16] observed that 70% of surveyed organisations in New Zealand had experienced a failure in the previous twelve months. Following interviews with 600 developers, Geneca [17] reported that 75% of project participants lacked confidence in project success, admitting that their projects are ‘doomed right from the start’.

Jones [18, 19] investigated the likelihood that the average US software project will be cancelled, typically due to cost and schedule overruns, failure to meet requirements, poor planning, estimating, quality control or excessive requirements creep, relative to size. The results indicate that none of the eight domains investigated are fully successful for large systems of above 10,000 function points in size, showing the average probability of cancellation at 36%. He warned that the ‘development of large applications in excess of 10,000 function points is one of the most hazardous and risky undertakings of the modern world’ [18, p. 54]. Applications in the region of 100,000 function points are more likely to fail with an average cancellation likelihood reading of 51%, with some sectors such as Management Information Systems displaying higher failure rates (70%). Jones [19, p. 308] concluded that: ‘Cancellations, major delays in excess of one calendar year, and cost overruns in excess of 100 per cent remain endemic problems for software applications in the 100,000 function point size range, and larger.’ Jones [20] further added that: “large software projects are almost always over budget, usually delivered late, and are filled with bugs when they’re finally delivered. Even worse, as many as 35 per cent of large applications in the 10,000 function point or more size range will be cancelled and never delivered at all.’

Flyvbjerg and Budzier [21] contended that IT projects are now so big and their influence so wide ranging across many aspects of the organisation, that they pose a singular new kind of risk that can sink entire corporations, cities, and even nations. Their global survey of 1,471 IT change projects showed that while the average cost overrun on large initiatives was 27%, one in six projects showed a cost overrun of 200%, on average, and a schedule overrun of almost 70%. As software is integrated into bigger products and systems, the concerns can become magnified.
‘The software industry has the highest failure rate of any so called engineering fields. An occupation that runs late on more than 75 per cent of projects and cancels as many as 35 per cent of larger projects is not a true engineering discipline.’ [20]

2.0 Beyond Simple Success Measures

The relationship between success and failure is not clear. Some view the relationship as a binary function so that a project is either successful, or not. The research by McManus and Wood-Harper [5] describes failure as ‘those projects that do not meet the original time, cost and requirements criteria’. The Standish Group makes a further distinction between ‘failed projects’ and ‘challenged projects’. Failed projects are cancelled before completion, never implemented, or scrapped following installation. Challenged projects are completed and operational projects which are over-budget, late, and with fewer features and functions than initially specified. Successful projects, in contrast, are completed on time, on budget, with all specified features. Figure 1 also shows the relationship between successful, challenged and failed projects. Observing the Standish figures over the past nineteen years, would appear to indicate a rough rule of thumb suggesting a split of 25% of projects being successful, 50% being challenged, and 25% failing.

The Oxford Dictionary defines success as: a favourable outcome; doing what was desired or attempted; the accomplishment of an aim or purpose; or the attainment of wealth or fame or position. Failure is broadly defined as lack of success supporting the idea of a binary relationship. In an attempt to make further sense of the relative positions of success and failure, software surveys have clearly found it useful to introduce the idea of partial failure (or challenged projects) as an intermediate position between success and failure, potentially indicating dissatisfaction with a two state explanation. Indeed many project outcomes do not fall directly into either category.

The majority of the studies mentioned above define success as meeting all the criteria associated with the budget, schedule and performance; with failure viewed as a failure to meet all of the same criteria. This implies that if a project is finished on time, within budget whilst offering the expected performance it can be viewed as successful. Conversely, failing to meet any of the criteria will deem it a failure. The view is predicated on the traditional measures applied in project management and generally known as the triple constraint, the golden triangle or the iron triangle. This idea presupposes high estimation accuracy with regard to the initial formulation of the variables of the triple constraint [11] when the degree of uncertainty is at its greatest.

Traditional project management theory holds that optimising the three criteria will result in ideal performance on a project. Typical projects thus require a balancing act between the so called triple constraints of time, cost and performance as expressed in the original triangle conceived by Dr. Martin Barnes in 1969 (see
figure 2). Note that the third corner is named ‘performance’. The original release named that corner ‘quality’, but this was soon corrected to performance ‘to reflect whatever the finished product was supposed to achieve’ [22]. Performance means satisfactory function of the product, which has to be fully defined. This could be specified in terms of rate of return, profit, beat the enemy, impress visitors, or in the case of software the scope and expected functionality. The whole point of the triangle is that the spot can be placed at such a point that its closeness to each corner represents its relative importance and helps the project manager to make informed decisions about the project. Trade-offs and adjustments are therefore made by restricting, adding to, or adjusting the cost, time and performance associated with a project. The triangle enables managers to consider each decision and its implications on the dimensions of time, cost and performance and integrate the different project management functions. For example, the more that is requested in terms of performance, the more it is likely to cost and the longer the expected duration. If the client needs to have a certain performance delivered very rapidly, this will increase the cost due to the need to work faster and have more resources involved in the development, albeit with increased communication costs. The more features expected from a system, the higher the cost and the longer the expected duration. Conversely, if the costs need to be kept to a minimum, one may need to consider the essential performance, or the overall project scope, and compromise there [23].

![Figure 2 Cost, time and Performance Trade-off](image)

The three factors clearly play a key part in determining the degree to which a project is challenged (or even deemed a failure); yet they may be uncontrollable by the project manager. Indeed, Capers Jones observed that the most common constraints encountered are: fixed delivery dates; fixed-price contracts; staffing or
team size limitations; and performance or throughput constraints [24] i.e. fixed
time, price, staffing level, performance and scope. Many managers are thus looking
to control other factors that may alter the outcome of the project, in particular, as
the constraints often occur in concert. Measuring success on the basis of pre-
established parameters that cannot be adjusted is therefore of limited value.

3.0 Rethinking Project Success

Project success is a rather nebulous concept and the focus on the triple constraint
can be too limiting. Indeed, Linberg [25] asserted that a whole new theory of
project success is needed. Pinto and Slevin [26] noted that success combines issues
related to the project itself with issues related to the client. Moreover, software
developers and systems analysts have recognised long ago that user involvement,
satisfaction and buy-in are crucial to the success of software projects. Prototyping
and user-driven approaches were developed to maximise the potential for
satisfaction for various stakeholders and thus increase the likelihood of user
acceptance of the ultimate system.

Baccarini identified the need to distinguish between project management success
and the success of the product which entails dealing with the effects of the
project’s final delivered product [27], thereby allaying the need to define a further
dimension concerned with client expectations which have already been expressed
in the desired performance functionality. Ironically, this chimes with the original
(but often misunderstood) intention of the Barnes’ triangle (figure 3) to capture the
agreed upon definition of the purpose of the project or how the complete project
would perform. Given that the product will be utilised by the client there is a
degree of correspondence between the dichotomies put forward by Pinto and
Slevin and by Baccarini. Indeed, de Wit [28] observed that measuring progress and
cost are part of project control, which should not be confused with measuring
success. Cooke-Davies likewise made a distinction between the focus on project
performance and the need to look at the success of a project [29].

Having multiple categories of success would suggest that it is possible to be
successful in some areas and not successful in others. It thus makes it possible to
understand mismatches between the different criteria and groups. Moreover, it
implies that the traditional triple constraints of cost, time and performance only
reveal part of the picture. In other words, it may be possible to maximise the
traditional criteria and yet deliver a product that is not valued by the users.
Likewise it is also possible to exceed the traditional criteria but deliver a product
that is valued and adopted by the user community, despite exceeding the budget or
the schedule, or even both.

The discussion thus far indicates that at least two different levels of success can be
identified. Indeed, according to Munns and Bjeirmi [30] it is possible to achieve a
successful project even when management has failed, and also possible to deliver a
failed project following successful management. However most studies and
surveys of software project failures tend to focus on the traditional criteria of efficiency embedded through the triple constraints of time, cost and performance. They thus ignore the deeper aspects associated with the delivered product, its perceived utility and value, the expectations and needs of stakeholders, the intended performance of the product and the project context.

Further evidence of the need to look beyond the traditional criteria is provided through Table 1, which summarises an extended and refined set of common issues that were originally identified across six project failures covered in detail in [31] and extended through a sequence of workshops with practitioners, the mapping of factors in 150 failed projects and a series of four international surveys resulting in the revised figure presented in this paper.

<table>
<thead>
<tr>
<th>Area</th>
<th>Typical additional issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationship management</td>
<td>Vendor-client disagreements, partnerships, long-term perspective, respect, joint working</td>
</tr>
<tr>
<td>Trust</td>
<td>Lack of trust, reliance, co-operation</td>
</tr>
<tr>
<td>Communication</td>
<td>Information, barriers, exchange, ambiguities</td>
</tr>
<tr>
<td>Management of expectations</td>
<td>Stakeholder engagement, needs assessment, involvement</td>
</tr>
<tr>
<td>Politics</td>
<td>Organisational politics, blocks, defensive routines</td>
</tr>
<tr>
<td>Escalation of commitment</td>
<td>Sunken costs, pressure, escalating investment</td>
</tr>
<tr>
<td>Risk management</td>
<td>Exchanging risks, effective transfer</td>
</tr>
<tr>
<td>Contract management</td>
<td>Contractual engagement, multiple interpretations, expected obligations</td>
</tr>
</tbody>
</table>

The obvious message from the set of issues is that the traditional efficiency criteria as embedded in the triple constraint do not appear to have played a part in the build-up to any of the failures. Instead, the issues identified were more concerned with the product (as well as the assumptions and expectations surrounding it) and the overall business success.

4.0 Towards Multiple Levels of Success

Success, it would appear, needs to be understood at multiple levels in order to appreciate the complex dynamics and subtle impacts. A tabular representation of four levels of success, which builds on the earlier discussion, is offered in Table 2.
Table 2 Levels of Success

<table>
<thead>
<tr>
<th>Levels of project success</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1: Project management success</td>
<td>Efficiency and performance</td>
</tr>
<tr>
<td>Level 2: Project success</td>
<td>Objectives, needs, stakeholders</td>
</tr>
<tr>
<td>Level 3: Business success</td>
<td>Benefits, value creation, delivery</td>
</tr>
<tr>
<td>Level 4: Future potential</td>
<td>New markets, skills, opportunities</td>
</tr>
</tbody>
</table>

Level 1 represents project management success and is thus concerned with internal efficiency and performance measurement and optimisation at the project level through the tracking of the cost, schedule and performance parameters. Level 1 success is therefore to do with project delivery against the constraints or measures imposed on the project.

Level 2 is focused on the overall effectiveness of the project through the lens of what is actually being delivered. Success is measured through the utility and acceptability of the output that has been delivered. The achievement of the objectives is thus assessed in terms of the satisfaction of the customer and the different stakeholder groups and the satisfactory addressing of their needs. Level 2 success reflects the acceptability and impact of the resulting artefact, its usefulness, the degree to which it is used, the match with the project objectives, needs and requirements, the relationship with the different stakeholder groups, and the overall impact on the customer.

Level 3 is centred on the business efficiency which is assessed through the realisation of identified benefits of the project and the creation and delivery of internal value. The outcome of the project contributes to business success through the satisfaction of business objectives and the delivery of identified benefits and realised value. Success equates to maximisation of financial and business efficiency measures, such as sales, profits or ROI, as well as realised benefits and delivered value.

Level 4 is forward looking and opportunistic and enhances the business horizon by projecting future gains and opening new avenues, capabilities, skills and markets. Strategic opportunities require a continuous and long-term approach that seeks to derive not just immediate benefit but also maximise opportunities for cornering the market, creating killer applications and building the potential for self-enhancing positive feedback loops to secure future growth. Level 4 success is achieved through the realisation of new opportunities and harnessing of new potential through the application of continuous improvement, growth and further
development. It may include new uses or ideas that were not originally considered as well as the development of new competence or capability.

The focus identified in Table 2 provides a clue as to the nature of measurements required at each level. Measurement at Level 1 focus on determining the progress and efficiency of the project management effort for example through the use of earned value management. Measures for Level 2 are concerned with measuring the achievements of objectives, needs, requirements and expectations. Measures for level 3 emphasise the business value using traditional economic measures such as sales, revenue, and delivered value, as well as applying benefit realisation approaches. Measures for level 4 require more creative measurement of opportunities, capabilities and market position. The combined levels offer a richer way of conceptualising and making sense of the complex phenomena surrounding success in and around projects.

5.0 Measuring Success

Determining the success of a project is not simple. It is often said that success is in the eye of the beholder, and can mean different things to different people. Consequently, analysing the dimensions of success and failure is a complicated task that requires an understanding of the different levels of success and what each one can offer:

**Project management success** implies tracking data related to predicted cost, time and scope. Measuring performance against efficiency considerations is relatively straightforward. Determining progress through monitoring the achievement of milestones (e.g. using Earned Value Methods) enables project managers to track the achievement of pre-defined targets. It is a very useful focus when there is little residual uncertainty or when the project is clearly understood. However it is debateable whether the measurement of an arbitrarily pre-defined target is completely meaningful, especially when project managers play little, if any, part in the initial estimation. Typical measures would focus on the efficiency of the process emphasising milestones, identified defects, and delivery and change management measures (including approved change requests), as well as earned value management measures showing project management progress, cost and schedule variances, cost performance index, and estimate to complete.

**Project success** relates to the effectiveness of a project and is normally considered in terms of the achievement of requirements and needs, and the acceptability of the outputs. In order to provide meaningful values, measures should relate to the requirements identified and be established and acknowledged as part of the needs assessment and requirements management processes. Stakeholder management is central to the identification and assessment of the concerns of different stakeholder groups and the issues impacting the development team. de Wit [28] defined success as encompassing a high level of satisfaction concerning the project result amongst key stakeholders and users, while Lytinen and Hirschheim [32] framed
the effort in terms of meeting stakeholders’ expectations in terms of the balance between the objectives, constraints and benefits. Project success can therefore be viewed as equalling and exceeding the expectations of clients, users and stakeholder groups, thereby emphasising elements of user acceptance, and stakeholder satisfaction and management. Typical project success measures would identify achieved project requirements, satisfaction levels, recorded complaints, usage figures for the delivered artefacts, and met expectations.

**Business success** pertains to the organisational value derived in terms of finance, environmental, and social concerns, and their balancing. The perspective often requires a longer timeframe that considers value creation and delivery over investment cycles and the contributions made towards the achievement of strategic objectives of the organisation defined in the business case. Business success can refer to the payback period, but often extends to consider the accumulated benefits (i.e. realising the stream of benefits allocated to the project as they are cascaded down from the strategic objectives) accruing from an investment in a project or initiative. Business success measures typically address realised benefits, delivered value, rate of return, breakeven calculations, payback calculations, sales achieved, revenue measures, environmental and social targets, and increasingly, may focus on reputation, influence marketing, and sustainability ratings.

**Future potential** extends the time horizon of consideration into the longer-term utilisation of the outcomes and results of projects and actions. It allows the accumulation of longer-term benefits that result from adjustments, improvement and re-balancing. The intention is to seek to increase the accrued value from projects by exploring and exploiting opportunities beyond the agreed business case. Given the long-term focus it cannot be assumed that project assumptions, and intended outcomes retain their relevance over time. The aim therefore is to maximise organisational value in accordance with the evolving strategic direction. When projects are completed under conditions of uncertainty, they are often subject to positive feedback cycles, systems dynamics, and complex interactions that uncover new opportunities and strategic openings. Potential opportunities can often lead organisations to explore new directions, expand into a particular market or occupy a certain leading position within a sector, and adjust their strategic intentions to match their new ambitions. Measures will focus on the identification and utilisation of emerging opportunities and adaptation to new market conditions that result from experiencing, learning, adaptation and strategic re-positioning.

### 6.0 Conclusions

Project failures have been used to highlight the need to improve IT software project practice. Many of the studies and surveys focus on project management success (or failure), which can be described as a subset of internal efficiency measures and imposed constraints ignoring the impact on the project and the business. In order to improve project performance project managers need to look beyond such measures and focus on project success—an area concerned with the effectiveness and relevance of the project output. Project managers are also
increasingly asked to consider the value derived from the project, the sustainability implications as well as issues related to environmental, social, and societal impacts.

Such considerations require a richer and more complex mapping of success. Success is a complex and multi-layered concept that needs to be understood at different levels and timeframes. Indeed, the impact of success often extends beyond a single project. This paper offers a wider perspective, which takes in a range of project success levels thus enabling practitioners to move beyond the simplistic measures that continue to be offered. The success view determines actions and colours new developments. Increased attention to enterprise objectives and utility, rather than simply endeavouring to optimise correctness according to pre-imposed constraints, can open a new dialogue about the needs of a profession seeking to fundamentally and essentially improve its track record and enable project management practice to rise beyond the continuous obsession with failure.

Further work is needed to encourage the research and practice communities to consider project management success at a number of levels. Practitioners will need to make links between strategy, business, and project management delivery functions, while researchers are likely to try and make sense of requirements and expectations that emerge from a multi-level model that invites new types of surveys to make sense of the success and failure in software projects. Ultimately, in order to overcome failure we must learn to appreciate success and grow up enough to look beyond the simplest manifestations of an imperfect practice.

“Success is not final, failure is not fatal: it is the courage to continue that counts.”
– Winston Churchill

7.0 References

8 Glass R (2005) IT Failure Rates—70% or 10-15%. IEEE Sw 22(3): 110-112
22 Barnes M (2013) Private communication, September 2013
Section 1

Software Quality Issues
Using Process Mapping Software to re-design a Management System

Sue Turner

Software Quality/Business Systems Consultant,
Camberley, Surrey
bc@turner-solutions.co.uk

Abstract

Management Systems are becoming *de rigueur* for Organisations, but many with existing Management Systems are finding that they are creaking at the seams. Changes to Standards, Regulations, Business Practices, Organisation structures and Products mean that Organisations have to be flexible and their Management Systems also. With Management Systems based on those written in the 1990s, companies are realising that they need to make a step change in order to maintain their competitive advantage.

This Management Summary will discuss why Process Mapping should be considered as a method for this improvement, what to consider when choosing a Process Mapping tool and how the change should be planned. It uses experience from several companies with which the author has been involved.

**Keywords:** Process Mapping, Management System, Quality

1.0 Introduction

Management Systems need to be continually reviewed to ensure that they keep in step with the changing Standards and Regulations. These may include Statutory requirements (for all countries in which a Company trades), Corporate rules (for those Organisations that are part of larger bodies), Industry-specific regulations (relating to the Industry they are in and the type of Customer they have) and Quality Standards (often required by the Customer). For all of these, Organisations have to demonstrate compliance.
2.0 Why a Re-design might be needed

Many Management Systems will have been written against earlier versions of the Standards and then reviewed and updated at each new version. This can result in a mixture of procedures where new practices have been added without wholly removing the old ones.

Business Practices are changing. Companies are now looking at their Supply Chains to gain competitive advantage and to ensure that only genuine parts are incorporated into products. This means that there is pressure on Suppliers to improve their Management Systems, so that they can provide this traceability and so on down the Chain.

Companies may add to their portfolio of project and product types and need to update their Management Systems to cover additional compliance. Project and product types may be removed from the types of work undertaken and the Management System reviewed to remove unnecessary procedures.

Organisations change. Small companies with their own certified Management Systems and acquired by larger companies, may be subsumed into the larger company's Management Systems. Large Companies which buy these smaller enterprises may need to adjust their Management Systems to include new products and services resulting from these acquisitions. Companies may divest themselves of parts of their business creating a new business in its own right and both parties then need to review their Management Systems. It is likely that the new company was covered by the Certifications of the parent, so will need to create their own Management System, though taking a copy and merely changing the company name will generally not be appropriate.

Staff move on. The people who originally wrote and approved the procedures may have left the company or moved into new jobs. Senior Management may have changed due to acquisition or divestment. There may be no-one now who understands the Management System in its entirety, so changes can have unexpected knock-on effects.

Internal audit and external assessment sometimes find flaws in the Management System where changes to working practices have not kept pace with the changes to the Management System or vice versa. This can mean that new procedures are added to the Management System to satisfy these findings without considering the effect of these new procedures.

Technology is always evolving. At one time, the Management System would be a single, typed document held by the Quality Manager. Now in tech-savvy companies, the Management System is available on the Company Intranet which can include hyperlinks from document to document.
3.0 What are Process Maps?

A process map is a diagram showing what the process is and can be used to minimise written procedures. There are several well-known notations, which are available in many tools, though some software tools offer their own notation.

3.1 Flowcharts

The use of pictures and diagrams has a long history. One of the earliest forms was the Process Chart which was introduced in the 1920s by Frank and Lillian Gilbreth [1], to document process flow. Through its use in Software Programming, the idea evolved into the Flowchart. While there are many symbols available [1], a simplified form is generally used for showing the flow of control. The flow is shown as a series of rectangular boxes containing the process steps, with diamond shaped boxes for questions with Yes/No answers to define the flow (see Figure 1).

3.2 Integrated Computer Aided Manufacturing (ICAM) DEFinition for function modeling (IDEF0)

In the 1970s, the IDEF0 Notation [3] was developed by the US Air Force Program for Integrated Computer Aided Manufacturing (ICAM) as a function model to represent the functions, activities or processes within the modelled system. It includes the Inputs, Outputs, Controls and Resources required (see Figure 2).
IDEF0 also introduced the concept of decomposition, where a single process step could be shown in another map broken down into its lower-level process steps. Each of the outputs of the intermediate process steps form the inputs of another (see Figure 3).

Figure 3: IDEF0 Diagram Example

3.3 Business Process Model and Notification (BPMN)

BPMN [4] was first released in 2004 and provides a graphical notation that can be used to define an Organisation’s processes. Like the flowchart, this has Activity boxes and Gateway diamonds for decisions and has Event circles to show the beginning and end of the process. It also has the capability of including data objects like Data Stores and differentiating the different types of flow, e.g. sequential flow and message flows (see Figure 4).

Figure 4: BPMN Diagram
3.4 Process Navigator

Process Navigator from Triaster [5] is a software tool with its own notation. Its process maps include Activities connected by deliverables (artefacts) that are used by or produced by the activities. All maps begin and end with deliverables i.e. an activity is always triggered by a deliverable and will always produce at least one deliverable. These may be tangible items like documents or products or could be intangible like ‘Monthly’ (triggering a monthly activity) or ‘New Employee added to the Payroll’. Process Navigator also differentiates between external deliverables, that come from or go outside the company, and internal deliverables which must be created and used at least once in the complete set of maps. Since the start and end internal deliverables must exist in more than one map, they are used to link the process maps together. Decision boxes are diamond shaped, but are treated as activities and have at least two deliverables as outputs though they could be a range of values as shown in Figure 5. Responsibilities can be added below the activities and decisions. Any activity or decision can be drilled-down to another process map, which shows the detailed process steps of that activity.

![Figure 5: Process Navigator Process Map Example](image)

4.0 Why use Process Maps?

Process Maps can satisfy the needs of the different groups of people who need access to or who provide the information for the Management System.

4.1 For the Users

Process maps allow the process to be seen at-a-glance. All the notations mentioned use boxes to state the process step or activity in about seven words. This means it is easy to scan the map to gain a high-level understanding of the process or to understand what needs to be done. This can reduce the amount of training required for new starters and for people changing jobs.

A Software tool can provide access to the other documentation and applications associated with the map using hyperlinks and is likely to be accessible on phones and tablets as well as Personal Computers (PCs).
4.2 For the Process Owners
The maps clarify the process and allow the responsibility and accountability for the activities to be defined. This can reduce the likelihood of mistakes being made.

4.3 For Internal Auditors and External Assessors
With the emphasis in ISO 9001 (Quality Management Systems - Requirements) [6] and AS9100 (Quality Management Systems - Requirements for Aviation, Space, and Defense Organizations) [7] for the auditing of processes, the process maps make it easier to understand what the process is and what records should be available.

Where Regulatory Compliance, e.g. Aerospace, Medical, Legal or Safety-Critical, is required, the maps can be used to show the appropriate part of the processes.

5.0 Re-designing the Management System
5.1 Initial Planning
Planning for the change from written procedure-based to process map-based Management Systems needs to be considered carefully. It is a major change and must have the buy-in and on-going support of the Senior Management as while a lot of the work will be done by the Quality Department (or equivalent), Senior staff and Subject Matter Experts from the other Departments will need to be involved. It should also be noted that this is the stage at which Quality Management Systems (primarily used to assure Quality Product) can metamorphose into Business Management Systems covering everyone in the Organisation and making everyone a stakeholder in assuring Product Quality Repeatability.

The initial approach to the Senior Management may be a presentation. It should describe the problems that the current Management System is having and how a new Management System supported by Process Maps will overcome these problems and help the Company to improve. It should also include the qualitative and quantitative benefits to the business of the new system.

The result of the meeting should be an agreement to proceed to the Business Case, probably with a budget limit.

5.2 Choosing the Process Mapping Software
5.2.1 Are there any Corporate Requirements to be considered?
If the Organisation is part of a larger company, the Corporate Systems current or planned have to be considered to ensure that there is no conflict. Large American companies usually have Corporate agreements with American Software vendors and expect all their subsidiaries to comply, sometimes regardless of whether the tool fits with the needs of those subsidiaries. Introducing a new type of tool, e.g. process mapping software, may appear to conflict if they have not understood its purpose.
5.2.2 Can the current Information Technology (IT) Infrastructure support the Tool and any additional software it needs?
The software tool may require a specific infrastructure, for example, a high specification server running Windows or Linux. This may already be available in the Organisation, but if not then the cost of the additional hardware should be a factor in the choice of tool. In addition to the Operating System there may be additional tools that are required. For browser-based applications, there will need to be a webserver and the mapping may require a tool like Microsoft Visio.

5.2.3 Can the Tool support interfaces with other Business Applications?
Many companies already use Enterprise Report Planning (ERP)/Material Requirements Planning (MRP) software systems to support their businesses. It would be useful if the tool could provide direct access into these Systems, e.g. hyperlinks from the maps into appropriate transactions.

5.2.4 What should the Process Maps look like?
Notations like Flowcharts (see section 3.1), IDEF0 (see section 3.2) and BPMN (see section 3.3) appear very technical and might not be easy to understand by everyone. Alternatively, if those notations are required, choose a tool that could convert the model into a more readable form.

If colour is to be used, especially to differentiate between different types of similar shapes, then the effect of colour-blindness [8] should be taken into account.

5.2.5 How will the Process Maps be provided to the Users?
If all staff have access to PCs which are always linked to the server, then a client-server application or browser-based application would be sufficient. If some staff will be using phones or tablets a mobile version of the Management System will be needed. If there are off-site workers who work without access to the server, they will need to have a variant that is standalone.

Accessibility [9] should also be a consideration and include the impact of both visual impairment and physical ability to use the Management System.

5.2.6 How easy is it to create the Process Maps?
The ease of use may affect the amount of resource required and the time taken to complete the mapping. The easier the maps are to create, the larger the pool of potential resources. With more resources able to map, the timescale could be reduced, but this may be balanced by the cost of the resources and the time lost to their usual work. If more skill is required, then fewer resources may be available and the timescale could be longer.

5.2.7 Can associated documentation be accessible from the Process Maps?
While the procedures may have been replaced by the process maps, it is likely that there will still be a need for Work Instructions and Forms and they should be accessible from the appropriate map. Consider also how the high-level Policy
documents like the Business Management Manual (formerly the Quality Manual) will be accessed. It may also be possible to add links to other company applications like SAP and Oracle.

5.2.8 Is there a defined methodology for creating the Process Maps?
The traditional method of mapping processes is to define the top level and then decompose the activities until the lowest level is reached. However, depending upon the complexity of the top level, there may be a varying number of levels across the map set.

The alternative method is to build from the bottom-up. These means that all the ‘working’ level maps are at the same level and could be linked horizontally. The higher level maps can be used to display summaries of the lower-level processes.

5.2.9 Can the Tool provide additional information about the activities?
The Responsibility, Accountability, Consulted and Information (RACI) model can provide the roles and their responsibilities in the processes. Given the visual nature of the process maps, it would be useful if this information could be recorded for each activity. Other information could be the Standard and Regulation clauses to which the activity provides compliance coverage.

5.2.10 What Reports and Metrics are available?
The tool is likely to provide a number of standard reports, but consider whether other reports are required and the ease (or cost) of creating them. If Standard or Regulatory compliance is required, then producing a report from the processes showing the compliance coverage will be useful for External Assessments.

The data from the Software Tool system should also include metrics on the usage of the Management System, for example, the number of people visiting the maps and the frequency of changes to them. It is likely that there will many visitors initially, but trailing off as they become familiar with the maps. There will also be a number of changes at the start, reducing as the maps become more mature. If the usage figures and the number of changes drop below a threshold, the Management System may be failing in its purpose.

5.2.11 Can the Tool provide a history of the changes to the Process Maps?
After the initial introduction of the new Management System, changes will be required. It would useful if the tool could record these changes together with a copy of each version of the maps. It may be necessary to revert to a previous version or to see the system as it was at a specific point in time.

5.2.12 Is there some form of automated approval?
As the changes are requested, there needs to be some form of approval that the change can be incorporated into the Management System to ensure the content is correct and that the compliance coverage is maintained.
5.2.13 What is the software licensing model and the cost?

If the tool is Open Source, check that this use of the tool does not break the terms of the licence, e.g. the GNU General Public License (GPL) [10].

If the tool is commercial, the licensing model needs to be considered. The software is sometimes provided “forever”, with upgrades paid for under the Support Contract or it may be a subscription model and the software is available while the subscription is being paid, after which it will cease to work.

There may be additional costs for setting up the system, for training of the mappers, customisation of the interfaces, e.g. to include the Company logo, and general consultancy.

The support and maintenance contract should include periodic upgrades, troubleshooting problems by email or phone and the facility to request improvements.

It should be noted that with large vendors most of these services would be provided by local consultants with the support run by the vendors themselves. This may mean that if any bugs are identified, there may not be a fix until the next upgrade cycle. Smaller software vendors may provide the consultants themselves and provide fixes to the bugs quite quickly.

5.2.14 Does anyone have experience of Process Mapping Tools?

Actual experience, good or bad, of using the tools or of the vendors is always valuable. Consider, though, how long ago they used it - bug fixes and additional functionality in the intervening years can improve a bad tool.

5.3 Writing the Business Case

The Business Case or Proposal builds on the information provided in the initial presentation but will also include the Approach, the level of Resources, the draft Schedule and the estimated Cost. At this stage, the re-design may be considered as a project.

5.3.1 Approach

The Approach includes whether the re-design is to be provided as a Big Bang or Phased, for example, by department. With the Big Bang approach, everyone’s processes will suddenly change which could be bewildering to the users, whereas with the Phased approach users will become more familiar with the system as they interface with those other departments already mapped. On the other hand, as departments are mapped their processes may affect those of departments already completed and changes have to be made to the latter.

As well as Big Bang versus Phased, the process by which the maps are to be created needs to be defined. Two alternatives are to have a small team of mappers who visit each of the departments, facilitating the capture of the process and
creating the maps or to have mapping teams within each department who capture their own processes and create the maps.

The Approach also need to state how the existing Quality Management System is to be phased out. With Big Bang, it will be here today and gone tomorrow, but with the Phased approach, the two systems will be running in parallel. There needs to be consideration of the effect this will have on the users, on internal auditors and external assessors and on project documentation referencing the old procedures.

The Approach should describe how the project is to be managed. Companies which are used to short projects may use their existing processes; other Companies may be required to use the Corporate procedures intended for multi-million pound projects, adjusted accordingly; small companies may assign a part-time project manager leaving the rest of the team to do the work.

5.3.2 Resources
The resources required for this project will need to include people to map the processes (as defined in the Approach), to provide the additional documentation, to review and approve the maps, experts to confirm that the Standards and Regulatory compliance can be met, librarians to ensure that the rules for the maps and the documentation have been met, IT specialists to maintain the technology and Managers to manage the team. They will all need to be trained to some level.

In addition, once the Management System has been finally released, there will be less to do and the size of the team can be reduced.

5.3.3 Schedule
Whether the final release is Big Bang or Phased, the mapping will have to proceed in stages, the easiest way being by Department. The order in which they would be done will need to be considered. Production schedules should be consulted to avoid taking people from their work to map just when they are needed for a production run. It is also probably not a good idea for the new Management System to be unveiled just before a Standards or Regulatory Compliance assessment.

For the Business Case, a full project plan may not be needed, a simple time-line with milestones may be sufficient.

5.3.4 Costs
The calculations on the cost of the software tool will have been obtained as part of choosing the software. The Resource costs will be related to the Approach and to the availability of staff. The IT Infrastructure cost will be related to any additional hardware, the additional software tools needed and all their associated maintenance costs. Some of this may be covered by Capital expenditure.

The costs for the Business Case may also need to include those for the long-term maintenance, e.g. the Software Tool licence and the Core Team.
5.4 Detailed Planning

Finally, once the go-ahead has been given, the detailed planning can take place.

5.4.1 Resources

While the Business Case stated what type and the number of resources required, it is now time to put in the names of the individuals, at least for the Core Team and to decide the method for obtaining the names of those in the departments who will be involved in the mapping process.

A decision will be needed on who will need training and to what level. Ultimately all the users will need to be trained to use the new Management System and this will also need to be included in the Induction process for new staff or those moving between departments.

5.4.2 Consistency

With the number of people involved in the mapping and creating the documentation, there needs to be consistency for both the look and feel and for the content. It is important that users feel that one person has written everything. This could include providing templates for the documents and forms with guidance for their completion, how the maps and documents are to be numbered and issued, the design rules for the process maps, how the responsibilities are to be described and what data is to be added to the maps to supply the reports.

This could be included in a Policy document with perhaps a guidance document to state how the rules are to be applied.

5.4.3 Schedule

The time-line or draft Schedule from the Business Case now needs to be converted into a Project Schedule. This will provide the Departments with dates for when they will need to release their staff and kept up-to-date on a regular basis.

The timescales for the acquisition and installation of the IT Infrastructure also need to be included as most of the work will be dependent upon this and if there are any complications, the delay can then be conveyed to the Departments. The training may also be a dependency on the IT Infrastructure if the training is to be done on the Software tool.

5.4.4 Communications

A Communication Plan will be required. It should include announcing the start of the project to all staff. Then, as each department is started, all their staff should be informed about what is to be done and how they are likely to be involved. When their section is completed, they will be trained to use it.

There should also be a Progress Report to all staff, either on a regular basis, e.g. in a Weekly Newsletter, or as each Department is completed.
Finally when the Management System is complete, there will be a presentation to all the staff.

**6.0 Next Step**

This paper has described how Process Mapping Software should be chosen and the re-design planned. The next step is to start the project and to discover what Lessons can be learnt for the Planning.

**7.0 References**

Determining the Quality of Mathematical Software Using Reference Data Sets

Keith J. Lines, Ian M. Smith

Data Science Group, National Physical Laboratory, Teddington, TW11 0LW, UK
keith.lines@npl.co.uk, ian.smith@npl.co.uk

Abstract

This paper describes a methodology for evaluating the numerical accuracy of software that performs mathematical calculations. The authors explain how this methodology extends the concept of metrological traceability, which is fundamental to measurement, to include software quality.

Overviews of two European Union-funded projects are also presented. The first project developed an infrastructure to allow software to be verified by testing, via the internet, using reference data sets. The primary focus of the project was software used within systems that make physical measurements. The second project, currently underway, explores using this infrastructure to verify mathematical software used within general scientific and engineering disciplines.

Publications on using reference data sets for the verification of mathematical software are usually intended for a readership specialising in measurement science or mathematics. This paper is aimed at a more general readership, in particular software quality specialists and computer scientists. Further engagement with experts in these disciplines will be helpful to the continued development of this application of software quality.

Keywords: Software, Standards, Traceability, TraCIM, ValTraC, Verification

1.0 Introduction

National Measurement Institutes (NMIs), such as the National Physical Laboratory (NPL) [1] in the UK and the Physikalisch-Technische Bundesanstalt (PTB) [2] in Germany, have developed methodologies to evaluate quality criteria, such as
numerical accuracy, of software implementations of mathematical algorithms (hereafter referred to as mathematical software).

Mathematical software is increasingly essential to modern-day measurement systems. Quantifying the effect software quality has on the accuracy of the measurement results provided by such systems is an important activity for NMIs and industry. The demand for ever more accurate measurements, supported by ever more complex software, can only increase [3].

In the following paper the authors present an overview of a methodology to evaluate the numerical accuracy of mathematical software. This methodology uses reference data sets, sometimes known as *numerical artefacts*. These artefacts are analogous to the physical artefacts with which NMIs establish the *metrological traceability* of measurements of physical quantities such as mass.

Sections 2 and 3 provide background information, including an introduction to metrological traceability. Section 4 provides an overview of the methodology. As well as being used for verification of mathematical software within measurement systems, the methodology can be applied to mathematical software more generally [4]. Section 5 explains how the concept of metrological traceability can be extended to the verification of mathematical software.

Coordinate measuring machines (CMMs) [5] are described in section 6. The verification of mathematical software within these devices is a major area of application for the methodology.

Section 7 contains an overview of the European Union-funded project TraCIM [6]. Amongst other deliverables the project developed services that allow reference data sets to be used, via the internet, directly in devices running the software to be verified [7]. Section 8 describes another European Union-funded project, ValTraC [8], that is exploring the use of the TraCIM system within general scientific and engineering disciplines. The authors end with some thoughts on future directions for this work.

### 2.0 Verification and Validation

This paper uses the definitions of verification and validation provided in ISO/IEC/IEEE 24765:2010 [9]:

<table>
<thead>
<tr>
<th>Verification</th>
<th>Formal proof of program correctness.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled.</td>
</tr>
</tbody>
</table>
Therefore by verification the authors mean —Has the mathematics been implemented correctly?— and by validation, —Has the implementation of the mathematics met a user requirement?

3.0 Metrological Traceability

Metrological traceability is a fundamental concept within measurement science (or metrology). The International Bureau of Weights and Measures (BIPM) [10] is the body with responsibility for ensuring and promoting the global comparability of measurements via the International System of Units (SI). In its International Vocabulary of Metrology (VIM) [11], BIPM defines metrological traceability as:

—Property of a measurement result whereby the result can be related to a reference through a documented unbroken chain of calibrations, each contributing to the measurement uncertainty”

Metrological traceability ensures that a kilogram of potatoes purchased from a food retailer in the UK must (subject to measurement uncertainty) have the same mass as the UK’s national standard kilogram held at NPL. The retailer, trading standards officers, etc. must use weighing devices calibrated in an unbroken traceability chain ending at NPL, and NPL must in turn be traceable to the BIPM.

International intercomparisons between NMIs demonstrate that the participants’ measurement results are equivalent and therefore verify the way they maintain and disseminate their national mass standards. Such intercomparisons are also subject to measurement uncertainties. A discussion of measurement uncertainty is beyond the scope of this paper. Reference [12] provides an introduction to the topic.

Figure 1 provides a simple example of a traceability chain.

Figure 1: Example of a traceability chain

As will be described in the section 5, the concept of metrological traceability can be extended to verification of mathematical software. But first, an explanation of some further terminology is required.
4.0 Verification of Mathematical Software

The following section provides an introduction to the methodology. It begins with a summary that introduces the concepts of computational aims and reference data sets.

4.1 Summary

The methodology described in this paper can be summarised as follows:

1. Provide a clear, complete and unambiguous statement of the mathematics to be implemented. A document called the specification of a computational aim (or simply computational aim) contains this statement. A computational aim does not provide any details of how the mathematics will be implemented. An online database of computational aims [13] was developed as part of the TraCIM project (see section 7).

2. A computational aim is used as the basis to generate reference data sets. A reference data set consists of a number of reference pairs. A reference pair comprises reference input data and corresponding reference output data.

3. The software to be verified is presented with a selection of reference input data as test data. The output generated by the software is compared with the corresponding reference output data. If the values agree according to stated criteria (e.g. to a certain number of decimal digits) the software is deemed to be of the required quality.

4.2 Generating Reference Data Sets

In theory, reference data sets could be generated by someone reading the computational aim and calculating the reference pairs by hand! In practice, a data generator is implemented in software to generate reference pairs. Data generators can be either static or dynamic. Static generators produce a file of reference pairs; dynamic generators produce reference pairs on the fly.

Reference data sets can be generated in one of two ways:

- **Forward**: Begin with reference input data and, using reference software, generate corresponding reference output data.

- **Reverse**: Begin with the reference output data and generate corresponding reference input data. E.g. for minimum circumscribed circle calculations, described in section 7.3.2, define a circle by selecting a radius length and the \((x, y)\) coordinates of the centre point. From these parameters, generate the \((x, y)\) coordinates of a set of data points such that the specified circle is the smallest that contains all of these data points.
In general, reverse generation is easier to implement than forward generation and is the preferred technique. The mathematics underlying reference data set generation is beyond the scope of this paper; reference [4] provides further details.

5.0 Software Traceability

The term traceability is well established within software engineering. For example ISO/IEC/IEEE 24765:2010 [9] defines requirements traceability as “Discernible association between a requirement and related requirements, implementations, and verifications.” Requirements traceability provides a means of validating software, i.e. ensuring requirements have been met.

In the context of this paper the authors mean traceability as providing evidence of numerical correctness through an unbroken chain ending with a computational aim via a reference data set (see figure 2, c.f. figure 1):

Figure 2: Software traceability chain

6.0 Coordinate Measuring Machines

Manufacturers of coordinate measuring machines are a major industrial user of reference data sets. A brief description of these devices and related standards will provide useful background information.

6.1 Overview of CMMs

Coordinate measuring machines (CMMs) [5] use probes to make measurements of the coordinates of points on the surface of an object. Mathematical algorithms implemented by software are used to infer information about the object (such as dimensions, positions of features, etc.). The object could have a complex geometry such as a camshaft or turbine blade.

As noted in [14], a complex engineering product (such as a gas turbine engine) can be constructed from tens of thousands of components. The importance of the accuracy and consistency of measurements made by devices such as CMMs cannot be overstated. If measurements made by one manufacturer’s CMM differs from that of another manufacturer’s, beyond an acceptable tolerance, then individual components might not fit together.

6.2 ISO 10360

International standards for testing the performance of CMMs help ensure consistency [14]. The ISO 10360 series of standards are among the most widely
adopted [15]. The majority of these standards are concerned with physical testing, such as the use of gauge blocks [16]. However, software that performs mathematical calculations (such as geometric element fitting) lies at the heart of CMMs. The quality of the CMM’s software should be as important as the quality of the physical construction of the CMM itself. But how can the quality of such software be determined? CMM manufacturers may wish their software and algorithms implemented to remain confidential [17]. Reference data sets offer one means of testing.

ISO 10360-6:2001 [18] describes a procedure for testing software using reference data sets. To quote from the introduction to this standard —The reliability of information about features that is determined from associated features is influenced by the quality of the software for computing those features.”

7.0 The TraCIM Project

“Traceability for Computationally Intensive Metrology” (TraCIM) [6] is an EU-funded project that ran from June 2012 to May 2015. Traceability, in the context of TraCIM, is described in section 5. Computationally intensive metrology refers to metrology applications that make significant use of mathematical software. Such applications include measurement devices such as CMMs.

The main aim of the project was to develop an information and communications technology (ICT) infrastructure for verifying mathematical software via the internet. This verification is traceable to computational aims via reference data sets maintained at NMIs. The work was undertaken by a consortium consisting of:

- The NMIs of the Czech Republic, Germany, Italy, the Netherlands, Poland, Slovenia and the UK.
- The universities of Huddersfield, Osfalia, York and Zwickau.

CMM manufacturers Hexagon, Mututoyo, Werth and Zeiss were unfunded industrial partners in the project. Their role was to provide guidance to ensure meeting the needs of industry was at the heart of TraCIM.

7.2 Work Packages

The work was divided into the following work packages (WPs):

- WP1: Framework for traceable computation in metrology
  The starting point for the project was to categorise various metrology areas (e.g. electricity and magnetism, length, mass and radiometry) and identify mathematical calculations relevant to those areas [19]. Calculations relevant to more than one metrology area are termed interdisciplinary.
Requirements for the ICT infrastructure were defined as part of WP1. The implementation was carried out in WP5.

- **WP2: Formal statement of computational aims**
  Software implementing a mathematical calculation can only be verified if there is a clear, complete and unambiguous statement of the mathematics. As stated in section 4.1, in the context of work summarised in this paper, these statements are called computational aims.

  The main aim of WP2 was to develop a selection of computational aims for mathematical calculations required by metrology areas identified in WP1. A common template for these computational aims was developed and a searchable database made available online [13].

- **WP3: Generation of reference data**
  In this WP, data generators were developed for a selection of computational aims identified in WP1 and WP2.

- **WP4: Performance metrics**
  In this WP, metrics were designed to evaluate the performance of software under test. E.g.:
  - How close is the reference data set to the true mathematical solution of the computational aim? Knowing the quality of the reference data set is a part of knowing the quality of the software under test [20].
  - Define a maximum permissible error (MPE) that applies in the relevant metrology domain.

- **WP5: Launch of TraCIM System**
  In this WP, the ICT infrastructure whose requirements were drawn up as part of WP1 was developed and demonstrated for the unfunded partners of the project.

Work packages were divided into deliverables. For the remainder of this section the authors provide further details of those deliverables most relevant to this paper.

### 7.3 WP2: Formal Specification of Computational Aims

Formal methods [21], such as Z [22], bring mathematical rigour to software specification. A point that arises almost immediately is — if metrologists use mathematics to document computational aims, how can you get more formal than that?" Computational aims are indeed heavily mathematical documents. However:

- Omissions and ambiguities may occur, even in computational aims expressed using mathematical notation; would the use of formal methods allow these omissions and ambiguities to be identified and addressed?
Would the added discipline that formal methods bring allow better, more clearly thought out, computational aims to be written?

Specifications using formal methods can be analysed using software tools. The University of York was awarded a one-year research grant to explore the use of formal methods for the specification and analysis of computational aims. The following subsections summarise the research. Anyone interested in further details of the formal specification work undertaken as part of TraCIM is welcome to contact the authors of this paper.

7.3.1 Initial Review

The first stage was to decide which formal specification language to use. Both VDM [23] and EXPRESS [24], which has received attention in the field of metrology, were considered. However, Z was selected, as its expressive style is closest to the mathematics used to write computational aims; the other languages considered use more software-oriented constructions. Z is supported by software tools that provide syntax and type checking, as well as other features [25]. An ISO standard has also been defined for Z [26].

7.3.2 Z Specification: Minimum Circumscribed Circle (MCC)

The next stage was to apply Z to specifying some existing computational aims. The first computational aim chosen was “Minimum circumscribed circle (MCC) to data in the $xy$-plane” (search [11] for “minimum circumscribed circle”). Being straightforward, this computational provides a good starting point for exploring the use of formal methods.

The informal description of this computational aim is “Determine centre coordinates and radius of the circle of minimum radius that circumscribes a given set of points in the $xy$-plane”; see figure 3.

$Z$ specifications are structured using schemas. The upper section of a schema contains variable declarations; the lower section defines the relationship between the values of the variables and any constraints on these values. Schemas can be named and combined with other schemas.

Schema $MCCInputs$, defined below, specifies the inputs to the computation:

![Figure 3. Minimum Circumscribed Circle](image-url)
• $M$, the number of data points
• $X$, a matrix ($m \times 2$) containing the coordinates of those data points; $m$ is constrained to be a non-zero natural – i.e. the set of data points must not be empty

\[
MCCInputs
\begin{align*}
  m & : N_1 \\
  X & : M \\
  X & \in \text{realmatrix}(m, 2)
\end{align*}
\]

Schema $MCCOutputs$ specifies the outputs of the computation. These outputs are:
• $X_0$, a vector containing two real numbers which are the $(x, y)$ coordinates of the centre of the MCC
• $r$, a real number that is the radius of the MCC. $r$ is constrained to be $\geq 0$

\[
MCCOutputs
\begin{align*}
  X_0 & : \text{realvector}(2) \\
  r & : R \\
  r & \geq 0
\end{align*}
\]

Schema $MCCComputation$ completes the specification of the computational aim. Including the name of other schemas in the upper section of a schema imports all variable declarations, constraints etc. from those schemas.

\[
MCCComputation
\begin{align*}
  MCCInputs \\
  MCCOutputs
\end{align*}
\]

\[
\begin{align*}
  (X_0(1), X_0(2), r) &= \text{safemin}_v( \\
  \{\langle x_0, y_0, r \rangle : R : r \in \mathbb{R} | \\
  \{i : 1..m : (x_0 X(1) - x_0^2 + (X(2) - y_0)^2 - r^2 \leq 0) \\
  \{i : 1..m : (x_0, y_0, r, r) \} \})
\end{align*}
\]

• $\langle x_0, y_0, r \rangle$ is a sequence of values that define the $x$ and $y$ coordinates and radius of a circle containing the data points held in matrix $X$
• $X(i)(1)$ and $X(i)(2)$ where $i = 1 .. m$ contain the $(x, y)$ coordinates of the data points
• The function $\text{safemin}_v$ inputs a set of pairs ($\langle x_0, y_0, r \rangle$, $r$) and returns the value of $\langle x_0, y_0, r \rangle$ where $r$ is a minimum

The definitions of $\text{safemin}_v$, $\text{realvector}$ and $\text{realmatrix}$ are not included in this paper due to space constraints.
7.3.3 Analysis of Computational Aims

Confidence in the validity of formal specifications can be increased by analysis using software tools. Mathematica [27] is a symbolic (and numerical) workbench supporting mathematical reasoning. It allows users to simplify formulae symbolically, via its FullSimplify function. Formulae that can be simplified include those that yield logical values.

FullSimplify can be used to confirm whether a formal specification has required properties. For example, the MCC for an input data set containing two (distinct) data points is the circle which has a diameter defined by those points. This property can be characterised in Z as:

\[
\text{PropertyDiag}[x_1, y_1, x_2, y_2] :=
\text{TwoPointMCCCircle}(x_1, y_1, x_2, y_2) ==
\text{DiameterLinetoCircle}(\text{Line}([x_1, y_1], [x_2, y_2]));
\]

This property is then characterised in Mathematica as:

\[
\text{PropertyDiag}[x_1, y_1, x_2, y_2] :=
\text{TwoPointMCCCircle}(x_1, y_1, x_2, y_2) ==
\text{DiameterLinetoCircle}(\text{Line}([x_1, y_1], [x_2, y_2]));
\]

where the definition of TwoPointMCCCircle contains the variable declarations, constraints etc. from MCCComputation. With the addition of some further functions, not discussed here for space reasons, FullSimplify returns True.

7.4 WP5: The TraCIM System

The TraCIM System [7] delivers traceable verification, by means of software testing, directly to systems containing the software under test. A web interface [28] allows service users to register with the system and order test(s). Each test relates to a particular computational aim. The service user will be emailed an order key that allows access to the test(s).

The TraCIM System consists of the following components:

- **The TraCIM Server**: The core software module that controls the flow of data to and from other modules. Typically the TraCIM Server is hosted by an NMI
- **Expert extensions**: For each computational aim, a software module called an expert extension implements all operations relating to that aim. As will be discussed, these operations include selecting reference pairs.
- **TraCIM Client**: This software module connects the software under test to the TraCIM Server, via the internet.
As illustrated in figure 4, a TraCIM Client requests some test data using the order key. In this example a client running within a CMM is used to contact the TraCIM Server via the internet.

![Figure 4. Request test data](image)

As illustrated in figure 5, the TraCIM Server calls up the relevant expert extension which makes a selection of reference pairs from the reference data set. Reference input data from the pairs are supplied to the software under test (ST):

![Figure 5. Obtain test data and input to software under test](image)

As illustrated in figure 6, the software under test processes the reference input data and generates output data. The TraCIM Client supplies the output data to the TraCIM Server:

![Figure 6. Software under test produces output data](image)

As illustrated in figure 7, the TraCIM Server supplies the output data generated by the software under test to the expert extension; this data is then compared to reference output data. If the output data and reference output data agree according to the required criteria (e.g. to a certain number of decimal digits) the software is deemed to have passed (i.e. is of the required quality); otherwise the software will
The TraCIM Server sends a test report to the TraCIM Client and the client outputs a PDF file:

![Diagram of TraCIM System](image)

Figure 7. Compare results and generate report

### 8.0 The ValTraC project

The “Validation of software development and analysis tools using TraCIM” (ValTraC) project is another EU-funded project; it is currently underway, running from July 2016 to December 2017.

ValTraC explores how the TraCIM System can be used to verify mathematical software from other application areas in addition to physical measurement systems such as CMMs. With guidance from industry, further expert extensions are being developed for the TraCIM System. The work is being undertaken by NPL, PTB and Osfalia University.

With reference to section 2, this project should have been named VerTraC.

### 9.0 Further Thoughts

TraCIM provides an infrastructure that allows the concept of metrological traceability to be extended to the verification of mathematical software. The key concept is a traceability chain that links the software under test to a computational aim. Further research could add greater formality to this chain. For example:

- The results of the work within TraCIM on using formal methods to define computational aims were encouraging and demonstrated there is further research that could be carried out. For example, could refinement [29] allow formal methods to extend a chain of traceability from a reference data set to a computational aim in a formal, documented manner? Would such an extension bring any benefit?

- Functional programming languages such as Haskell [30] evaluate expressions rather than assign values to variables. Mathematical libraries are being developed for Haskell [31]. Developing data generators using Haskell would be an interesting test of those libraries.

It is hoped that this paper will make a useful contribution to such research being undertaken.
10.0 Acknowledgements

This work has been carried out as part of the European Metrology Programme for Innovation and Research (EMPIR) project 15SIP06. The EMPIR initiative is co-funded by the European Union’s Horizon 2020 research and innovation programme and the EMPIR Participating States.

The research on formal methods within TraCIM was undertaken by Andy Galloway and Richard Paige of the Department of Computer Science at the University of York. Section 7.3 was adapted from reports written for TraCIM by Andy Galloway.

The authors would like to thank our colleagues Peter Harris for reviewing this paper and Stuart Davidson for his assistance in writing section 3

11.0 References

1 National Physical Laboratory. Retrieved 7th March 2017 from National Physical Laboratory: http://www.npl.co.uk/
18 ISO 10360-6:2001 Geometrical product specifications (GPS) – Acceptance and reverification tests for coordinate measuring machines (CMM) – Part 6: Estimation of error in computing Gaussian associated features
26 ISO/IEC 13568 Information technology — Z formal specification notation — Syntax, type system and semantics. First edition 2002-07-01
A Methodological Proposal and Tool Support for the HL7 Standards Compliance in the Development of Health Information Systems

A. Martínez-García¹, M.A. Olivero², A. Suárez-Bote², J.M. Sánchez-Bejines², F.J. Domínguez-Mayo², M.J. Escalona², M. Mejías², C.L. Parra-Calderón¹

¹Group of Research and Innovation in Biomedical Informatics, Biomedical Engineering and Health Economy. Institute of Biomedicine of Seville, IBiS / Virgen del Rocío University Hospital / CSIC / University of Seville, Seville, Spain
Av. Manuel Siurot, s/n, 41013, Seville (Spain)
{alicia.martinez.exts, carlos.parra.sspa}@juntadeandalucia.es

²Web Engineering and Early Testing Group, Computer Languages and Systems Department, University of Seville
ETSII, Avda. Reina Mercedes S/N, 41012, Seville (Spain)
{almudena.suarez, miguel.olivero, juan.sanchez}@iwt2.org
{fjdominguez, mjescalona, risoto}@us.es

Abstract

Health information systems are increasingly complex, and their development is presented as a challenge for software development companies offering quality, maintainable and interoperable products. HL7 (Health level 7) International, an international non-profit organization, defines and maintains standards related to health information systems. However, the modelling languages proposed by HL7 are far removed from standard languages and widely known by software engineers. In these lines, NDT is a software development methodology that has a support tool called NDT-Suite and is based, on the one hand, on the paradigm of model-driven engineering and, on the other hand, in UML that is a widely recognized standard language. This paper proposes an extension of the NDT methodology called MoDHE (Model Driven Health Engineering) to offer software engineers a methodology capable of modelling health information systems conforming to HL7 using UML domain models.

Keywords: HL7, UML, Model-Driven Engineering, health information systems, MoDHE
1.0 Introduction

Today, health information systems are increasingly complex [1]. Developing quality, maintainable and interoperable products is a challenge for software development companies wishing to find a market in healthcare systems. The need for a shared clinical history at the global level is a reality [2]. For this, it is essential to use health informatics standards that allow the establishment of standards for the exchange of clinical information [3].

There are proposals such as NDT [4], which are included in the paradigm of model-driven engineering, covering the requirements and analysis phases of web systems. However, NDT has a generalist character, therefore, in this paper we present in detail the methodological proposal that we have named MoDHE (Model Driven Health Engineering) because of its close relation with the MDE paradigm, the health field, and because it is based on the NDT methodology. Nowadays, this methodology focuses on the requirements definition phase, since it is the most critical phase in clinical projects because there are many HL7 standards that apply to that specific phase of software engineering, standards that must be applied to obtain software Interoperable health. Once the requirements phase is solved, the rest of the life cycle phases of the health software could be developed in the way equivalent to any other non-sanitary software project. In any case, the methodology has been developed with the capacity to be extended to the rest of the life cycle phases.

This methodology allows the software engineer to systematically model health information systems by working on UML diagrams and ensures compliance with HL7 standards including extension mechanisms that allow for any existing standards in HL7.

The MoDHE methodology is based on 3 main pillars. The first pillar, the methodology, offers a procedure that allows designing HL7 domain models as part of the development of a health information system. The second pillar, the HL7-based modelling language, extends to UML to model health information systems conforming to the full spectrum of HL7 standards. The third pillar, the derivation mechanisms, makes possible the interoperability between standards, facilitating the maintainability and extension of the systems. At the time of develop a first version of the methodology of MoDHE, it was decided to work with 3 of the main standards of HL7: v3, CDA and v2.x.

To automate this frame of reference, a support tool has been developed, registered as MoDHE Suite. The main objective of the tool in which we focus on this paper is to support the development of health information systems conforming to HL7, through a framework of reference that allows approaching the standards of HL7 and the modelling language of general purpose UML, using the MDE paradigm. The MoDHE methodology makes real this framework. Section 2 of this article describes the HL7 standard and its relation to UML, as well
as what is NDT. In section 3 we talk about our proposed NDT extension, MoDHE. In section 4 we present the tool that supports this methodology, MoDHE Suite, concluding with the conclusions in section 5.

2.0 Related work and context

This article proposes the joint use of two initially independent worlds of wide relevance in software engineering applied to the health environment: UML [5] (Unified Modeling Language) and HL7 (Health Level Seven).

2.1 UML y HL7 (CDA, v3 y v2.x)

UML [6] is the standard modelling language proposed by the OMG (Object Management Group) [7], an organization that promotes the use of object-oriented technologies by creating and maintaining guidelines, standards and specifications. A domain model is a conceptual model that describes the entities, attributes, roles, relationships, and constraints related to the domain of the problem [8, 9]. Instead of describing concepts of a software system, it describes the concepts of the reality of the problem itself. On the other hand, MDE (Model Driven Engineering) is a paradigm that focuses on the creation and exploitation of domain models, allowing software engineers to become independent of representation and to focus on concepts [8, 9]. In this way, a metamodel describes the concepts used in a particular domain model [8, 9]. When representing metamodels, there are many accepted notations. One of the most commonly used notations is UML.

On the other hand, HL7 defines domain models in each of its standards [10], ranging from information necessary to define messaging between systems, to the clinical documents themselves, in order to represent each of the problems or work scenarios that HL7 has gone identifying over time [11]. The HL7 standards have a common metamodel, called MIF (Model Interchange Format), from which all HL7 domain models can be modelled [12]. The MIF is formally defined in one of the HL7 standards [13]. It should be noted that the MIF is so extensive and presented in such an abstract way that, although it is very interesting from a conceptual point of view, it can cause much difficulty in its management and learning.

Each HL7 standard has an underlying metamodel, which specializes and extends the MIF. In some cases, the metamodel is not explicitly defined in a diagram, but is defined verbally in different documents. In other cases, this metamodel is explicitly defined in diagrams that use a graphic language of HL7. Given that the metamodels of some HL7 standards are described verbally in large documents, and that other HL7 standards are modelled using their own graphical language, we think that it is not easy for a software engineer to design the domain model of a software solution conforming to a specific HL7 standard. Unlike HL7 standards, software engineers in general are comfortable with more general modelling languages, such as UML.
The HL7 CDA and HL7 v2.x standards are the most used in Spain [14]. The HL7 v3 standard is the reference model where, by refinement of a subset of RIM (Reference Information Model) elements, domain models such as the HL7 CDA itself are generated. Therefore, when deciding which standards to include as part of the MoDHE framework, it was decided to work on these 3 standards. The HL7 v3, Version 3 Product Suite, consists of a set of RIM-based specifications that provide implementers with the necessary resources to work with messages, data types, and terminologies. It is considered a more robust standard than HL7 v2.x, since it reduces the semantic ambiguity and improves the processes, by having an underlying information reference model (this reference model is the RIM).

HL7 CDA is a document marking standard that specifies the structure and semantics that any clinical document must meet to be exchanged between health care providers and / or patients. This standard works on the R-MIM information model, which is a subset of RIM.

### 2.2 NDT

NDT is a methodology based on model-driven engineering that provides formal and complete support for software lifecycle management (feasibility study, requirements, analysis, design, implementation, maintenance, testing) [4]. Using NDT, we can cover the phases of the software engineering life cycle in a structured way, reducing errors and redundancies.

The NDT methodology extends the UML metamodels, supporting the design of models in each phase of the software life cycle, representing these models using UML diagrams. To support the limitations identified in each phase of the software life cycle, it defines constraints. In addition, it defines transformations between models, allowing to automatically generate the model of a specific phase, taking the information previously modeled from diagrams or models of previous phases.

Most HL7 standards are framed in the NDT Requirements Engineering phase (DRS phase). Therefore, of all the phases covered by NDT, in this paper we focus on the DRS phase. In addition, of all phases of the software life cycle covered by the NDT methodology, the DRS phase is the most complete because NDT covers a large percentage of the needs presented by this phase of the software life cycle. The final objective of the DRS phase is to model a catalog of requirements that define the needs of the system, establishing these requirements cataloged per their typology, without entering aspects related to development.

In this paper, efforts have focused on three HL7 standards that can correspond to the software information requirements (as part of the DRS phase of NDT), defining the static or structural part of the system. These standards are HL7 v3, HL7 CDA and HL7 v2.x. These 3 standards can be matched with elements modeled in the
'Data Storage Requirements' (RA) of the NDT DRS phase. Therefore, to encompass these 3 HL7 standards, the NDT metamodel corresponding to the DRS phase has been extended, adding the proper elements of each standard.

3.0 The Model Driven Health Engineering (MoDHE) framework

The MoDHE methodology enables the software engineer to systematically model health information systems by working on UML diagrams, and ensures compliance with HL7 standards including extension mechanisms that allow for any existing HL7 standards to be included. This methodology is part of one of the three pillars defined in the previous section. Thanks to this methodological proposal once the software engineer has modeled the information requirements according to HL7 using the methodology of MoDHE, you can automatically generate the analysis phase of the system and the later phases of the software life cycle as if of a project Non-health software. In this way, we can systematize the development process.

The MoDHE methodology extends to the NDT methodology, extending the metamodels that cover the elements of the software life cycle, and contemplating the metamodels of the HL7 standards, thus allowing a formal and complete framework that allows modeling a system of Health information according to HL7 in a systematic way using UML models. The MoDHE methodology allows a software engineer to model requirements using the UML language, defining in a transparent and systematic way the HL7 requirements. The following figure (Figure 1) shows the overall process.

![Figure 1: Global process](image)

The NDT methodology extends the UML metamodels, supporting the design of models in each phase of the software life cycle, representing these models using UML diagrams. To support the limitations identified in each phase of the software life cycle, it defines constraints. In addition, it defines transformations between models, allowing to automatically generate the model of a specific phase, taking the information previously modeled from diagrams or models of previous phases.
Using the MoDHE methodology, the software engineer, when constructing a sanitary software system, would perform the same phases as when using the NDT methodology (EVS, DRS, DAS, DDS, Construction, DPS, DMS), with the difference that in the DRS phase, in addition to defining non-health information requirements (and other types of requirements, such as objectives, actors, new natures, etc.), it would define health information requirements conforming to HL7.

It should be noted that, considering that the NDT methodology uses UML notation, the learning curve of the software engineer using NDT for the first time is minimal.

The MoDHE methodology extends the DRS metamodels of NDT. The DRS phase of NDT has as its final objective to model a catalog of requirements that defines the needs of the system, establishing these requirements catalogued according to their typology, without entering into aspects related to development. Specifically, MoDHE extends the storage requirements (RA) metamodel to include elements of the HL7 standards. Within the RA metamodel, the MoDHE methodology focuses on information requirements. Thanks to this feature, a software engineer can define the catalog of health requirements according to HL7.

Figure 2: Formal definition of the approach

Health software, from the point of view of the software life cycle, runs the same process as any other software development. Each HL7 standard provides guidelines and recommendations focused on a specific phase of the software lifecycle [15]. Therefore, the MoDHE methodology has been developed as an extension of the NDT methodology, supporting the development of all phases of the health software lifecycle as can be viewed on Figure 2. It should be noted that, focusing on the modelling of health requirements, it is not mandatory to design the models according to all 3 HL7 standards, the standards to be used will depend on the concrete scenario.
Considering that the HL7 v3, HL7 CDA, HL7 v2.x standards have the common characteristic that they cover the definition of information requirements of a health system, common elements in those standards are identified. These common elements allow you to define simple transformations between entities from one standard and another, allowing you to partially create the structure of the model based on a standard taking as input the model based on another standard. Taking as reference the formal definition of the metamodels that has been realized in other studies, these can be analysed and formalized the established semantic relations, by means of which the target model can be obtained following a specific source model. This process, considering such semantic relations, establishes a rule-based transformation mechanism to obtain the final model. Thanks to this transformation process, a traceability between the two metamodels is established, allowing the automation of development, as well as improving the quality and consistency of the models. For example, if you have modelled a model based on the MoDHE v2.x metamodel, you can automatically generate part of the MoDHE CDA model with the same information previously modelled by the MoDHE v2.x metamodel.

By implementing these transformations between the different MoDHE metamodels (each representing an HL7 standard), the MoDHE methodology allows reuse of models already created in an HL7 standard when modelling requirements per another HL7 standard, reusing the information, removing duplication, redundancy, and reducing errors. In addition, based on the NDT methodology, the MoDHE methodology covers the entire software lifecycle, including modelling non-health requirements in the requirements engineering phase.

4.0 MoDHE Suite: A tool to support the framework

The MoDHE framework, as we have seen, allows the design of HL7 domain models based on UML, using techniques based on the MDE paradigm. This framework is composed of the methodology and metamodels needed to make this design, as well as the transformations necessary to create models by taking the information previously modeled in other models. To make possible the practical use of this theoretical framework that allows us to generate UML models conforming to HL7, it is necessary to have a CASE support tool. This tool, which we have called MoDHE Suite has been implemented in C# as a plugin for Enterprise Architect.

To develop the MoDHE Suite tool, this research has been used as an EA basis, an already existing modeling tool that provides extension mechanisms through plugins, as well as because this modeling tool is widely known by companies and organizations in the that MoDHE Suite can be validated and evaluated once it is set up.
The definition of the specific syntax has been made using UML profiles, more specifically, UML version 2.5 [5] has been used. A UML profile is a formal extension of the UML language itself with the objective of defining new concepts from existing UML constructors, to provide them with a more precise and concrete semantics. It has been chosen to use a UML extension as a mechanism to define the concrete syntax of the UML and HL7 metamodels, since there is no problem with the use of a new specific language.

The UML extension protocol is based on 3 mechanisms:

- **Stereotypes.** Thanks to the stereotypes it is possible to define each of the elements of a specific domain that in turn will extend specific UML metaclasses.
- **Tagged value.** Tagged values allow you to add particular properties to any defined stereotype within the profile.
- **Constraints.** The constraints define the semantic conditions that apply to the stereotypes of the profile and that condition the instantiation of the metamodel.

In developing all UML profiles for MoDHE, all stereotypes extend the UML Class metaclass. It has been chosen to extend this UML metaclass in question because this metaclass aims to specify a classification of objects and to specify their properties (attributes, operations, associations, etc.), characterizing the structure and the context of these objects. Thus, when modeling an element of HL7, attributes, associations, etc. can be defined using a nomenclature similar to that used by the UML class metaclass, thus reducing the learning curve in the use of the methodology and the tool one, and improving the usability of these.

Considering that the MoDHE methodology extends the NDT methodology, the MoDHE Suite tool is based on the implementation of the NDT-Suite tool, which in turn uses the software architecture provided by EA. One of the most important aspects in the development of the MoDHE Suite tool is the translation of the concrete syntax within EA. For this, the MDG Technologies module of EA has been used. In the case of the MoDHE Suite tool, since it is an extension of the NDT-Suite tool, an MDG Technology project has not been created from scratch, failing which the 5 existing MDG Technology projects have been used for the implementation of the tool NDT-Suite, corresponding to the following phases of software development: requirements, analysis, design, testing, maintenance.

For each MDG Technology project, EA creates 2 packages by default:

- **Toolbox package.** It contains the set of stereotypes that make up the UML profile, along with its tagged values. Each of these stereotypes must be linked to the appropriate UML metaclass through a "extend" relationship. Each of the contemplated stereotypes is defined by means of a set of tagged values, which correspond to the attributes of the stereotypes defined in the UML profile.
• Profile package. It contains all those EA artifacts necessary to define the creation of diagrams according to certain stereotypes defined in the previous package, and previously selected. From this set of artifacts, the user is given the ability to model following a UML profile.

When configuring our plugin, and running EA, we have a new menu that allows modeling in UML health information requirements conforming to HL7 standards. It should be noted that the MoDHE Suite toolbox for modeling information requirements, in addition to including the elements of HL7 standards, includes the necessary elements to model non-health information requirements (subsystems, storage requirements, new natures) elements of the NDT-Suite tool.

The EA plugin for MoDHE Suite includes the profile implemented in EA, and implements the methods necessary to verify that the constraints specified in this section satisfy. It also implements the transformations defined for this tool. These transformations have been previously and theoretically defined in QVT language [16].

The QVT language is a standard language proposed by the OMG for the definition of M2M transformations. This language was born at the end of 2005 as a common proposal of several research institutions and companies. For the definition of the structure and syntax of the metamodels, this language is based on the specifications of the MOF [17] and OCL [18] standards proposed by the OMG. The use of QVT against other transformation languages such as ATL (Atlas Transformation Language) is due to the proposal presented is an extension of the NDT methodology, which uses specifications of transformations in QVT, therefore, using QVT will improve the Compatibility between both sets of transformations (those of NDT, and those of MoDHE).

These transformations have finally been implemented in C# language. These transformations could have been implemented alternatively in any other general purpose language such as Java, Python, etc.

To cover this development, first, the concrete syntax of the metamodels has been defined, and secondly the code that models the concepts of the metamodels and the transformations previously defined in QVT has been implemented.

5.0 Conclusions

In this paper, we have presented the Model Driven Health Engineering (MoDHE) framework. This proposal is an extension of NDT that is based on 3 main pillars. The first pillar, the methodology, offers a procedure that allows the design of HL7 domain models as part of the development of a health information system. The second pillar, the HL7-based modelling language, extends to UML to model health information systems conforming to the full spectrum of HL7 standards. The third pillar, the derivation mechanisms, make possible the interoperability between
standards, facilitating the maintainability and extension of the systems. At the time of developing the methodology of MoDHE, it was decided to work with 3 of the main standards of HL7: v3, CDA and v2.x

To automate this reference frame, a support tool has been developed, registered as MoDHE Suite. This tool allows the design of HL7 compliant domain models using a UML-based interface. It also allows the generation of models of a specific standard based on existing models of another specific standard. This tool has been validated in a real case study extracted from a project in which the Technological Innovation Group of Virgen del Rocio University Hospital participated, demonstrating that the solution developed is very useful.

The present paper proposes the development of a reference framework that facilitates the design of HL7-compliant domain models using a UML-based interface.

6.0 Acknowledgments

This research has been partially supported by the POLOLAS project (code TIN2016-76956-C3-2-R) of the Spanish Ministry of Science and Innovation, the KNOWBED project (code PIN-0213-2016) founded by the Andalusian Regional Ministry of Health, and Carlos III Health Institute within the call Strategic Help in Health (PITeS TiSS project, code PI15/01213), and FEDER funds.

The authors are grateful to Carlos III Health Institute for promoting the Network for Innovation in Medical Technologies and Health (‘Plataforma ITEMAS’ in Spanish, CODE PT13/0006/0036). Finally, we would also like to thank Universia Foundation for awarding PhD students with research grants.

7.0 References


Products and Prototypes:
What’s the Difference?

John Estdale

IT Architecture Specialists Ltd
The Springers, Broadlayings, Woolton Hill, NEWBURY
john.estdale@acm.org

Abstract

Prototypes are intended to demonstrate or test an idea. Commercial Off-The-Shelf products are intended for ongoing profitable sales. Their quality requirements are different: the former should be as cheap as possible whilst meeting the need for an adequate Proof-of-Concept or Demonstrator; the latter should be fit-for-purpose, cost-effective and an attractive, reliable solution to real world needs.

Selling a prototype as a product risks customer dissatisfaction, complaints, legal challenges and reputation damage. Often the prototype has to be re-written to meet product quality-level expectations.

This paper reviews the quality properties required of a product ready for delivery. This follows the ISO/IEC 25010 Quality Model, then adds important missing elements that lie “behind the scenes” in customer support, product management, legal aspects and defensive programming. It draws on a lifetime’s experience working on software products, products containing software and Software as a Service, providing facilities to end users.

Keywords: Software prototype, COTS, software product quality, supportability, ISO 25000

1.0 Introduction

ISO/IEC 25010: 2011 Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models [1] is according to Estdale [2] the leading Quality Model for software products and so called "software-intensive
computer systems’. This paper applies the Product Quality model from section 4.2, with its defined characteristics (and subcharacteristics) denoted here in italics. ISO/IEC 25010 does not evaluate the application’s specific functionality and features, as any assessment would inevitably be very dependent on the needs of the individual Customer. Typically, functionality will be missing from prototypes, affecting the 25010 qualities: *functional completeness* and *functional appropriateness*. Other qualities may be somewhat lacking too, with areas where work is deliberately left outstanding, to enable feedback on the prototype to affect the design before commercial release. Many commercial products evolve through a series of maturity levels, such as Winkler et al’s Research Vision, Research Concept, Research Prototype, Quality-assured Prototype, and Industry Product [3]. The purpose of this paper is to outline a standard for a ‘good’ finished product. If a prototype is built to this standard, then it will be more like a ‘Beta Test’ or ‘Field Trial’, as discussed in CenterCode [4], and there will be less re-engineering needed before launch.

ISO/IEC 25010 mentions “secondary users who provide support” but the quality of such advice and support is absent from the model, so Customer Support is considered here in section 3 and other issues outside the delivered software in section 4. Section 5 discusses defensive programming, which can impact many product characteristics and support issues, and section 6 draws the conclusions.

### 2.0 Standard Product Quality Properties

#### 2.1 Functional Suitability

ISO/IEC 25010 divides this into the subcharacteristics *functional completeness*, *appropriateness* and *correctness*, so a releasable product should cover a coherent set of tasks and objectives and meet clear needs.

#### 2.2 Performance Efficiency

This includes *time behaviour*, *resource utilization* and *capacity*. Short-cuts are likely to have been taken to produce a prototype, and the product will need to be good enough for at least the initial sales.

#### 2.3 Compatibility

ISO/IEC 25010 divides this characteristic into two:

##### 2.3.1 Co-existence

With the arrival of proper protected multi-programming operating systems on PCs, and the owner’s ability to install anything (s)he wanted, it became important that products were ‘well-behaved’ rather than ‘misbehaved’ [5, 6], so should:

- use supplied services for co-operative sharing of resources with the other (unknown) applications present on the platform, both physical: eg the display, physical memory, printers, CD writers, and logical: files, IP socket port numbers, etc. Applications with real-time requirements also need to co-ordinate use of processor time through scheduling, semaphores and other synchronization functions.
• hold onto resources only while obviously necessary (avoiding the ‘Deadly embrace’ of two applications each holding a resource that the other needs to proceed).
• accept that other apps may change the operating environment and affect the results. Windows PCs will commonly have some dozen scheduled jobs to check for and install updates. Meanwhile an Administrator may change the computer name, folder names and locations, system variables, and many other aspects short of triggering a reboot, and will expect the new product to continue unhindered.
• allow users to leave applications running for extended periods: days or even weeks, so there should be no unnecessary limits or timeouts, and resources such as memory should not ‘leak’ away.

2.3.2 Interoperability
This is defined as “the degree to which two or more systems, products or components can exchange information and use the information that has been exchanged”. Even if the product has no advertised interfaces to other products, users will normally expect the following capabilities:

• cut/copy/paste text between fields on a display screen, and objects between application windows
• allow the full range of Unicode characters
• send reports to a physical printer
• move or rename folders and files of user data, change access rights
• backup and restore via a variety of commands including simple copy
• recreate an output file and compare with the previous attempt, finding differences only in any timestamps.

2.4 Usability
Human Factors experts often recommend that they be given a prototype of an application in sufficient time for them to run tests with banks of users, and propose changes for implementation before product release.

2.5 Reliability
This includes fault tolerance and recoverability. In the real world, these are not purely application features, but oblige human users and production staff to handle whatever the application does not do for itself.

The first step towards fault tolerance is actually fault detection: recognising more faults with more detail, prior to trying to handle each instance automatically. The key quality technique is to check for failure immediately after every external service call, so that issues which are logically “fatal” to the continuation of the program can be announced immediately with the minimum damage, rather than emerging as incorrect results or being confused by later processing:
• with the appropriate details for the user, ie meaningful error messages that specify what failed, the impact on the application, and the corrective action needed. Good operating systems have standardised error management, with system-wide error codes and a well-established and understood set of severities, such as Fatal, Error, Warning or Comment.
• logged with the relevant information for the maintainer, including the source code location.
• (more sophisticated/optional): followed up with the appropriate clean-up/rollback, including the return of any resources whose reservation would otherwise survive program termination. Good practice is to release the rest too, as this effectively demonstrates that they are fully accounted for, which would be useful if the code is reused within some larger program.

This approach can also be applied when calling normal subroutines or functions elsewhere in the source code, as these can fail too, eg due to divide by 0, memory shortage or memory access violation.

*Recoverability* is what follows a failure (or an interruption): can the application recover the data and continue? This will depend on the developers including features such as transaction locking and Checkpoint/Restart. Any random-access files need careful design to distinguish the solid and trustworthy from transient content in the midst of some aborted change. Historically many Windows customer issues resulted from users removing disks/drives without first doing an Eject to clear any file buffers remaining to be written.

### 2.6 Security

Security is not normally something that can be easily added to a prototype, but requires more design and testing considering the expected operating environment, so will often be the focus of a later release.

### 2.7 Maintainability

ISO/IEC 25010 defines *maintainability* as “the degree of effectiveness and efficiency with which a product … can be modified by the intended maintainers”. This is then broken down to include:

#### 2.7.1 Modularity

ISO/IEC 25010 defines this as “the degree to which a system … is composed of discrete components such that a change to one component has a minimal impact on other components”. Thus one should minimise the coupling between the components [7], both in the context of the running system, and to facilitate unit testing and integration.

#### 2.7.2 Analysability

This covers the assessment of the impact of an intended change, diagnosing deficiencies and failures and identifying parts to be modified. There is no rule that maintainers are limited to the executables as delivered, or the corresponding source code. The features and assets available to maintain a product will typically include:
• user documentation or functional specification,
• design documentation including interface specifications and a data model,
• management processes, eg Woherem [8],

for defects detected during execution, but not yet identified in the code:
• logging of internally significant events,
• optional traces of execution paths, with the parameters and structures passed between components. The standard notes that a product may "analyse its own faults and provide reports": see 5.0.

and particularly for problems noticed first in the field, configuration identification:
• version numbers in all delivered items (and in these untrustworthy times, checksums for them)
• derivation details in all permanent and temporary user files created.

As the number of ‘specials’/patches for different customers multiply, it will become harder and harder to keep track, and to identify what should go into the next general release. Compiler directives or other conditional compilation techniques supported by the development environment can be used, but the variants in the source code must be visible to those editing it. Changes to fix bugs should be clearly identified, as Jones [9] suggests that 7% of them will prove to be incorrect.

2.7.3 Modifiability
Typical contributions include:
• Simple, easy-to-understand design and code, with appropriate comments
• Operating well within the algorithm and environment limits and explicitly rejecting any attempt to exceed them
• Reliable and repeatable source code retrieval and build scripts, with all tools under Configuration Management
• A managed Release process used for the last few cycles of System Test, that ensures the Release Package components are complete, consistent, configuration managed and archived.

2.7.4 Testability
It is important that applications are easily testable, both in principle (by allowing components to be tested independently and automatically) and in practice, by developing and maintaining regression test suites to demonstrate that previous functionality has not been lost. Thus, one could separate an app with a GUI into two parts, the GUI alone, and the set of functions that it invokes. One could then add a command-line interpreter component to enable test scripts to exercise the functions, unsupervised, and repeatedly, with a variety of values.

2.8 Portability
ISO/IEC 25010 talks of transferring a product between environments, rather than re-engineering the source code for new environments. Runtime dependencies were discussed under 2.3.1 Co-existence.
It is useful to treat source code portability as a goal, to minimise reliance on the
required operating environment’s implementation-specific features, whether they be (using terminology from language definitions eg [10]):

- ‘implementation-defined’ "possibly differing between processors but defined for any particular processor",
- ‘implementation-dependent’ "possibly differing between processors and not necessarily defined for any particular processor", ie absolutely anything could happen!

One should also avoid non-standard features, and suppliers with a poor commitment to upward compatibility. Any reliance on doubtful features should be checked in a regression test suite, for validating new/modified environments against the product’s virtual platform specification [4]. Ideally the product designer should be a trusted partner involved in each environmental component supplier’s Beta Trials.

2.8.1 Adaptability
Installation-dependent constants such as the organization’s name and address should be configurable, either during installation, or more flexibly, through a privileged System Administrator role and screens.

2.8.2 Installability
Installation is traditionally done by a script, which is part of a Release Package containing all the deliverables, and subject to the same quality standards, including fault tolerance and recoverability. The Testing Standards Working Party (sponsored by the British Computer Society Specialist Interest Group in Software Testing) developed a comprehensive checklist for installability requirements [11]. In addition, to minimise the number of unnecessary calls to Customer Support, the script should:

- confirm that the proposed installation will result in a fully functioning application: necessary services and resources are available, and there are no conflicts with the operating environment: version, patch state, time zone, country, display capabilities, keyboard character set (layout), security software, etc. As connectivity and Internet bandwidth may be changed over time, it may not be conclusive to test these here, and more appropriate to remind the owner of the installation requirements, and the consequences of not satisfying them.
- follow the expected practice on the product’s specified platform: conform to the relevant standards (eg [12]), for file location, user access rights, Windows Registry keys and values, etc.

The package should include:
- a reliable Uninstall, so that a user who has cancelled his license is given a clear contractual obligation to satisfy. It helps with supportability if the script rolls back all changes made during installation, so that a new installation can be attempted. Customer data is the customer’s property so should be preserved, even if we believe they have left themselves with no legitimate (licensed) way to examine it.
any other diagnostic and reporting tools thought appropriate (see 3.3). Mantyla [13] covers the broader issues involved in software deployment.

2.8.3 Replaceability
This paper is only concerned with ‘self-replaceability’: updating an existing installation with a new version of the same application. This puts additional requirements on the installation script, to:

- support customer migration, in particular conversion of any existing data and restructuring any databases where the schema is to be changed,
- not be upset by previous successful or aborted attempts at installation,
- allow multiple independent installations, eg for large customers wishing to trial a new version before a managed rollout programme with user training and data migration. This can also aid the supplier in testing with different environments and application configurations.

3.0 Customer Support

3.1 Supported Functionality and Features
Years ago, a proposed product would have a Functional Specification, and while development proceeded, a technical author would turn it into a User Manual and a Reference Manual. Development considered the functional spec to be the authority on what the product was supposed to do, so fixes were intended to correct any deviations in the implementation. Some customer Fault Reports would be resolved by corrections to a manual, and others would be rejected as a customer misunderstanding with a reference to the relevant section in the manual, or possibly an update to clarify its meaning.

As a Support Engineer, the key questions are still: “Is our software currently intended to do what the customer believes? Do we accept their claim as potentially a bug, rather than a user error?” Customers today have little documentation they can refer to, so are forced to rely on the function names offered by the UI and their interpretation of what these mean, so are in a far weaker position. On the other hand, the support team may have no authoritative source to appeal to, so conversations then involve the product’s Architect or Manager, who are open to persuasion.

3.2 Supportability
Supportability is undoubtedly a crucial characteristic of a product to its developer and distributors, and the availability of competent support is generally important to customers and users too. ISO/IEC 25010 does not directly address supportability and its definition of maintenance focusses on modification. ISO/IEC 15504-5: 2006 [14] includes a Customer Support process (OPE-2) whose purpose is “to establish and maintain an acceptable level of service through assistance and consultation to the customer to support effective use of the product”.

71
ITIL [15] states:
“The primary goal of the incident management process is to restore normal service operation as quickly as possible and minimize the adverse impact on business operations, ensuring that agreed levels of service quality are maintained.”

This does not make clear that there can be many activities involved:

1. Restore the application to a running state within an acceptable time (the ‘Recovery Time Objective’).
2. Restore user accessibility to all the available data that’s unaffected, ensuring that the system does not lose more than a certain amount of recent data (the ‘Recovery Point Objective’).
3. Recover/repair any corrupted data and complete its processing.
4. Install a Quality Assured fix to the problem.

The Supplier’s support staff may assist with all these steps, but generally restoring the service (points 1 and 2) is top priority. The Producer will need to complete a software maintenance and test process before point 4, which could take months, even a year or more. In the meantime, Support should aim to provide a ‘workaround’ so the user can proceed. Tourniaire [16] suggests that less than 5% of calls are actually the result of a software bug.

Support’s objective should be to resolve issues and minimise the need for repeat calls, described by John Seddon as avoidable ‘failure demand’ [17], that is “demand caused by a failure to do something or do something right for the customer”. Kaner [18] provides practical advice from the customer’s point of view.

Creation of ‘Patch Releases’ should be avoided (except in the case of security), due to the disruption to ongoing development, the need to System Test them in a variety of configurations, the expense of distribution and the complexity of supporting additional product variants in the field. Fixes are much better addressed in the next planned release.

### 3.3 Support Tools

Traditionally annual Support and Maintenance contracts were priced at 12-20% of the product’s sale price for access to a telephone Help Desk, but there are now many alternative approaches involving the web [16], which may be accessible to users, eg Pitney Bowes [19].

The support team need to be able to do the following for the product:

- diagnose problems occurring on customer site,
- restore service on customer site,
- recover and reinstate lost/corrupted data on customer site,
- duplicate defects on own support systems, so that maintainers can study and fix them,
- distribute fixes to affected customers,
- distribute new Release Packages across the customer base, with Release Notes, installation scripts, migration tools etc.
These generate functional and non-functional requirements on the product and its supporting assets.

Some customers may not allow operational data to be sent to Customer Support, for security, privacy, or competitive reasons. It is useful to include the support team’s diagnostic and reporting tools within the distributed Release Package, such as a System File Check and Report utility for the application and an Integrity Checker for customer files and databases, with any possible data validation, data integrity checks, and higher level semantic consistency checking. One can restrict their use by providing a limited license, or a time-restricted password.

The support team will need access to every variant of any hardware components of the product currently in the field. Good practice is to maintain three examples of each somewhere, in case one does not behave as expected, and we need a majority vote to adopt as the standard for behaviour.

Ideally one would have the same for all supported operating environments: PCs, smartphones, browsers etc, but this is something not even Microsoft attempt – hence the huge Beta Tests prior to their new releases, where it is left to hardware manufacturers and software developers to confirm their compatibility with the Microsoft standard. Egan [20] stated that “Windows 10 was tested by over 4 million people around the world before its launch”.

4.0 Product Management

Software Products have a lifecycle of their own, from launch, through multiple versions, major and minor, to retirement [21]. Over the years, the Product Manager role and a Body of Knowledge have developed [22].

4.1 Release Management

One key aspect, Release Planning and Management [23], requires the Product Manager to determine the future release Roadmap. In the early days, the focus will be on new functionality driving new sales. Jansen [24] discusses a situation where up to 70% of revenue comes from existing service contracts, and the different views that existing customers have of the cost/value of taking a new release. Tourniaire [16] argues the benefits of maintenance(-only) releases.

Established commercial practice is that any new release will at least maintain all previous functionality and features, and maintain access to existing customer data, ie be ‘upward compatible’. The responsibility to ensure that a proposed release will be acceptable to existing customers usually lies with Marketing.

Around each release in a small company there is effectively a Board-level decision involving Sales and Customer Support, with input from Marketing, Development and QA, on whether to release the Software Under Test with the defects discovered, or to delay for another System Test cycle incorporating Development’s
latest fixes. Of course, no new fixes should be incorporated after the final System Test, as this risks catastrophic regression.

4.2 The Legal Aspects

A decision to purchase a software product is made with the intention of using it. If deployed as planned, purchasers inevitably become dependent on the software working as expected, and their investment will grow over time. The Customer makes the assumption that Customer Support will be available, and that the Producer will not go bust. Therefore Purchasing should require that appropriate risk mitigation is in place [25]: through such measures as warranty, escrow, and management of the Producer’s Intellectual Property. Unfortunately, one cannot be sure that the software will continue working and not require maintenance: a mandatory security fix to an operating environment could break it. What’s needed is a Supplier with the desire and capability to modify the software, which is rather more than just having a recent copy of the source code: see 2.7.3.

Most people are familiar with the lengthy EULA (End User License Agreement) provided with most software and websites such as Microsoft, Adobe, Google, Amazon Web Services, eBay, Facebook, Twitter, etc. Software is generally licensed not sold, so that the Producer has stricter control over copying, onward sales, and usage, rather than having to depend on Copyright law [26]. The license will limit the Supplier’s liability, and contain many other detailed Terms & Conditions, generally to the detriment of the purchaser [27]. Licensees may only do what’s expressly permitted – anything else and they forfeit the right to use the software.

5.0 Defensive Programming

This term has been used historically, but was not well-defined and there appear to be no recognised standards for it. It is interpreted here as an application going beyond detection of faults in external services (see 2.5) to detect some of its own internal faults (2.7.2). This is code that does not deliver user functionality, but is additional discretionary code that contributes to prompter fault detection and localisation, and hence reliability, modularity, analysability, modifiability and testability.

To be robust against incomplete changes, modules should defend themselves against unreasonable requests and reject them at the interface, rather than accept them as legitimate, and process them as normal to “see what happens” – which will generally be random data corruption, like the notorious security vulnerability ‘buffer overflow’. It is also useful to provide a well laid-out, intelligible dump of any structures passed, again to assist localisation of problems. Both are particularly helpful while integrating contributions from different programmers.

With a product released into the real world, users may abuse it, so self-protection is recommended for each separately executable part of a Release Package. One important question for Support is to clearly identify whether a reported incident indicates a problem within the product, or bad data from somewhere else, so every external interface should be checked and be log-able. Maintainability is further
improved by applying the same approach to any major, complex internal functions, such as between phases of a compiler.

More extensive consistency checking of design or implementation invariants can be included in the Release Package, controlled by runtime switches. An example would be to check an extended dataset of numbered items, to identify whether there are any sequence numbers missing or duplicated.

6.0 Conclusions

A good software product is much more than a prototype. It sits comfortably within any of its intended operating environments and reliably produces correct results, or clear justified explanations when it can’t. It is engineered to be maintained over many years through a series of public releases. It is backed with relevant legal precautions, and a Customer Support team with appropriate tools.

7.0 References

1 ISO, ISO/IEC 25010: 2011, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models
6 Gibson S, Programs that 'behave' lend themselves to compatibility successfully, InfoWorld, Nov 3, 1986, Vol 8(43), pp69


CoFeD: A Visualisation Framework for Comparative Quality Evaluation

Elli Georgiadou, Anthony White, Richard Comley

Middlesex University, London, UK
e.georgiadou@mdx.ac.uk; a.white@mdx.ac.uk; r.comley@mdx.ac.uk

Abstract

Evaluation for the purpose of selection can be a challenging task particularly when there is a plethora of choices available. Short-listing, comparisons and eventual choice(s) can be aided by visualisation techniques. In this paper we use Feature Analysis, Tabular and Tree Representations and Composite Features Diagrams (CFDs) for profiling user requirements and for top-down profiling and evaluation of items (methods, tools, techniques, processes and so on) under evaluation. The resulting framework CoFeD enables efficient visual comparison and initial short-listing. The second phase uses bottom-up quantitative evaluation which aids the elimination of the weakest items and hence the effective selection of the most appropriate item. The versatility of the framework is illustrated by a case study comparison and evaluation of two agile methodologies. The paper concludes with limitations and indications of further work.

Keywords: Visualisation, Composite Features Diagrams (CFD), Evaluation, Agile Methods

1.0 Introduction

Quality evaluations for the purpose of selection are an everyday occurrence informing all decisions spanning the most trivial to the most profound in our individual lives, our professional lives and our scientific endeavours. The challenge of making the most appropriate selection especially from a plethora of available options becomes enormous when the risks of making the wrong choice are imminent and when they have the potential of high and even catastrophic impact.
Comparative quality evaluations are often depicted in tabular form which is a powerful representation technique [1]. The criteria and features of items under evaluation can be simply marked (ticked) as present showing which item offers the most features thus helping the evaluator home in on a short-list of plausible choices. A simple Present/Absent indication can be enhanced by the inclusion of a qualitative description such as a grade (A, B, C...) or an indication of size (Small, Medium, Large...) or even a numerical indication on an ordinal scale such as Likert Scale [2].

Despite the widespread use of tables the attributes of the items under evaluation are presented in a flat group with no attempt to classify them into primary, secondary or lower levels thus the tabular representation does not show dependencies between attributes, sub-attributes, and so on. This particular deficiency of tabular representations can be somewhat addressed by the addition of Venn Diagrams which can show inclusion in a higher set, and decomposition to lower level attributes. A further representational improvement is offered by tree structures which show hierarchical dependencies of gradually decomposed, high level groups of attributes to lower layer sub-attributes.

Despite the strengths of tree representations it is often required to model and evaluate more than one tree for the same item under evaluation. For example the ISO/IEC 25010 [3] quality model comprises 6 primary features namely Portability, Efficiency, Reliability, Functionality, Usability and Maintainability. Each one of these characteristics is subdivided into lower layer features until no more decomposition is possible.

Composite features are not directly measurable. Feature Analysis can be used for identifying factors of interest and for prioritising them Kitchenham and Jones [4] suggested the concept of complex characteristics which need to be gradually decomposed into sub-characteristics until simple and directly measurable sub-characteristics are reached.

2.0 The CoFeD Framework
2.1 A combination of Techniques

The CoFeD Framework uses a variety of techniques for capturing requirements and facilitates refining, customising, measuring, and understanding of the process as well as evaluating possible choices and solutions. These representation techniques are Tabular representation, Tree structures, CFD and Kiviat Diagrams. Figure1 shows the architecture of the CoFeD Framework and the representational techniques used at every stage of the evaluation process.
Evaluation for the purpose of selection is based on selection criteria, certain imperatives, some highly desirable features, some desirable and some ‘nice to have’. Thus the stakeholder’s requirements together with the tolerance levels for deviations need to be taken into account. CoFeD revolves around the central REVIEW process emphasising the need for early detection of problems and continuous improvement at all stages.

2.2 Composite Feature Diagrams (CFDs)

Composite Features Diagrams (CFDs) provide a visual qualitative comparison of candidate items under evaluation. Using the principles provided by Georgiadou et al. [5] for the evaluation of methods and tools. This is particularly useful when there is a large number to choose from. CFDs were subsequently used for the development of the GEQUAMO (Generic Quality Model) with emphasis on different stakeholders’ main interests/preferences [6].
The eventual choice will normally come from a short-list of items that satisfy the selection criteria unless other problems are unearthed in the process. The detailed algorithm and CFD construction rules can be found in [5, 6]. Items violating one or more of the desirable and imperative features are rejected. The genericity and versatility of the CFDs multiple tree structures were developed and demonstrated by Georgiadou where the CFD representation depicted the GEQUAMO customisable software quality model [6].

2.1.2 Kiviat Diagrams

Kiviat diagrams sometimes referred to as radar diagrams visualise multiple (often more than 3) dimensions in one combined diagram [7, 8]. All items under evaluation are compared to a set of requirements. Absences or violations of acceptable limits (threshold values) of features gradually help eliminate items enabling the evaluator to short-list all items that have the required features. Superimposing Kiviat diagrams give a visual image making the final selection both easy and demonstrable.

3.0 A case study of comparative evaluation

3.1 Agile Development Methods

Long before the term Agile Methodologies was coined there was considerable interest and discourse on iterative, incremental and evolutionary development as reported by Larman and Basili [9].

The main features of agile methodologies are the agility and simplicity of process, requirements based initially on partial knowledge, functionality delivered in small releases, small co-located development teams, pair programming, continuous testing, and on-site customer [10, 11].

According to Beck and Anders [12] traditional lifecycle models are inadequate and should be replaced by incremental design and rapid prototyping using one model from concept to code through analysis, design and implementation. Extreme Programming (XP) was successful because it emphasises customer participation and customer satisfaction. XP empowers developers to confidently respond to changing customer requirements even late in the development cycle. XP is one of several ISDMs (Information Systems Development Methodologies) such as XP, SCRUM, Crystal and EVO. These ISDMs are known as agile or lightweight to denote a breakaway from too many rules and practices Larman [13].

EVO (Evolutionary Development and Project Management) “is particularly good at dealing with large, complex, and innovative systems – it does so by breaking down the project into a series of numerous small incremental steps. Each
EVO step (or iteration) is both an opportunity to deliver some useful results to the stakeholders, and an opportunity to learn more about the system” [11].

EVO is based on ten principles including Decompose by performance results and stakeholders, Focus on improving your most valuable performance objectives first, Base your early evolution on existing frameworks and stakeholders, Motivate your team by rewarding results, Prioritise changes by value, Note place in queue, Learn fast, change fast, adapt to reality fast [10, 11]. When needing to adopt a method or tool it is important to evaluate and select the most suitable to specified requirements and acceptable levels, strength or tolerance.

3.2 Comparison of EVO and XP

Using the evaluation framework suggested by Avison and Fitzgerald in [10] and principles outlined in Kitchenham [4] each of the 7 primary elements/features was gradually decomposed into sub-features, and each sub-feature was further decomposed into its own sub-features. For example the Philosophy of a methodology can be People Oriented, Process Oriented, Organisation Oriented. Organisation oriented is further decomposed into Soft Systems and Process Innovation as shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1: FEATURE ANALYSIS AND DECOMPOSITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Philosophy</td>
</tr>
<tr>
<td>1.1 People Oriented</td>
</tr>
<tr>
<td>1.2 Process Oriented</td>
</tr>
<tr>
<td>1.3 Organisation Oriented</td>
</tr>
<tr>
<td>1.3.1 Soft Systems</td>
</tr>
<tr>
<td>1.3.2 Process Oriented</td>
</tr>
<tr>
<td>2. Model</td>
</tr>
<tr>
<td>2.1 Verbal</td>
</tr>
<tr>
<td>2.2 Analytic</td>
</tr>
<tr>
<td>2.3 Pictorial</td>
</tr>
<tr>
<td>2.4 Simulation</td>
</tr>
</tbody>
</table>

The Table in the Appendix at the end of the paper shows the decomposition of all 7 primary features into secondary and tertiary level.
However, presenting the features in lists and tables is not as understandable as a visual representation such as the tree representation shown in Fig 2.

![Diagram](image.png)

**Figure 2 - A tree representation of feature decomposition**

The decomposition of features to increasingly lower levels continues until all seven elements suggested by Avison and Fitzgerald is complete. In essence 7 tree structures are obtained.

### 3.3 Depicting multidimensional information in a single diagram

CFDs were developed as an integrating pictorial representation depicting all the features and their decompositions in one single, understandable diagram. Such profile

The primary features shown in the table found in the Appendix are depicted in the inner circle of Fig.3. The secondary and tertiary features are shown as branches in the second and third concentric circles respectively. The resulting CFD shows the complete, generic profile of an ISDM.

Fig. 3 is a Generic CFD showing the primary features and their respective, secondary and tertiary decomposition into sub-features and sub-sub-features of an ISDM brings together all 7 trees into three concentric circles the innermost circle.
showing the 7 primary features. It can be seen that each feature may have no lower layers or may have 2 or more sub-features and so on. The genericity and versatility of these multiple tree structures were demonstrated in Georgiadou [5] where the CFD representation depicted the GEQUAMO customisable software quality model [6]. Each node can be decomposed to 0, 2 or more sub-features.

See Appendix for detailed list and names of all features and sub-features

Figure 3 Generic CFD for ISDMs

3.4 Visual Comparison

Figure 4 shows how colour can be used to juxtapose two items under evaluation. This theoretical example shows that Item 2 has many more features than Item 1 and it would therefore be the obvious choice between the two.
3.5 Quantification of the strength of features

When required features are absent the choice between two or more items is straightforward. However, when the results look very similar quantification is necessary. Fig. 5 shows that the two methodologies (EVO and XP) are very similar. This can be addressed by looking not only at the presence of features but also at their strength i.e. **How well they perform**. The table in the Appendix shows the expert evaluation results for the two Methodologies and the **Threshold values** (in the last column) specified by the user. Different stakeholders are likely to place different importance and acceptance thresholds i.e. minimum required strength of the feature [7].

It is only necessary to quantify the outer branches of the trees. The strength of each branch (feature) is given on a Likert scale of 0 to 5, where 0 shows absence of a feature, 1 shows weak presence/support and so on, and 5 for extremely strong.
The table in the Appendix shows the decomposition of all the attributes as well as the averages of the expert evaluation. Only the outer branches need to be quantified.

Fig. 6 shows the profiles of EVO and XP with the average strengths of each outer branch as given by the experts and listed in the Appendix.

![Figure 6 Visual Comparison of EVO and XP](image)

It can be seen that neither of the two methodologies support simulation but they support all the other features. Unless the strength of the features is calculated it is
not possible to select the most appropriate methodology. This can be established by carrying out the numerical evaluation.

### 3.6 Numerical evaluation, final visual profile and selection

The average value (strength) of all sub-features is propagated to the parent branch until the innermost layer (primary features) is reached. Fig. 7 is a Kiviat diagram with the evaluation results superimposed. Also the threshold values specified are shown. The profiles are very similar but EVO is shown to be the best of the two short-listed methodologies.

![Figure 7 The profiles of EVO, XP and Threshold values](image)

**Key**
- **EVO in Mauve**
- **XP in Blue**
- **Threshold in Red**

A total of 37 experts 20 from industry and 17 from academia were asked to score each outer level sub-feature using the Likert scale 0, 1, 2, 3, 4 and 5 where 0 is absence of feature, 1 is present but very poor, 2 is poor, 3 is average, 4 is good and 5 very good. The averages of the experts’ scores are shown in the Appendix.
4.0 Conclusion

Visual representation has been used by scientists and non-scientists to present data and information in simple and thus understandable formats. The Chinese adage “a picture is worth a thousand words” is undisputed and is proved every day with statistics, graphs, and other sophisticated formats such as animated and colourful information made possible by powerful graphics tools and applications.

Diagrammatic representations, models and pictures form the fundamentals of scaffolding. Visualisation and representation can aid comparisons and evaluations which in turn aid the selection of appropriate solutions/items even by non-specialists.

McCandless demonstrated the importance of such representations in his readable, colourful, understandable yet powerful and encyclopaedic book Information is Beautiful [15].

Kelleher and Wagener [16] among others proposed ten principles for visualisation in scientific publications. In particular they suggest that “when necessary, multi-dimensional data can be visualised in 2D space by changing colours, shapes, and sizes”.

Evaluation for the purpose of selection is carried out by comparing to an ideal/desirable list of requirements which the items under evaluation must satisfy. A list of desirable features (requirements) is specified and then comparisons and qualitative as well as quantitative evaluations were carried out.

In this paper we proposed the CoFeD framework for visualisation, evaluation and selection which incorporates several representational techniques. This framework is suitable for comparative quality evaluation. It has been used in industry and in academia for a variety of evaluations including:

(i) mobile phones,
(ii) TV sets,
(iii) applicant selection for jobs/promotions,
(iv) university courses,
(v) CASE tools,
(vi) software development lifecycles, etc.

CFD is a representation technique for depicting Feature Analysis, gradual decomposition of attributes/features from high level (primary) to sub-attributes (secondary), sub-sub-attributes (tertiary) etc. This decomposition process produces a number of classes in the form of trees which can be represented together in a multidimensional diagram of concentric circles showing the gradual decomposition of primary features to lower levels.
The case study presented in this paper illustrates the genericity and versatility of the CFDs which aid the selection of an appropriate solution, particularly when there are a large number of options under evaluation.

Additional work needs to be carried out to replicate various evaluations where profiles of the evaluators will be included so that variability due to different levels of expertise, years of experience, specialism/job role, and possible biases can be controlled and factored out.

5.0 Acknowledgements

Many thanks are due to the expert evaluators and to our Masters students who were both critical and enthusiastic in experimenting with using CFDs (and Kiviat Diagrams) for comparisons and evaluations of methods, tools, lifecycles etc. over several years. The critical observations and comments of both the evaluators and the students helped us refine the framework. Variations and replications of the evaluations carried out enriched the current paper. Finally many thanks to the reviewers for their helpful comments.

6.0 References


### Appendix (part1)

- Tabular representation showing Feature Decomposition with averages of scoring as well as threshold values are shown.

<table>
<thead>
<tr>
<th>3. Scope</th>
<th>3.4 Implementation</th>
<th>3.5 Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3 Design</td>
<td>3.3.1 Logical</td>
<td>3.5.1 Unit testing</td>
</tr>
<tr>
<td>3.2 Requirements</td>
<td>3.2.2 Specification</td>
<td>3.5.2 Functional testing</td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3.2.1 Analysis</td>
<td>3.5.3 Integration testing</td>
</tr>
<tr>
<td>1.3 Organization Oriented</td>
<td>1.3.1 Soft systems</td>
<td>3.5.4 Acceptance testing</td>
</tr>
<tr>
<td>1.2 Process Oriented</td>
<td>1.3.2 Process innovation</td>
<td></td>
</tr>
<tr>
<td>1.1 People Oriented</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Philosophy</th>
<th>Model</th>
<th>Scope</th>
<th>Implementation</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 People Oriented</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1.2 Process Oriented</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>1.3 Organization Oriented</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.1 Verbal</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.2 Analytic</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2.3 Pictorial</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2.4 Simulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Continued on next page
| 4. Outputs | 4.1 Feasibility Report | 3 | 2 | 2 |
|            | 4.2 Requirements Specification | 4 | 3 | 2 |
|            | 4.3 Working Implementation | 3 | 4 | 4 |
|            | 4.4 Documentation | 3 | 2 | 2 |

| 5. Practice | 5.1 Methodology Background | 5.1.1 Commercial | 4 | 3 | 3 |
|            |                            | 5.1.2 Academic   | 3 | 2 | 3 |
|            | 5.2 User Experience | 5.2.1 Beginner | 3 | 2 | 2 |
|            |                            | 5.2.2 Intermediate | 4 | 3 | 4 |
|            |                            | 5.2.3 Expert | 4 | 4 | 4 |
|            | 5.3 Participants | 5.3.1 End Users | 4 | 4 | 4 |
|            |                            | 5.3.2 Developers | 4 | 4 | 4 |
|            |                            | 5.3.3 Management | 4 | 3 | 4 |

| 6. Applicability | 6.1 Multiple Domains | 4 | 4 | 4 |
|                  | 6.2 Complexity | 6.2.1 Low | 4 | 4 | 4 |
|                  |                            | 6.2.2 High | 4 | 3 | 4 |
|                  | 6.3 Size | 6.3.1 Small | 4 | 4 | 4 |
|                  |                            | 6.3.2 Medium | 3 | 3 | 4 |
|                  |                            | 6.3.3 Large | 4 | 3 | 4 |

| 7. Techniques | 7.1 Data | 3 | 3 | 4 |
|               | 7.2 Process | 3 | 3 | 3 |
|               | 7.3 Object Oriented (classes, sequence diags) | 4 | 4 | 4 |
Section 2

Risk Issues
Evaluating the Impact of Networking Security Issues on Business Productivity and Efficiency

Nickos Paltalidis ¹, Alexandru Cristina ²

¹School of Electronic Engineering and Computer Science
Queen Mary, University of London, Mile End Road, London, E1 4NS
n.paltalidis@qmul.ac.uk

²National Express
Stanstead Airport Coach Station, CM23 1RW, Essex
alexandru.cristina.ion@gmail.com

Abstract

In the 21st century, the business environment has become very sophisticated with the development of information technology. Networking systems are the essential tools used by organisations to communicate but mainly to process, store and transmit business data. Thus, the networking systems are playing an essential role for business productivity and efficiency, and without proper secured networking systems, the business development is at risk. This work aims to evaluate the impact of networking security systems on business productivity and efficiency, based on data collected from employees in small to medium-sized companies in the United Kingdom. The study’s results provide clear evidence that the network security issues are disrupting business activities, and eventually the productivity and efficiency are affected.

Keywords: (Networking Systems, Business Productivity and Efficiency)

1.0 Introduction

In our days, the business environment has undergone extreme changes through technological developments. Networking systems are one of the aspects of technological developments which have caused major changes in industry structure, market structure and business models. For example, networking systems with enormous capabilities are used by organisations to process, store and transmit
Taking into consideration the globalisation aspect, businesses have the opportunity using networking systems to work with suppliers and customers around the world where required materials can be purchased at lower rates and sales can be increased by achieving new international markets [1].

This new way of business operation generates opportunities but problems as well. InformationWeek estimates that losses due to security among organisations exceed £1 trillion annually [2]. Organisations are anxious about data security and they realise that the lack of security over a network system can have a negative impact. The damage caused by network security issues is defined as a phenomenon in the 21st century that can harm business activities. Network security is considered as one of the most essential things on the current business world. Damaged reputation, customer’s confidence loss, stolen intellectual property or customer’s access prevention can significantly affect the organisation’s efficiency and productivity [3].

2.0 Theoretical Foundation

In last years the importance of network security has been significantly increasing, especially among enterprise networks which require complex security policies [4]. According to Ling-Fang [5] the edge networks such as, LANs, PANs, WLANs and WSNs that are delimited within specific organizations and households are found at several locations through the internet. As the internet becomes larger each day and the edge networks become deeper, without protection or limited protection they are subject to intrusion from malicious users. For an efficient protection of edge networks against invaders, perimeter security such as Firewalls, IPSec gateways, intrusion detection and prevention are provided by the network security systems. Aslam, N. [6] found that in recent years network security systems have emerged as one of the greatest challenges for many organisations. The huge proliferation of the Internet and appearance of unified data, voice and multimedia services are the features of emerging communications models that can be essential for effective business activity. Nevertheless, the growing number of sophisticated security threats can significantly obstruct productivity, interrupt business operations and result in loss of information for any organisation.

New threats are appearing daily, especially in when it comes to intranet technology. The organisations are treating intranets as an internal device that is concealed behind the corporate network and by some means is invulnerable to external intimidating attacks [7]. But the intranet is a Web Application exposed to the same intimidating environment as a corporate Web site. Therefore the intranet is weakened by the scope of identical threats. Intranets are designed for employees and other trusted parties, but they not provide any guarantee against attacks by hackers, viruses or spam. If the companies are not applying dedicated intranet security policies, they may face a series of issues which are associated with
sensitive information outflow and data loss. Intranets contain private assets which are highly essential for both day-to-day activities and strategic business development. An effective attack over the intranet can interrupt business operations, cause substantial damage to an organisation's reputation, and violation of legal regulations. These factors can affect the productivity and efficiency of a business.

Vinay M. Igure [8] discovered that cumulative interconnectivity is exposing the Supervisory Control and Data Acquisition (SCADA) networks to a large range of network security issues. SCADA networks are part of the current manufacturing facilities such as electric power generation plants, oil refineries and chemical factories. Also SCADA networks can be found among engineering facilities which are huge distributed developments. Plant operatives have the duty to constantly supervise and control diverse units of the plant to guarantee its appropriate operation. The increasing development of networking technology made possible the viability of remote command and control being used for the plant. At the present time, industries are facing competitive markets, therefore it is essential to update their digital SCADA networks in order to reduce costs and to boost efficiency. SCADA networks are also connected to an organisation's corporate network and to the Internet. This connectivity may help to improve manufacturing and distribution processes, but at the same time may expose the industrial network to several security issues that are posed by the Internet. Processes that are monitored and controlled using SCADA networks may face a malicious attack which can cause considerable damage to the plant. Aside from producing physical and economic damage to an organisation, an attack over a SCADA network can affect both the environment and public safety [8].

Summarising, organisations and researchers conclude that the first time a network security system causes some problems to system holders, administrators, or operators, then it affects the performance, the complexity or the cost of maintaining the operation of any system which is held by an organisation [9]. Generally it is suggested that organisations must follow appropriate network security policies in order to avoid security issues such as cyber-attacks and denial of service, especially in the small businesses sector. Nevertheless, the majority recognises that networking security systems can affect business productivity and efficiency; and they raise the issues of network security which can affect business but are failing to be more precise on the productivity or on the efficiency side.

Therefore, the research question has been defined as “How do network security issues effect business productivity and efficiency.” The study aims to identify the common security issues in small - medium size companies and to analyse their affect into supply chain and business operation activities.
3.0 Method

The research used a quantitative deductive approach based on a survey. The survey was comprised of a questionnaire which targeted employees from small to medium-sized companies in the United Kingdom. In order to test the questions' viability, a pilot questionnaire was presented to business and IT specialists.

The questionnaire aimed a) to determine the security issues experienced by the companies, and b) to evaluate how network security issues affected business productivity and efficiency.

Thus, the questions focused on how systems such as service and support, Customer Master Data Management and production lines which are related to the company’s day by day activities were affected by network security threats. In addition, the questionnaire explored whether business process like customers’ or companies' online orders could be carried out during a network failure caused by security threats. Finally, the questionnaire evaluated whether the business plan of the companies was changed because of network security issues.

The sample was formed of forty two participants, 69% were men and 31% were women; mainly, they were Office Managers (22.22%), IT Administrators (22.22%), and Assistant Managers (19.44%).

<table>
<thead>
<tr>
<th>Occupation</th>
<th>Response Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Administrator</td>
<td>22.22%</td>
</tr>
<tr>
<td>Office Manager</td>
<td>22.22%</td>
</tr>
<tr>
<td>Assistant Manager</td>
<td>19.44%</td>
</tr>
<tr>
<td>Other</td>
<td>19.44%</td>
</tr>
<tr>
<td>Accountant</td>
<td>11.11%</td>
</tr>
<tr>
<td>Desk Officer</td>
<td>2.78%</td>
</tr>
<tr>
<td>Marketing Analyst</td>
<td>2.78%</td>
</tr>
</tbody>
</table>

The participants' age ranges was, 39% were between 25-35 years of age, 36% were between 35-55 years of age, and 25.00% were 55 years old or more.

4.0 Results and Discussion

The analysis of the survey’s data resulted in the identification of the network security issues experienced by small-medium companies, which in turn lowers productivity and efficiency.
4.1 Network Security Issues and Threats

a) The participants had knowledge about the network security issues experienced by their companies (Figure 1); 38.1% mentioned that their company has to deal with a virus or a worm, 30.95% of the participants are experienced a spyware attack, 26.19% face a browser-based attack, and 26.19% with lack of user awareness.

![Figure 1 Network Security Issues](image1.png)

b) When asked if the auto-logon process and the intranet are network security threats (Figure 2), 50% of the participants' answer was neutral, 14.3% disagreed and 9.5% strongly disagreed. A respectable 21.4% of the participants agreed that both the auto-logon process and the intranet can be considered as network security threats.

![Figure 2 Network Security Threats Cross Tab Results](image2.png)
In this situation, these network issues are linked with slow network performance, failure to access websites and disconnection from wireless or cable networks. Also, the intranet may encounter suspicious traffic such as spam, phishing, spyware, adware and malware that can affect business activity. The auto-logon process can have a huge impact on a company’s activity which can be heavily disrupted and cause enormous losses for a business if unauthorized access of private data is provided. Under these circumstances, the activity of any department in the company will be disrupted and the business may be affected by substantial losses. This could be the result of improper network security systems or users who have not received computer security awareness training [10].

### 4.2 Network security issues and business productivity and efficiency

a) In order to identify if network security issues were affecting business operations, two questions were addressed (Figure 3). The results of cross tabulation show that 47.6% of the participants' opinion was neutral, 19% disagreed, and 9.5% strongly disagreed. At the same time, 19% of the participants agreed and 4.8% strongly agreed that business operations were affected by network security issues.

![Figure 3 Network security issues affecting business operations](image)

The reason behind these well-divided opinions could be the type of business operation, such as processing of customers' orders or manufacturing department operations, and how dependent these are on the company’s network. The findings support the theory of William R. [11] which suggests that firewalls can be useful but they will not do everything. Firewalls are considered as the first line of defence when it comes to network attacks. Improper configuration of a network security system could be the cause behind the disruption of business operation.
b) Day to day business operating costs may be fixed if everything is running as planned, or costs may be variable if the materials used have a different price or the company is facing some issues. In other words, if the operation's cost is increased the business is less efficient and could generate job losses for employees and negative revenue for the stakeholders. The following chart (Figure 4) presents findings that directly support the correlation between business operations and the increased cost of operations while the company is facing a network security issue.

![Figure 4 Business operations increased cost](image)

It can be summarized that 44.44% of the participants have given a neutral answer. A large number of participants, 36.11%, agreed that the business operations' costs were increased by a network security issue. A respectable 11.11% strongly agreed that the operations cost was increased. Only 5.56% disagreed and considered that the costs were not increased. Finally, 2.78% of the participants strongly disagreed that the costs were increased.

c) In the 21st century procurement can be made online by both customers and businesses. An organisation can do E-business online by purchasing their required materials from other companies or they can do E-commerce by selling their products online to customers. When asked if their companies could not make any online purchase or process customers’ orders during a network security issue (see Figure 5) 22.2% of the participants gave a neutral answer, 5.6% disagreed and 16.7% strongly disagreed. However 47.2% agreed that their companies could not handle any online purchase and 8.3% strongly agreed.
In this case, if a company is relying on E-commerce in order to conduct their business but cannot process any orders this means that the activity is suspended for a short or a long period of time. The results correlate with the findings of Aslam, N. [6] who found that in recent year’s network security systems have emerged as one of the greatest encounters for many organisations. Nevertheless Aslam, N. [6] suggests that the growing number of sophisticated security threats can significantly obstruct productivity, interrupt business operations and result in loss of information for any organisation.

d) Every company must develop a business plan in order to define what they want to achieve and how they will achieve it. If a company experiences network security issues they may consider changing their business plan because they need a new strategy to achieve their goal.

Research shows (figure 6) 5.6% of the participants disagreed that the business plan had to be changed while the service and support system of the company was out of order because of network security issues. Interestingly, 44.4% of participants were
in agreement that the business plan had to be changed, while 33.3% gave a neutral answer. Furthermore, 16.7% strongly agreed that the business plan had to be changed because of network security issues which were affecting the service and support system of their company.

5.0 Conclusions and Further Research

Summarising the findings, the small to medium-sized medium companies are affected by specific network security issues, which in turn lowers productivity and efficiency. Network security issues such as spyware, lack of user awareness and a virus or worm are the most common experienced by the businesses. Also, intranet and auto-logon processes are considered as network security threats which provide unauthorised access to various entities that can cause losses to a business.

The effect of the above issues is demonstrated in the supply chain where the efficiency of business operations is disrupted and the cost of business operations is increased. Finally, the findings show that businesses could not make or process any order online due to network disruption and also the business plan has to be changed accordingly to the effect.

Future research is recommended that should focus on particular companies in order to analyse further aspects of their productivity and efficiency while been affected by network security issues

6.0 References

A Socio-Cognitive and Computational Model for Decision Making and User Modelling in Social Phishing

Sunil Chaudhary¹, Eleni Berki¹,₂, Linfeng Li³, Juri Valtanen⁴, Marko Helenius⁵

¹Faculty of Natural Sciences, University of Tampere, Kanslerinrinne 1, Pinni B, 30014, Tampere, Finland, chaudhary.sunil.x@student.uta.fi, eleni.berki@uta.fi

²Department of Computer Science and Information Systems, University of Jyväskylä, P.O. Box 35 (Agora), 40014, Jyväskylä, Finland

³Beijing Institute of Petrochemical Technology, Information Engineering College, 19 Qingyuan North Rd, Daxing, Beijing, China, lilinfeng@bipt.edu.cn

⁴Faculty of Education, University of Tampere Åkerlundinkatu 5, FI-33014 University of Tampere, Finland valtanenjuripetri@gmail.com

⁵Department of Pervasive Computing, Tampere University of Technology, P.O. Box 553, 33101, Tampere, Finland marko.t.helenius@tut.fi

Abstract

Systems software quality, and system security in particular, is often compromised by phishing attacks. The latter were relatively easy to detect through phishing content filters, in the past. However, it has been increasingly difficult to stop more recent and sophisticated social phishing attacks. To protect the citizens from new types of phishing attacks, software quality engineers need to provide equally sophisticated preventive technology that models people’s reactions. The authors considered the behaviour of people on the Internet from a socio-cognitive perspective and deduced who could be more prone to be spoofed by social phishing techniques. The authors herein propose a computational and interdisciplinary metamodelling methodology, which can assist in capturing and understanding people’s interactive behaviour when they are online. Online behaviour can reveal Internet users’ knowledge, information, and beliefs in a given social context; these could also constitute
significant factors for trust in social phishing circumstances which, in turn, can provide valuable insights and decision making meta-knowledge for recognition of potential victims of phishers. The proposed modelling approach is illustrated and explained using real-life phishing cases. This meta-model can i) help social computing and phishing researchers to understand users’ trust decisions from a socio-cognitive perspective, and ii) open ways to integrate artificial intelligence design techniques within software quality management practices in order to protect citizens from being spoofed by social phishing attacks. Thus, this software design quality approach will increase system security as a proactive maintenance strategy.

Keywords: security, social phishing, trust, user behaviour modelling, metamodelling, finite state machine (FSM), usability and learnability, social cognitive theory, software quality engineering.

1.0 Introduction

Social engineering is the psychological manipulation of people’s vulnerabilities to produce a desired effect based on people’s emotions and general predictive behaviour [1]. Social engineering techniques take advantage of people’s cognitive biases and misconceptions or misinformation. In the security field, this is also referred as the art and science of human hacking [2]. Currently, social engineering is one of the most prominent methods used to conduct phishing attacks. The main reasons for this are: i) social engineering is relatively easy to apply compared to other hacking techniques and ii) there is no limit to social engineering techniques as long as there exists imagination and dark creativity to exploit different situations and contexts [3]. Such phishing attacks that employ social engineering tricks are also known as social phishing [4]. Social phishing is increasing and becoming more deceptive and sophisticated and, thus, difficult to be recognised at once [5]. For example, socially-aware phishing, context-aware phishing, reverse social engineering and baiting are types of intelligent and sophisticated phishing attacks.

Since the available anti-phishing technologies are not well-designed to protect against social phishing attempts, there is a need to improve the design and software quality of social computing technologies, so they, in turn, can help protect the citizens [6]. From the software engineering perspective, the quality of the software should be assured mainly by focusing on the users, i.e., considering usable security methods and strategies [6, 7]. Equally important is equipping the people with effective anti-phishing knowledge, skills and awareness [6, 8, 9]. Otherwise stated, by understanding the vulnerabilities of the potential (and ideal for social phishers) victims, the researchers and practitioners in the areas of social computing, usable security, security education, training and awareness should be able to integrate the existing adaptive techniques (e.g. [10, 11, 12]) to protect people from being spoofed by social phishing attacks.
This research study is led by the research questions: RQ1. ‘Who are more prone to be spoofed by social phishing techniques?’ and RQ2. ‘Could online users’ behaviour and decision making be modelled?’ To answer these questions, the authors need to study the role of beliefs and contexts in the ‘netizens’ decision making process in the social phishing context, and upon this propose a logic model of the netizens’ behaviour. This model can help social computing researchers understand people’s trust decisions from a behavioural perspective, and eventually point towards adopting existing artificial intelligence techniques to adaptively prevent people from being spoofed by social phishing attacks.

In this paper, the authors first review the related user behaviour research in phishing. Second, the authors propose the way of defining a model of ICT users’ behaviour towards social phishing attacks. Last, the authors carry out a test to verify the proposed model.

2.0 Literature Review and Related Work

There exist user-related studies on phishing attacks. Some researchers conducted usability research on end-users: Zhang et al. observed users when they were using different types of anti-phishing toolbars, and discerned their usability defects [13]. Li conducted usability evaluation studies and i) concluded on what information and knowledge anti-phishing toolbars should convey and ii) gave valuable advice on how to present this security-related information in a usable way [7].

Besides the usability research, some researchers delved into phishing problems from the users’ behaviour perspective. Some researchers conducted research to find out about modelling of phishing in the social phishing context [7, 14, 15] and proceeded in constructing some models (or meta-models) of phishing under certain abstract conditions and circumstances. For example, Jakobsson conceptualised and implemented a graphical representation to capture and model the essence of phishing attacks. Such models can help in detecting the vulnerabilities of a system and determine suitable defence mechanisms for the users. [14]. Similarly, Li and other researchers built a mathematical model to depict users’ behaviour in the phishing context [7]. In this model, it was emphasised that the appropriate available knowledge is the key factor to impact the decisions of online users and thus influence their choices and beliefs in the content of a phishing attempt. Finally, in the work of Dong et al., a model for visualising the interaction between user and phishing was designed [15]. This type of model can help security professionals in determining the mismatches between users’ perceptions of phishing attacks and the attacks in reality, so that the latter can be captured and handled through the design of anti-phishing applications and suitable education.

Although the above mentioned studies contribute to the phishing research in a meaningful way, they significantly lack in addressing the problem of social phishing by considering substantial details. Further, in order to understand the end users’ interactive behaviour in the socio-cognitive context of this particular type of
emerging phishing attacks, some user behaviour research needs to be further conducted. The user behaviour modelling can actually be proved helpful in viewing and defining a sequential order of the decision making process. Therefore, one alternative to model online citizens’ behaviour is to utilise decision-based making theories, which could also be useful as supportive references for social phishing research. For example, a decision tree is one technique that is simple to understand since it describes the decision making process using a flow-chart-like model [16, 17], which consists of three types of nodes, decision nodes, chance nodes, and end nodes. A decision tree can sequentially depict how factors can affect the final decision. Adopting a more abstract computational perspective, a decision tree can be considered as a deterministic finite state automaton, in which the sequence of the factors is deterministic and each node of a decision tree is a state in the automaton. However, the rather fixed properties of the deterministic and non-deterministic finite state automata are not enough for so many different users’ behaviour modelling. This is so mainly because for different individuals (with different beliefs, knowledge and value systems) the sequence of the factors is not and cannot always be the same [7] due to the richness of the socio-cognitive context and situations and the different trust requirements and dependencies.

In previous social and cognitive context studies [18, 19, 20, 21], trust has not been considered as only a mental attitude/attribute or a pure internal belief. Instead, the concept of trust is described as consisting of three basic elements: a mental attitude, a decision to rely upon the other, and behaviour. Moreover, a mental attitude represents a belief from the evaluation of the agent’s trustworthiness, and a prediction based on the agent’s willingness and ability to produce some effects [20]. A decision to rely on the others refers to the intention to delegate the production of a desired goal [20]. Behaviour means to take actions to trust another agent and build a practical, informational relation between the parties [20]. In this way, trust is described as a framework, whose elements are isolated from each other. For example, Durante presented that the internal attribution of trustors and the environmental attribution of trustees affect the trust [20]. Another example is provided by other researchers, who also tried to present the trust model as capital and studied the cognitive dynamics from the capital perspective and point of view [21]. Furthermore, Castelfranchi et al. [19] built a tree model to describe how users make trust decisions. In this tree model, the researchers defined the different weighed value(s) for different factors of trustees. The use of different weights can result in finding how much the different factors can affect the trustors’ attitude on trust. Although these studies apply the trust theory from the social and cognitive perspectives, the models themselves are not able to define how (or how much/far) the different internal and external attributions affect each other and correspondingly affect the final trust decision. Considering the previous, the authors proceeded to a mathematical (computational) method on how to model the behaviour pattern of an online user.
3.0 Behaviour Modelling Methodology for Online Users

In order to define as many factors as possible considered in the social phishing research, the authors proceeded to an analysis of the phishing context from a socio-cognitive perspective. In the social phishing context, trustors are the potential victims who receive phishing information, and trustees are the phishers associated with their own phishing information. To build a trust relationship, trustees should present their internal and external attributions so that trustors may believe that trustees can be trusted. In a socio-cognitive theoretical framework, this means that these attributions are induced from the trustors’ perspectives and affect the trustors’ decisions [19, 20, 21, 22]. Although these studies highlighted the combined effect of competence and sincerity on the trustworthiness of information, the online security research field still lacks studies on how and on which order the attributions affect each other in their models.

The behaviour modelling methodology introduced in this paper is based on the finite state automata theory [23]. This theory represents a dynamic and computational modelling approach, which is able to describe a sequence, a selection, a multiple selection, and a repetition (or iteration) of events and/or attributions through transitions of states, which depict the situational context of phishing. The finite state automaton (FSM) has a limited, finite number of possible states. It has initial states, final states and current states. At each change of states, a deterministic or/and non-deterministic input is given, and the next state is correspondingly transitioned. The new state depends only on the current state and the symbol input. Regarding the representation of the FSM, the conventional notation is also followed, that is: a circle represents a state, an arrow represents state transition and the arrow label indicates the input value corresponding to the transition. The initial state is usually represented by an arrow with no origin pointing to the circle and the final state is drawn by a double circle. Next the authors illustrate the above through examples in which they define the relationships among these internal and external factors that outline and determine the characteristics of the potential human victims, who should rather be considered as a more-prone-to social phishing attacks group of citizens.

3.1 Modelling Internal Attributions

The internal attributions utilised and considered for modelling here include [20]:

- **Competence**: Trustee’s qualities such as skills, expertise, and knowledge needed to perform the task.
- **Willingness**: Trustee’s intention and readiness to perform the task.
- **Persistence**: Trustee’s steadiness in the intention to perform the task.
- **Dependence**: Trustor’s belief that it is either necessary or preferable to rely on the trustee in order to obtain a goal.
- **Fulfilment**: Trustor’s belief that the goal will be achieved due to the trustee.
- **Motivation**: Reasons that persuade the trustee to adopt the goal.
The authors use a phishing email as an example and subsequently analyse the case utilising the above attributes:

**Case 1:** Your flight is cancelled; please transfer 100 euro to bank account xxxx-xxxx-xxxx-xxxx as collaterals to reserve the seat for your next flight. We will refund the money back to you after your trip.

In case 1, this is how the trustee(s), i.e. the fraudster(s), presented or implied their internal attributions, factors in the social context of phishing. The *competence* is that the agency is competent on the ticket reservation; the *willingness* is that the agency is going to give this offer to every passenger, whose flight is also cancelled; the *persistence* from the message implies the money is required based on the flight company’s regulation; the *dependence* refers to paying the money is one preferable way to reserve the flight; the *fulfilment* means only the flight agency knows how to reserve the flight ticket in this special occasion; and the *motivation* of this message is that this is a part of the agency’s commitment.

To apply the finite state automata theory to model the above internal attributions, one state and its subsequent states associated with their corresponding inputs need to be defined. In the social-cognitive theory, *internal and external attributions* are the factors that impact the attitude of trustors. Herein, the authors firstly consider the internal attributions as the direct input of the model. The internal attribution of trust depends on the evaluation of the trustee’s qualities and defects. An evaluation result of different internal attributions can be selected as inputs, and the different inputs can make the current state to the different subsequent states respectively. Next, the internal attributions are discussed situation wise.

**Situation 1:** Herein, the authors firstly consider the internal attributions as the direct input of the model. If a person receives a piece of phishing information and believes that s/he has adequate knowledge on phishing prevention, the person has no intention to rely on the phishing information (trustee). This description is able to be defined in finite state machine as Figure 1. In the figure, the $S_0$ is the state that a person receives a piece of phishing information, $i_1$ stands for the input that the person believes that s/he has adequate knowledge on phishing prevention. $S_1$ represents the state that s/he has no intention to rely on the phishing information.

![Figure 1: Transferring from one state to another state with internal attribution as an input](image)

**Situation 2:** Many different internal attributions can be selected as inputs, and the different inputs can make the current state to the different subsequent states respectively. For example, in Figure 2, the $S_0$ is the state that a person receives a
piece of phishing information, $i_1$ stands for the input that the person believes that s/he has adequate knowledge on phishing prevention, and $i_2$ is that s/he has an intention to achieve the goal (e.g., update the security protection of their online banking services) in the phishing information. The state $S_1$ to be transferred with the input $i_1$ means s/he would not intend to rely on the phishing information, and the state $S_2$ to be transferred with the input $i_2$ indicates that s/he intends to rely on the phishing information.

![Figure 2: Transferring state from one state to two different states with two internal attributions as inputs](image)

**Situation 3:** When defining the ways to transfer to the target state, there might be many possible cases. Firstly, it is possible to transfer from a source state with different inputs, e.g., in Figure 3 the state $S_1$ can be reached from $S_0$ with different inputs $i_1$ and $i_2$. For example, the $S_0$ is the state that a person receives a piece of phishing information, $i_1$ stands for the input that the person believes that s/he has adequate knowledge on phishing prevention, and $i_2$ is that s/he has no intention to achieve the goal (e.g., update the security protection of their online banking services) in the phishing information. The state $S_1$ to be transferred with both inputs means s/he would not intend to rely on the phishing information.

![Figure 3: Transferring from one state to another one state with two internal attributions as inputs](image)

**Situation 4:** It is also possible that a subsequent state can be transferred from different previous states. For example, in Figure 4, a person receives a piece of phishing information directly ($S_0$) or this person is asked for help from someone who claims to be his or her friends or relatives ($S_0'$). From the both source states, the person could believe in s/he has adequate knowledge ($i_1$) so that not to intend to rely on the phishing information ($S_1$).

111
In this study, the authors assume that all the parties in the phishing context share the same scale to measure the trustworthiness; also when a trustor evaluates the trustworthiness of a trustee, a trustor may use his/her own expectations as the benchmark value to compare with the corresponding value internal attributions of trustees. If a benchmark value is different from the value of the corresponding internal attributions, the trustor may have positive or negative attitude towards the corresponding internal attributions. The formula would be as follows:

\[ w_k = A_k - E_k \] (1)

In this formula, \( A_k \) is the presented value of the \( k \)th internal attribution, \( E_k \) is the expectation value of a trustor towards the \( k \)th internal attribution, and \( w_k \) denotes the trustor’s attitude towards the \( k \)th attribution. If the presented value of an internal attribution is higher than a trustor’s expectation value, the trustor will have a positive attitude on the corresponding internal attribution. The previous research on emotion analysis and deep learning on text [10, 11, 12] can be applied to quantify \( A_k \) and \( E_k \).

Previous research studies have observed that the adaptability of agents in the trust relationship [20] and the feedback effects of some internal attribution of trustees [19] are also emphasised while modelling the trust relationship. Therefore, when defining a user behaviour model in social phishing context, feedback (i.e., past experiences) should also be considered.

For example, in the aforementioned case 1, a person receives the phishing email (\( S_0 \)), and s/he evaluates the phishing email (\( S_1 \)) utilising the adequate knowledge s/he possesses (\( i_1 \)). After his/her evaluation, s/he detects trustee’s defects, which is true in actual. Thus, s/he makes a correct decision protecting himself/herself from falling for the attack, and the positive feedback from this experience is given as input (\( i_1 \)) when the same state (\( S_1 \)) is reached next time. This means that the same input (\( i_1 \)) will be used for the same state (\( S_0 \)) in the future. Otherwise, the negative feedback is given when the same state (\( S_1' \)) is reached again, which means the same input (\( i_1' \)) is not taken into account and other positive-feedback input, for example (\( i_2 \)), is going to be used. In addition, the accumulated influence of the same feedback should be considered in the model, i.e. the feedback impacts the
attitudes towards the internal attributions in an accumulated way, and the formula (1) is now improved as follows:

\[ w_k = A_k - E_k + w_k' \quad (k = 1, 2, \ldots, 6). \] (2)

In the above formula, \( w_k' \) denotes the adjusted value according to the feedback from the past experiences, which means the \( w_k' \) can be calculated with the following formula:

\[ w_k' = A_k - E_k' \quad (k = 1, 2, \ldots, 6). \] (3)

In the above formula, \( A_k \) is the presented value of the \( k \)th internal attribution, and \( E_k' \) is the adjusted expectation value of a trustor towards the \( k \)th internal attribution, which is changed according to the feedback from the past experiences. Apparently, if the feedback is positive, \( E_k' \) is changed to a smaller value than the previous time. Otherwise, \( E_k' \) is changed to a bigger value than the previous time.

### 3.2 Modelling External Attributions

External attributions refer to positive and/or negative environmental conditions, including opportunities, resources, interferences and adversities [18, 20]. If the same case is considered, the environmental conditions can be, for example, that the phishing message is received during a festive season when people need to travel; or it is bad weather, such as a stormy or a foggy day when flights are often cancelled due to low visibility; or the phishing email recipient has a busy schedule with no time to check the credibility of the email; and the list can go on. When a person receives such a phishing message, s/he may refer to these environmental conditions as well. Similar to the internal attributions, the external ones are also considered as the inputs for user behaviour models. The only difference in the way the external attributions are treated is, the way to express how the external attributions affect the trust attitude.

For the internal ones, the trust attitude is based on trustors’ expectation value of certain internal attributions. However, in their modelling method, the authors consider that the external attributions may respectively have an effect according to the perceptions of different individuals. For example, someone may consider online social tools as a trustworthy resource (one type of external environment), but others may not think of that due to the different perception and possible negative or positive experiences regarding the security and privacy of these online social tools. Therefore, when the authors define how the trust attitude is affected, they only use the perception to depict the trustors’ attitude on external attributions. This means that when a person perceives an external attribution as a positive one, the corresponding attitude has a positive value. Otherwise, the corresponding attitude has a negative value. This could be written as follows:
In formula 4, $W_k$ is trustors’ attitude on the $k^{th}$ external attribution, and $P_k$ is the trustors’ perception on the $k^{th}$ external attribution. Same as in internal attributions, the feedback effects also apply to external attributions. The trustors’ attitude value regarding the feedback on the $k^{th}$ external attribution is given as follows:

$$W_k = P_k, k = 1, 2, \ldots, n.$$  \hspace{1cm} (4)

In formula 5, $W'_k$ is the feedback value of the $k^{th}$ external attribution from the last time, which equals the adjusted perception ($P'_k$) on the $k^{th}$ external attribution as follows in the next section.

$$W'_k = P'_k, W'_{k'}, k = 1, 2, \ldots, n.$$  \hspace{1cm} (5)

### 3.3 Modelling User Behaviour: To Trust or Not to Trust?

The authors model user behavioural patterns mainly focusing on how people consider the internal and the external attributions. Therefore, modelling of user behaviour proceeds to combine the possible states (behaviour steps) and inputs (internal and external attributions) to describe the whole process of how the users’ trust decisions are made step by step. In this case, the resulting model of user behaviour is different respectively. This is because the different expectation value of internal attributions results in the different trust attitude on the internal attributions, and the different perception of external attributions leads to the trust attitude respectively. Instead of giving a specific user behaviour model, the authors only present a modelling methodology described as follows:

- Every input is given a certain weighted value so that to explicitly add a trust attitude towards an internal or an external attribution.
- The inputs associated with their weighted value are selected, defined and added between two states.
- The order of state transitions is defined, i.e., to define the initial state, the final state(s) and the possible reachable/reached states transferred between the initial state and the final state(s).

In this way, the model finally ends up with one of the final states where a trustor makes a decision whether to trust or not to trust, in order to proceed to the next action. With this modelling methodology, the resulted model is a directed and weighted graph. Based on the definition of the weighted value of each input, the authors define that the path with the biggest weighted value is the most vulnerable mental model to social phishing attacks.

### 4.0 Modelling of Social Phishing Cases

Again, let us consider a real life phishing case that happened in China and analysed in detail by the researchers of the article in reference [24].
**Case 2:** On the 9th of February 2014, soon after the 2014 Chinese New Year, the police in Guangdong province launched a raid on saunas, karaoke bars and other venues of ill repute in Dongguan, a city in Guangdong province of China, famous for manufacturing and a highly developed sex industry. The police detained 67 people and shut down 12 venues. The news should have confirmed that the sex industry is not protected by the law in China, however. Some people got phished because of (the reporting of) these social events happening. According to the text from a newspaper published in Guangdong, some victims reported that they received phishing SMS messages saying: "Dad, I have been caught by the police when I played in Dongguan last night, please transfer *** yuan as bail to the bank account ******."

In the above case 2, a person receives the message claiming to be from a close relative and asks for money to bail out (state: S0). In this simple case, let us take a look at how the receiver will make the decision on trusting the message or not.

Firstly, the authors assume that the content of the message is considered and checked against the recipient’s knowledge. This means that the relationship should be guaranteed from the attributes of the message (input: i), e.g., the subscription number of the sender’s phone, the sender’s accent, and the certain implicit behavioural patterns. If these internal attributions in the message present higher value than the receiver’s expectation, receivers may give positive trust attitude towards these attributions (weighted attitude: wk1). The next state (state: S1) in the mind of the receiver is to consider the external attributions (input: i2), e.g., the various sources of news related to the content of the message, his/her children or relatives are actually visiting the said city, their children or relatives have habits of gambling and visiting similar venues, their children or relatives are out of the mobile phone range, and so on. If the receiver believes that the message content as described should have happened in the current occasion, the receiver’s attitude would be towards trusting the message (weighted attitude: Wk2). After the evaluation of internal and external attributions, the receiver makes a decision (state: S2).

It is also possible that after the message is received, the receiver firstly considers the external environment, which is just right at the time after the sex raid in Dongguan (input: i1’). Then the receiver has a certain attitude on the external attributions (Wk1’), the environment (state: S1’). Compared to the state S1, the receiver’s mind may be affected by the external attributions, the state S1’ is, therefore, different from his/her mental perspective. Regarding the receiver’s next move, it can be assumed that s/he may believe that s/he knows the sender so well that it is assured that the sender is caught in the sex raid. Therefore, when the receiver considers the internal attributions of the message (input: i2’), s/he has a lower expectation value on these internal attributions, and give higher weighted value on trust attitude (wk2’). After the evaluation of the internal attributions, the receiver may make another decision (state: S2’). Both the behavioural patterns are depicted in Figure 5. From the figure, one can easily compare and find out the most
vulnerable behavioural pattern, i.e. the path in the diagram with the biggest weighted value of trust attitude.

![Diagram of user behaviour modelling in the social phishing context]

**Figure 5:** One example to show the user behaviour modelling in the social phishing context

In Figure 5, let us assume that the sum value of trust weight is $\sum W$, now

- If the trustor (i.e., the SMS receiver) chooses the first path, then
  $$\sum W = w_{k1} + W_{k2} \text{ (where } k_1 = 1...6; k_2 = 1...n)$$

- If the trustor chooses the second path, then
  $$\sum W = W_{k1} + w_{k2} \text{ (where } k_1 = 1...n; k_2 = 1...6)$$

In order to trust the mobile message, the sum value of trust has to be positive, i.e., $\sum W > 0$. The higher will be the sum value of trust, so will be the possibility that the trustor will respond to the phishing attack. For the zero and negative values, i.e., $\sum W \leq 0$, the chance that the trustor will respond to the SMS will be minimal.

Humans have always been the weakest link in information security and, thus, the main target of deceitful attacks. If the question is why humans are easily deceived one can find many personal factors and reasons. Among the first that come to mind are: Socio-cultural issues such as a person might wonder how s/he will be remembered (was s/he helpful or not) or a person might not want to insult the other person (follows a high moral code), or a person thinks this is a good relationship investment (since social relationships are valued high), or a person is in a much lower social/power status (and thus does not dare to reject requests from others).

It constitutes a great challenge to consider assumptions of perceptual thresholds of multi-state positions (see e.g. [25]) to essentially argue why people lie and deceit [26], and why people trust and how they make trust decisions. Fareri et al., arguing on computational substrates of social value in interpersonal collaboration, support that ‘our brains reward us for taking the risk to trust’ [27]. Others support that people feel guilty if they do not trust other people relationships, businesses, governments etc. [28], and that this trust to strangers, even when it does not make
sense, seems to be the main reason that those bank scams on the Internet continue to flourish [28]. This study supports the need for adopting theories of vulnerable behaviour detection in assisting social phishing victims. This represents a radical innovation in design thinking about the way information is processed and informed decisions are made. There might also be limitations in utilising these theories. Special cases could be people who have lost the ability to make own decisions, such as ill people. They may have special security needs. (see e.g. [29]).

Many adaptive techniques [10, 11, 12] of learning users’ behaviour types and emotions can complementarily be applied to collect people’s emotional states and consequent attitudes, so that someone could calculate each input value to find out the most vulnerable behaviour towards social phishing attacks. With this computational and dynamic approach, the resulted (meta)model is a directed and weighted graph, that can be verifiable and testable due to its formal semantics and syntax [30]. Following this specification model someone can finally reach one of the final states, where a truster, in order to proceed, makes a decision whether to trust or not to trust. For instance, based on the definition of the weighted value of each input, the authors define and verify that the path with the biggest weighted value is the most vulnerable mental model to social phishing attacks.

This is the first time that weighted FSM modelling is used in the context of social phishing and security. Similar weighted FSM models have been applied in protocol specifications and performance analysis [31] and in speech recognition [32].

5.0 Summary, Conclusions and Future Work

As used in psychology, education, and communication, socio-cognitive models depict an individual’s knowledge use and acquisition and show how it can be directly related to observing people within the context of their social interactions, experiences, and external influences. Based on the latter, the authors introduced a computational modelling methodology to describe how people’s beliefs, knowledge and social context affect their trust decisions in the case of social phishing attacks. This methodology utilises knowledge from interdisciplinary areas, including analysis of online users’ needs, theories about trust and trustworthiness and classic computational theories through deterministic and non-deterministic modelling. Through this conceptual computational modelling, researchers and practitioners should be able to investigate vulnerable behavioural patterns of social phishing.

This particular modelling can also be helpful to anti-phishing software designers because it can assist in learning the vulnerable human behaviour patterns and warn users when a spoofing scam exploiting users’ behaviour vulnerabilities is detected. It is convenient to apply the behaviour model in software when implementing the social-context phishing prevention tools. This type of modelling that targets to assist in the learning of vulnerable human behaviour patterns could increase anti-phishing software tools’ learnability through computational intelligence.
techniques, such as machine learning. For instance, the software design of anti-phishing toolbars and related technology could outperform when implementing the related design knowledge. This work captures a new type of design thinking, rich and abstract enough to model critical user needs and details. This user behaviour modelling method will further be implemented and integrated with adaptive algorithms to support adequate technology for public awareness.

6.0 References

3 Simms C (2016). Is social engineering the easy way in? BCS the Chartered Institute for ITNOW, 58 (2), 24-25


25 Dember WN & Warm JS, Psychology of perception. Holt Rinehart and Winston 1979, USA.


Section 3

Quality Issues
The Challenges Facing Road-Side ‘m-Preneurs’ in Leap-Frogging the Constraints of ICTs in DCs: A Nigerian Case Study

Dili Ojukwu, Ph.D

Department of Mathematics/Computer Science/Statistics/Informatics
Federal University Ndufu-Alike Ikwo (FUNAI)
PMB 1010, Abakiliki, Ebonyi State, Nigeria
Email: dili.ojukwu@funai.edu.ng; dili.ojukwu@yahoo.com.

Abstract

Studies have shown that the number of mobile telecommunications operators in Nigeria has been on the increase since the first Global System of Mobile Telecommunications (GSM) licence was granted to MTN (Mobile Telecommunications Network) in 2001.

With the increase in mobile telecommunications operators also has come an unprecedented increase in subscriptions. According to the Nigerian Communications Commission (NCC) data, between 2001 and 2006 for instance, mobile telephone subscribers in Nigeria jumped from 266, 461 to 32.3 million indicating an increase of 12, 030.18 %.

One of the major by-products of the huge increase in mobile telephone subscription in Nigeria is the advent of the road-side MOBILE ENTERPRENEURS or ‘m-Preneurs’. The sight of these road-side operators (also popularly called ‘mobile engineers’) is common in most commercial cities and villages of the country. These operators, some of whom are university graduates who have taken to self employment for one reason or another, can be found operating under umbrellas, under trees and road-side shacks, under over-head bridges and in some cases, in shared shop outlets. In this paper, we investigate the challenges facing this group of young operators especially in their effort to deliver quality services to their burgeoning clients.

The paper also explores some possible solutions to the challenges facing the operators and suggests ways of quality improvement.

Keywords: m-preneur, m-commerce, leapfrog, constraints, Nigeria, e-commerce, road-side operators
1.0 Introduction

The information and communication technology (ICT) evolution has changed the face of businesses globally. In the area of telecommunication technologies (especially mobile telecommunications), the impact on business transactions has been exponential.

Mobile telephony has caught the fancy of Nigerians, young and old. With a tiny handset on their hands, they can communicate seamlessly (and with high fidelity and clarity), with the rest of the world! Gone are the days when a telegram sent from a loved one living and studying in London (UK) or Houston (US) for instance, would be read by a proud and “blessed” father in front of his cheering crowd of kinsmen. And these telegram messages, in most cases just a few lines long, came once in six or more months. How time has changed!

Today, Nigerians living in the diaspora call and talk to their loved ones, including the very old ones (who, by the way, own and operate their own mobile gadgets).

The advent of mobile telecommunications technologies in the country has empowered a growing number of young Nigerians to depend on their resourcefulness, innovation, talent and ingenuity in creating a more independent life and subsistence for themselves.

Nigeria liberalised mobile telephony in 2001 when the first GSM (Global System of Mobile Telecommunications) licence was granted to MTN (Mobile Telecommunications Network) Nigeria and ECONET Wireless Nigeria. According to the figures published by both the Nigerian Communications Commission (NCC) [1] and the National Bureau of Statistics (NBS) [2], prior to this liberalisation exercise, there were only 266,461 subscribers in the country (representing a teledensity of 0.4). Between then and 2006, this numbers increased to 32.3 million. Furthermore, the figures have continued to rise exponentially to 139.1 million in 2014 and 154,529,780 as at December 2016.

2.0 Review of Literature

Various studies have looked at the development of m-Commerce practices in various sectors of the Nigerian economy. Some looked specifically at the inhibiting factors of the technologies of the Internet in the banking and finance industry. In their study for instance, Chiemeke et al identified insecurity, inadequate operational facilities including telecommunications facilities and electricity supply as some of the major factors inhibiting the growth of Internet is the banking industry [3].
Ayo et al investigated the implementation of m-Commerce in the Nigerian banking sector and established that while most Nigerian banks offered some forms of m-Banking services, “the level of patronage, quality of cell phones, lack of basic infrastructure and security issues” posed a major threat to its wide scale implementation [4].

Exploring how small business performance in Nigeria can be boosted using m-Commerce, Okolo and Obidigbo noted that the flexibility, convenience and ubiquitousness of the mobile technology would help the small and medium sized enterprises (SMEs) operating in the country [5].

Figures have shown that with the increase in mobile telecommunications operators also came an alarming increase in subscriptions. The ever-growing popularity of the mobile telephony industry in Nigeria, coupled with a corresponding jump in the number of subscribers have created a niche market for an emergent group of roadside entrepreneurs.

As a matter of fact, one of the most positive stories to emerge from Nigeria is that a lot of unemployed Nigerians, especially the young, talented and enterprising ones, have latched on to these technologies to eke out subsistence for themselves without recourse to white collar jobs, be they from government or private sector. One of those technologies is in the area of mobile telecommunications.

Figure 1: A Typical Road-Side m-Preneur in Lagos, Nigeria (Ojukwu, 2011)

The attractiveness of mobile telephone is one major reason a lot of people get hooked to it. There is, for example the characteristic of mobility. Unlike the PSTN (publicly switched telephony network) which restricts the users’ movement, one can move about with one’s mobile handset and still communicate.
There is also the flexibility it affords its users in accessing goods and services regardless of location, or time [6]. Others include convenience, ubiquity/immediacy, real time, context awareness, personalisation [7, 6, 8, 9].

While m-Commerce can be employed for various purposes (mobile banking, mobile reservation, mobile payment, mobile marketing, etc), the question becomes: is every user of mobile technology in buying or paying for goods and services also an m-Preneur? The simple answer to that is no. For, there is a huge difference between one who buys and one who offers goods and services for sale.

One of the major by-products of the huge increase in mobile telephone subscriptions in Nigeria is the advent of the road-side ‘technologists’, ‘engineers’ and merchants popularly known locally as ‘m-preneurs’.

Available literature also identified some of the major problems confronting m-Commerce operators like the road-side m-preneurs including the issues of suitable tax laws, clear regulatory frameworks, and infrastructure [10, 13].

2.1 Who is an m-Preneur?

An m-Preneur, simply put, is an individual or body corporate who sells products and or services including exchange of information or ideas for money, using the mobile telephony technology and other mobile digital devices (such as i-Pad, Tablet, PDAs, etc). It is important to add that all the activities of an m-preneur should not necessarily start and end in the mobile wireless technology. But, as long as a big chunk of the activities and transactions are mediated by or conducted using the mobile wireless telecommunications technology, such an individual engaging in such transactions is regarded as an m-preneur.

m-Preneurship is similar to n-preneurship (or net-entrepreneurship). The only basic difference being that the latter has to do more with ICT-mediated internet activities via the use of static personal computers (pcs) than mobile telecommunications–based gadgets.

Lennon [11] observed that there are many types of m-Commerce services that are emerging in the markets today. These include mobile entertainment, like purchasing ringtones and games, mobile banking that allow consumers to conduct financial transactions from their mobile devices, and mobile brokerage that gives consumers the opportunity to buy and sell stocks from any location.

The road-side variant of m-preneurship is the type that relies, to a large extent, on the road users for quick, sometimes on-the-move services rendered to customers who are in transit. Of course, some of the services, like after-sales and after-repairs services, may continue even after the client has left the scene.
2.2 Factors Driving *m-Preneurship* in Nigeria

There are conflicting assessments of the impact of mobile commerce (m-Commerce) in Nigeria's economy. While some researchers insist that the advent of mobile telecommunications technology in the country has brought in some positive, salutary effects, others contend its impact is negative.

According to Okolo et al, while internet penetration impacted positively on Nigeria’s gross domestic product (GDP), mobile penetration had negative and statistically insignificant effects on the country [12]. Their argument is hinged on the premise that while mobile penetration aids m-Commerce in Nigeria, it negatively affects Nigeria’s trade balance and economic growth due to huge reliance on imports of mobile technologies.

As has been observed, there is a huge jump in the number and geographical spread of the people being attracted to the use of mobile technologies in Nigeria. This widespread ownership and use of mobile phones has created an increasing need and demand for professionals who can repair and service mobile phones and other digital mobile devices in the country.

On the other hand, the number of well-established companies catering for this huge demand for repairs and maintenance of damaged or malfunctioned gadgets are very few in the market. Investigations into their major motives show that they are primarily driven by the quest for survival and independence. There is a high level of unemployment (especially youth unemployment) in Nigeria.

The other thing that drives a lot of Nigerian youth to seek survival and salvation on their own is the growing unemployment rate in the country. According to figures released by the National Bureau of Statistics [2], Nigeria’s unemployment rate grew from 10.4 percent in the first quarter in 2015 to 13.9 percent in the third quarter of 2016 resulting to a total of 1.5 million unemployed Nigerians.

The rising numbers of unemployed Nigerians have been cited as being responsible for the high incidence of poverty, crime and insecurity in the country. On the other hand however, one major contributory factor to the growing numbers of *m-preneurs* in Nigeria is high unemployment situation in the country.

Our study shows that the score recorded by those who attributed their being in the business to the fact that they were unemployed was 288. As can be seen in the table below, other notable reasons include “Quick Returns”, “Expanding Markets” etc.
Table 1: Scores Recorded for Respondents’ Reasons for Becoming m-Preneurs

<table>
<thead>
<tr>
<th>Variable Code</th>
<th>WHY I STARTED THIS BUSINESS</th>
<th>TOTAL SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>Increasing number of mobile phone users</td>
<td>259</td>
</tr>
<tr>
<td>W2</td>
<td>Few blue chip companies servicing the market</td>
<td>46</td>
</tr>
<tr>
<td>W3</td>
<td>Few job opportunities (Unemployment)</td>
<td>288</td>
</tr>
<tr>
<td>W4</td>
<td>Expanding market (Job Security)</td>
<td>246</td>
</tr>
<tr>
<td>W5</td>
<td>Poorly paid job options</td>
<td>225</td>
</tr>
<tr>
<td>W6</td>
<td>I have the expertise to do the job</td>
<td>286</td>
</tr>
<tr>
<td>W7</td>
<td>More lucrative than other jobs</td>
<td>271</td>
</tr>
<tr>
<td>W8</td>
<td>Job satisfaction</td>
<td>248</td>
</tr>
<tr>
<td>W9</td>
<td>Quick returns</td>
<td>320</td>
</tr>
</tbody>
</table>

The high rate of underemployment among age “25-34” years showcased the difficulty associated with youth and graduate unemployment. This is particularly worrisome because this category falls within the very active labour force population. Policies that will address youth and graduate unemployment phenomenon in the country are, therefore required.

3.0 Research Methodology and Results

There is a yawning lack of empirical research in the area of m-preneur, particularly in Nigeria. This therefore meant that most of the empirical materials used in this investigation are in the areas of m-Commerce, e-Commerce, Telecommunications and Internet technologies. From this perspective therefore, this research work is considered exploratory.

The author conducted field surveys in two states in the South-East of Nigeria. The states are Enugu and Ebonyi. Using their capital cities of Enugu and Abakiliki respectively, we administered questionnaire copies to randomly selected members of road-side dealers in mobile devices including Cellular Phones, i-Pad, Tablets and other digital wireless devices.

With a return rate of over 70%, 85 of the 120 questionnaires administered were collected and analyzed using SPSS software. The results of the analyses helped in throwing more lights on some very important challenges confronting this group of upstarts struggling to raise their heads in the turbulent Nigerian business terrain.

In order to elicit from the respondents the level of “severity” or otherwise of the impact of the challenges confronting them in their business operations, we used a four-response-set Likert Scale ranging from “Very Low” to “Very High”. Values
were also assigned to the response sets as follows: 1 for “Very Low”, 2 for “Low”, 3 for “High” and 4 for “Very High”.

To Calculate the Rankings of the Responses (for instance, for Variable “C6”), we used the following equation:

\[ C_6 = \sum (ax4) + \sum (bx3) - \sum (cx2) + \sum (dx1) \]

(where \( C_6 \) = “Lack of Constant Supply of Electricity”, “a” = “Very High” and “d” = “Very Low”.)

\[ = \sum (80x4) + (3x3) - \sum (1x2) + (1x1) \]

\[ = (320 + 9) - (2 + 1) \]

\[ = 329 - 3 \]

\[ = 326 \]

(This process was repeated for all the Variables in order to find their scores).

An analysis of the scores generated by this measurement mechanism shows that these operators have five dominant factors inhibiting the development of their business endeavours. As can be seen in Fig. 1, these dominant challenges are:

- Lack of Constant Supply of Electricity (326)
- Lack of Government Support (316)
- High Cost of Equipment (304)
- Lack of Bank Support (319); and
- No Standardised Quality of Service (240).

![Challenges and the Levels of Impact](image)

**Figure 2: Impact of Challenges on m-Preneur Operations**
The issue of “Lack of Regular Supply of Electricity” (or what some of them called ‘epileptic power supply’) has continued to undermine the efforts of a lot of business activities in the country. So important is this issue in the country that a lot of politicians use it as an electioneering campaign issue promising to make the supply uninterruptible if elected.

For the Road-Side m-Preneur, irregular supply of power eats deeper into their income than any other consideration. This is because most of them have to buy or hire stand-by generating sets in order to keep their businesses running. In some cases, they might have electricity supply for only three or four days in a week. Even so, as is the case in a lot of places, when the power is supplied, it would not be strong enough to power some of the equipment they use due largely to low current resulting from illegal connections.

To buy a small 1.5kva-capacity generator, one needs to cough out something in the neighbourhood of 70,000 of the local currency (the Naira). Then, to adequately and constantly fill a generator of that capacity with petrol or diesel would eat up a thousand or two (on the average) of the local currency a day.

Table 2: Scores for Impact of Challenges on m-Preneurs’ Operations

<table>
<thead>
<tr>
<th>Variable Code</th>
<th>Challenges Facing Your Operations</th>
<th>Impact Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>No Standardised Quality of Service</td>
<td>240</td>
</tr>
<tr>
<td>C2</td>
<td>No Conflict Resolution Mechanism</td>
<td>120</td>
</tr>
<tr>
<td>C3</td>
<td>High Cost of Equipment</td>
<td>304</td>
</tr>
<tr>
<td>C4</td>
<td>No basic entry qualifications</td>
<td>170</td>
</tr>
<tr>
<td>C5</td>
<td>High Cost Undercutting Practices</td>
<td>213</td>
</tr>
<tr>
<td>C6</td>
<td>Lack of Constant Supply of Electricity</td>
<td>326</td>
</tr>
<tr>
<td>C7</td>
<td>No Health and Safety Regulations</td>
<td>88</td>
</tr>
<tr>
<td>C8</td>
<td>No Formal Certification before Entry</td>
<td>117</td>
</tr>
<tr>
<td>C9</td>
<td>No Organised Regulatory Body</td>
<td>33</td>
</tr>
<tr>
<td>C10</td>
<td>Poor Work Environment</td>
<td>113</td>
</tr>
<tr>
<td>C11</td>
<td>No Regulatory Laws</td>
<td>102</td>
</tr>
<tr>
<td>C12</td>
<td>Lack of Government Support</td>
<td>316</td>
</tr>
<tr>
<td>C13</td>
<td>Lack of Banks Support</td>
<td>319</td>
</tr>
</tbody>
</table>

Our investigation also shows that most of the respondents also identified lack of professionalism and regulation on the free-for-all nature of their practice. There are no rules of engagement as anybody can set up a business at any time without requisite qualifications beyond the six or so months of apprenticeship. As can be
seen in Fig. 1 and Table 2, a good number of these operators feel strongly about “No Formal Certification before Entry” (149 points) as well as “No Organised Regulatory Body” (120). These are some of the reasons which gave rise to “No Standardised Quality of Service” (264) as some of them do not pay adequate attention to the Quality of the services they render to their clients.

Contrary to Adi’s [14] conclusion that there is indeed high competitive pressure in the Nigerian telecommunication industry, the kind of competition among m-preneurs is the one brought about by “High Cost of Undercutting Practices”. This occurs when customers patronize those who are more likely to charge them less for services provided rather than the quality of the services rendered.

As shown in Table 3 (below), we also sought to find out the types of business activities they engaged in.

<table>
<thead>
<tr>
<th>S/N</th>
<th>BUSINESS ACTIVITY</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA₁</td>
<td>Installation of Software/Applications on Digital Devices</td>
<td>65</td>
</tr>
<tr>
<td>BA₂</td>
<td>Maintenance services and repairs</td>
<td>80</td>
</tr>
<tr>
<td>BA₃</td>
<td>Installation of Multimedia (Music, Video, Pictures, etc)</td>
<td>88</td>
</tr>
<tr>
<td>BA₄</td>
<td>Formatting/Flashing/Reconfiguration of Digital Devices</td>
<td>58</td>
</tr>
<tr>
<td>BA₅</td>
<td>Unlocking of Digital Devices</td>
<td>55</td>
</tr>
<tr>
<td>BA₆</td>
<td>Troubleshooting Digital Devices</td>
<td>60</td>
</tr>
<tr>
<td>BA₇</td>
<td>Sale of New/Used Mobile Digital Devices</td>
<td>100</td>
</tr>
<tr>
<td>BA₈</td>
<td>Sale of Hardware &amp; Accessories</td>
<td>100</td>
</tr>
<tr>
<td>BA₉</td>
<td>After-Sales/Repair Support Services</td>
<td>100</td>
</tr>
</tbody>
</table>

The result shows that these operators engage in four dominant categories of activities, namely:

- Repairs/Maintenance;
- Installation of Software/Applications;
- Sales of new and used devices; and
- After-sales Support.
4.0 Conclusions and Recommendations

There is no doubt that these young operators are determined to succeed in their chosen ventures. However, their level of determination alone is not sufficient to guarantee success. Given the number and magnitude of the challenges they have to contend with, the authorities should step in and help alleviate some of these operational bottlenecks.

One of the best ways to achieve this is through the provision of the necessary infrastructure like electricity. The government should also provide affordable and reliable Internet backbone, wireless connectivity, broadband and fibre-optic cables so as to make connectivity really seamless.

Furthermore, there is the need for the enactment of laws to regulate, control and define the rules of engagement for the operations of this group of very important operators. The laws when enacted, should also be able define the standards for measuring quality of performance; perform conflict resolution mechanisms, and create basic entry qualifications into the industry and well as acceptable conditions of service for the operators.

The banks need to liberalise their loan systems with a view to providing the loans needed by these operator to enable them expand their businesses and provide more employment opportunities to the teeming youngsters in the job market.

5.0 References

Gamified Budgeting for the Management of Household Finances

Michael Sievenpiper¹, Prins Butt²

¹School of Media Arts and Technology, Southampton Solent University, East Park Terrace, Southampton, Hampshire, SO14 0YN
sievm82@solent.ac.uk

²School of Media Arts and Technology, Southampton Solent University, East Park Terrace, Southampton, Hampshire, SO14 0YN
prins.but@solent.ac.uk

Abstract

The current UK economic climate is leading households into debt. The rising cost of living and inflation are resulting in households struggling with financial management. This has implications on the quality of life and economic mobility. Early motivation for and the utilisation of financial management tools can alleviate the risk of spiralling debt. In this paper, we present the case for a gamified collaborative financial management tool. We explain how current research has focused on individuals yet households often have shared and interweaving finances which would benefit from collaborative tools. We articulate the importance of motivation in financial administration and discuss the potential of gamification to motivate households in the proactive management of finances. In this regard, we describe the results of conducting a survey to investigate the case for gamification in household financial management. Our findings suggest that gamification may offer new ways to motivate household financial management and can help households manage their exposure to debt.

Keywords: Financial Management, Budgeting, Gamification, Collaboration.
1.0 Introduction

The past decade has experienced a global recession and its detrimental impact on the quality of people’s lives. The economic downturn resulting from the recession has been seen all over the world with the UK being no exception. The UK experienced a 6.4% reduction in Gross Domestic Product (GDP) over 18 months immediately following the recession [1]. In its aftermath, commodity markets impacted UK households with higher prices, rising inflation and reduced economic opportunity [2]. Households have felt the pressure of this decline through reduced prosperity, dwindling effective income and a reduction in the quality of life [2, 1].

The prevailing uncertain climate continues to strain households and is forcing many to resort to desperate and often unorthodox measures, such as mounting debt [1], which further compounds their suffering. To alleviate households from this situation there is a greater need to manage household finances and rationalise these with the economic reality.

Budgeting is how funds are allocated and utilised to reduce unnecessary expenditure and can serve as a critical tool in household financial management. Budgeting places importance on risk mitigation through careful planning and reflection and the utilisation of finite resources. It helps anticipate expenses and reduces the possibility of emotional error in spending. Despite its benefits, many households do not budget or are unable to budget effectively.

Households that struggle with budgeting tend to ignore common practices due to bias or negative perceptions towards the process [3]. They feel budgeting is unrewarding and often treat it as a trivial activity where little effort should be exerted [3, 4]. The experiences of individuals in the categorisation of expenditure also impacts the budgeting process often leading households to under utilise resources in key areas such as savings, food and travel [3]. Furthermore, households can find the process overwhelming and out of reach due to the complexity involved in the management of finances. This convoluted nature of budgeting leads to households reverting to mental strategies, making estimation errors or struggling with the timely tracking of expenditure. Mental strategies, despite their fallibility, remain the typical approach taken by households and individuals when budgeting mainly due to their flexibility and effortlessness [3, 4]. When mental strategies are used, they can lead to forgotten purchases and estimation errors which result from misunderstandings regarding the value of resources. This can in turn lead to household debt [2, 1]. Indeed, households that do not hold an active record of expenses fail to mitigate the impacts of previous expenditure on subsequent purchases [3] further contributing to the complexity of budgeting and reducing the value households place on budgeting.

Eliminating negative perceptions towards budgeting and promoting sound financial management practices remains difficult but there are existing approaches including the use of incentives, technology and training which can be effective in addressing such issues [5, 6]. However, these approaches are not without their own
challenges. Financial institutions, for example, provide a range of online banking services to support their customers. These services include financial advice, management tools and recording features. However these tools currently remain underutilised [3, 7]. A range of online web services also exist that offer households a plethora of modernised services, options, and tools towards better financial management but households have been slow to adopt such tools due to preconceived notions and lack of motivation [7]. This motivational apathy remains a persistent issue and its resolution lies at the core of effective budgeting strategies.

2.0 Gamification

Motivation strategies continue to be explored extensively in the literature with one particular trend being gamification. This is the application of game elements and principles in non-game contexts. This approach attempts to capitalise on the strengths of games as powerful motivational mediums that engross players into endless hours of gameplay. Games are enjoyed by players from diverse backgrounds and demographics. They come in various forms and have the support and following of large communities of players. They are currently the fastest growing market amongst entertainment sector due to their mass appeal [8]. In their online form, they are played by over 700 million people and make up 45% of the internet population [8]. Games and gaming transcend common barriers such as age and gender, with 46% female players and 54% male players across a wide age spectrum [8]. Many of the mechanisms utilised by games such as rewards, challenges and goals resonate with modern motivation theories which identify reinforcement, autonomy and growth as motivators [9, 10]. It is the mass appeal and motivational power that makes games and their application alluring to researchers interested in harnessing and realising their benefits in non-game contexts.

The utilisation of game elements for the purpose of enhancing motivation in non-game contexts has shown some success with an increasing number of studies indicating significant impact on motivation and engagement [11, 12, 13]. The extent to which game elements are incorporated into non-game contexts varies from study to study and has given rise to a range of terminology including game-based learning and serious games [16]. Where gamification, game-based learning and serious games depart is the scope and breadth to which game elements are applied. Game-based learning incorporates games into the learning experience and allows players to explore the learning content of a game and often involves utilising games designed for entertainment in the learning context. Gamification on the other hand takes a considered approach to applying game mechanics and principles to the non-game context [11]. Serious games attempt to go further in that they apply game elements to every aspect of the experience. These are essentially games that are designed with a serious purpose in mind – a purpose other than entertainment. Each approach has its merits and varying degrees of success. The gamification approach is particularly attractive due to its flexibility and contextualisation in comparison to a game-based approach. The gamification approach also has lower barriers when contrasted with the serious games approach.
which entails the design of a complete gaming experience. Furthermore, a number of existing studies support the case for exploring the gamification of financial management [11, 13] and have shown promising results in this regard.

3.0 Perceptions of Gamified Budgeting

In this study we investigated the strategies and perceptions of households towards budgeting and the potential for its gamification. A survey based methodology derived from survey construction techniques [14] was utilised to gather information on household financial management strategies and their limitations as well as the role that gamification can potentially play in this respect. The survey was designed with six dimensions in mind relating to personal, household, physical and local, online and mobile financial management as well as the gamification of financial management and gamified collaboration. Each question set was designed to target different aspects of financial management and thereby elicit the budgeting habits of the participants.

3.1 Demographic Data

The survey was delivered online and completed by 32 participants of whom 50% were male and 50% were female. The predominant age group was 18-23 and the respondents included 50% married or civil partnerships, 46.9% single, never married individuals as well as 3.1% divorced. Furthermore, 56.3% were currently working, 21.9% were students and 21.9% were not working.

<table>
<thead>
<tr>
<th>Demographic Variables</th>
<th>N (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Age</strong></td>
<td></td>
</tr>
<tr>
<td>18-23</td>
<td>16 (50.0)</td>
</tr>
<tr>
<td>24-29</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td>30-40</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td>41-49</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td>50+</td>
<td>5 (15.6)</td>
</tr>
<tr>
<td><strong>Gender</strong></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>16 (50.0)</td>
</tr>
<tr>
<td>Female</td>
<td>16 (50.0)</td>
</tr>
<tr>
<td><strong>Household Standing</strong></td>
<td></td>
</tr>
<tr>
<td>Single, never married</td>
<td>15 (46.9)</td>
</tr>
<tr>
<td>Married or civil partnership</td>
<td>16 (50.0)</td>
</tr>
<tr>
<td>Divorced</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td>Separated</td>
<td>0 (0.0)</td>
</tr>
<tr>
<td>Widowed</td>
<td>0 (0.0)</td>
</tr>
<tr>
<td><strong>Professional and Employment Status</strong></td>
<td></td>
</tr>
<tr>
<td>Employed</td>
<td>16 (50.0)</td>
</tr>
<tr>
<td>Self-employed</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td>Student</td>
<td>7 (21.9)</td>
</tr>
<tr>
<td>Retired</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td>Unemployed and looking for work</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td>Unemployed and not looking for work</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td>Unable to work</td>
<td>3 (9.4)</td>
</tr>
</tbody>
</table>
3.2 Results and Discussion

The first dimension of the survey focused on the individuals and the methods and technology utilised by them to manage finances. Table 2 shows that over 81% of the participants managed finances using electronic means and did so on a semi regular basis. However, 34.4% of the individuals did not record their expenditure suggesting an ad hoc approach to financial management. Interestingly, 71.9% of the participants saved or planned towards a goal showing trends found in other studies [15] and which are particularly suited to gamification where there are goals and rewards. Saving towards a goal presents a regular opportunity for an individual to track, record and reward a user on the path towards their financial goals.

Table 2: Personal Financial Management

<table>
<thead>
<tr>
<th>Question</th>
<th>Variables</th>
<th>N(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>How often do you check your bank balance?</td>
<td>Once a day</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Multiple times a day</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Multiple times a week</td>
<td>18 (56.3)</td>
</tr>
<tr>
<td></td>
<td>Multiple times a month</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td>How often do you record your spending?</td>
<td>Every time I make an expenditure</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td></td>
<td>Multiple times a day</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td></td>
<td>Once a day</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Multiple times a week</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td></td>
<td>Once a week</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td></td>
<td>Multiple times a month</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td></td>
<td>Once a month</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td></td>
<td>Never</td>
<td>11 (34.4)</td>
</tr>
<tr>
<td>What do you use to record your expenditure?</td>
<td>Notebook or paper methods</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td></td>
<td>Use a digital document</td>
<td>9 (28.1)</td>
</tr>
<tr>
<td></td>
<td>Using a website</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td></td>
<td>Using a mobile application</td>
<td>7 (21.9)</td>
</tr>
<tr>
<td></td>
<td>Other</td>
<td>0 (0.0)</td>
</tr>
<tr>
<td></td>
<td>No response</td>
<td>9 (28.1)</td>
</tr>
<tr>
<td>Do you save or plan towards a goal?</td>
<td>Yes</td>
<td>23 (71.9)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>8 (25.0)</td>
</tr>
<tr>
<td></td>
<td>No response</td>
<td>1 (3.1)</td>
</tr>
</tbody>
</table>

The second dimension of the survey focused on establishing the way households managed finances. Table 3 shows that amongst participants who shared finances with their partner, 68.9% of participants communicated decisions pertaining to financial matters with their significant other with 50% doing so at least on a bi-weekly basis. Majority (95.8%) of the participants indicated that they communicated with their partners verbally suggesting that the participants tend to rely on informal or more intimate approaches when discussing financial matters. Given that over 95% of the participants actively communicate about their finances with their partner and 50% of these do so on a fairly regular basis, there is potential for collaborative gameplay in the gamification of household budgeting.

139
Table 3: Household Financial Management

<table>
<thead>
<tr>
<th>Question</th>
<th>Variables</th>
<th>N(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does your household share finances between you and your partner?</td>
<td>Yes</td>
<td>18 (56.3)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>5 (15.6)</td>
</tr>
<tr>
<td></td>
<td>I do not have a spouse or partner</td>
<td>9 (28.1)</td>
</tr>
<tr>
<td>How frequently do you discuss finances with your spouse or partner?</td>
<td>Once a day</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Once a week</td>
<td>10 (31.3)</td>
</tr>
<tr>
<td></td>
<td>Once every two weeks</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td></td>
<td>Once a month or more</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td></td>
<td>Incomplete or no response</td>
<td>10 (31.3)</td>
</tr>
<tr>
<td>If you do, how do you communicate the information to them?</td>
<td>I talk about my finances</td>
<td>23 (95.8)</td>
</tr>
<tr>
<td></td>
<td>I show them my finances</td>
<td>0 (0.0)</td>
</tr>
<tr>
<td></td>
<td>I give them access to my finances</td>
<td>1 (4.2)</td>
</tr>
<tr>
<td>Would you like your spouse or partner to be able to see your finances?</td>
<td>Yes</td>
<td>13 (54.2)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>11 (45.8)</td>
</tr>
<tr>
<td>Would you like to be able to see your spouse or partner’s finances</td>
<td>Yes</td>
<td>16 (66.7)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>8 (33.3)</td>
</tr>
<tr>
<td>Do you and your partner save and plan to a goal together?</td>
<td>Yes</td>
<td>18 (75.0)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>6 (25.0)</td>
</tr>
</tbody>
</table>

A further dimension addressed in the survey relates to the nature of financial information tracked by households and the means by which this is tracked. Table 4 shows the responses of participants who actively recorded or tracked their expenditure. The results suggest that 62.5% of participants who tracked expenditure tended to track a summary of expenses. This may be due to convenience or limitations of their financial management strategy. As the participants tend to rely on mental strategies, tracking detailed information may be difficult if not impossible.

The sensitive nature of personal financial data means that households can be concerned about how data is stored and processed. This is evidenced in the results of Table 4 where 71.9% of the respondents did not want their personal information stored online and 87.6% did not want their financial data shared with a third party. This suggests that autonomy is an important motivator for the respondents as it allows them to self-manage and gives them a sense of control over how their sensitive data is captured and processed. This suggests autonomy needs to be a fundamental consideration in any effective budgeting strategy if it is to be adopted by households.

Past experience and perceptions about the value of technology also seem to play a role in the motivation of the participants and affects the strategies they adopt.
Table 4: Physical and Local Financial Management

<table>
<thead>
<tr>
<th>Question</th>
<th>Variables</th>
<th>N(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>What information do you track?</td>
<td>Where you spent</td>
<td>13 (40.6)</td>
</tr>
<tr>
<td></td>
<td>What time of the day you spent</td>
<td>0  (0.0)</td>
</tr>
<tr>
<td></td>
<td>What day you spent</td>
<td>8  (25.0)</td>
</tr>
<tr>
<td></td>
<td>What company or place the transaction took place</td>
<td>10 (31.25)</td>
</tr>
<tr>
<td></td>
<td>Whether you used cash or a credit/debit card</td>
<td>6  (18.9)</td>
</tr>
<tr>
<td></td>
<td>A summary of how much you spent (daily, weekly etc.)</td>
<td>20 (62.5)</td>
</tr>
<tr>
<td>I do not want to share my information with another party (i.e. company or institution)</td>
<td>Strongly disagree</td>
<td>0  (0.0)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>1  (3.1)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>3  (9.4)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>10 (31.3)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>18 (56.3)</td>
</tr>
<tr>
<td>I do not want my information stored online</td>
<td>Strongly disagree</td>
<td>0  (0.0)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>1  (3.1)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>8  (25.0)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>13 (40.6)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>10 (31.3)</td>
</tr>
<tr>
<td>I prefer my style of tracking and monitoring finance information over any other online/mobile tool I have tried or seen</td>
<td>Strongly disagree</td>
<td>2  (6.3)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>6  (18.9)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>10 (31.3)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>7  (22.9)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>6  (18.9)</td>
</tr>
<tr>
<td>Do you share this information with your spouse or partner?</td>
<td>Yes</td>
<td>18 (56.3)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>6  (18.9)</td>
</tr>
<tr>
<td></td>
<td>No response</td>
<td>8  (25.0)</td>
</tr>
<tr>
<td>Does a spouse or partner share this information with you?</td>
<td>Yes</td>
<td>15 (46.9)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>8  (25.0)</td>
</tr>
<tr>
<td></td>
<td>No response</td>
<td>9  (28.1)</td>
</tr>
</tbody>
</table>

Table 5 shows that amongst the participants only 40.6% had attempted to or currently use a mobile application or website as part of their financial management strategies. Furthermore 62.5% of the participants were not in favour of utilizing an application or website that stored their data online. This reluctance to make use of available online tools is consistent with findings of other studies [7] and seems to stem from the limitations of the technology as well as the concerns relating to the personal and sensitive nature of the data.
<table>
<thead>
<tr>
<th>Question</th>
<th>Variables</th>
<th>N(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have you tried or use an application or website to manage your finances?</td>
<td>Yes</td>
<td>13 (40.6)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>19 (59.4)</td>
</tr>
<tr>
<td>The application or website was too complicated/hard to use</td>
<td>Strongly disagree</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>7 (21.9)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td>The application or website did not have a good tutorial or walk through</td>
<td>Strongly disagree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>5 (15.4)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>0 (0.0)</td>
</tr>
<tr>
<td>The application or website did not have the tools I wanted</td>
<td>Strongly disagree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td>The application or website did not provide me with enough information</td>
<td>Strongly disagree</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td>(e.g. reports, calendars or graphs)</td>
<td>Disagree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>5 (15.4)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td>I could not share/sync the information between my devices</td>
<td>Strongly disagree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td>I did not want my finance information on the internet</td>
<td>Strongly disagree</td>
<td>2 (13.3)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>0 (0.0)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>7 (46.7)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>5 (33.3)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>1 (5.6)</td>
</tr>
<tr>
<td>I could not share the information with others</td>
<td>Strongly disagree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>0 (0.0)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td>Would you use or try another application or website that gave you the</td>
<td>Yes</td>
<td>12 (37.5)</td>
</tr>
<tr>
<td>option to store them (your finances) on the internet?</td>
<td>No</td>
<td>20 (62.5)</td>
</tr>
</tbody>
</table>
Another dimension explored by the survey was the attitudes of participants towards gamification. Amongst all the participants only 25% had previously encountered the term gamification. Despite this the participants showed a strong appreciation for the mechanisms used for gamification.

Table 6: Gamification of Financial Management

<table>
<thead>
<tr>
<th>Questions</th>
<th>Variables</th>
<th>N(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have you heard of Gamification before?</td>
<td>Yes</td>
<td>8 (25.0)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>24 (75.0)</td>
</tr>
<tr>
<td>Rewards and progression will make me want to keep using the</td>
<td>Strongly disagree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td>application or website</td>
<td>Disagree</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>15 (46.9)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>8 (25.0)</td>
</tr>
<tr>
<td>If these rewards were new features and tools I would use the</td>
<td>Strongly disagree</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td>application or website</td>
<td>Disagree</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>11 (34.4)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>12 (37.5)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td>If these rewards were tutorials and courses I would use the</td>
<td>Strongly disagree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td>application or website</td>
<td>Disagree</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>14 (43.8)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>9 (28.1)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td>If I could disable the rewards but have the option I would</td>
<td>Strongly disagree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td>use the application or website</td>
<td>Disagree</td>
<td>4 (12.5)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>17 (53.1)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>6 (18.8)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>2 (6.3)</td>
</tr>
</tbody>
</table>

Over 71% of the participants felt that rewards and progression would influence their motivation for using an application or website with 50% of participants favouring rewards in the form of new features and tools.

The final dimension explored by the survey focused on the collaborative aspects of household budgeting. Over 34.3% of the participants showed an interest in using collaborative tools where their partners could contribute towards the rewards suggesting that collaborative gamified approaches may be able to further motivate household financial management and budgeting.
Table 7: Gamification Collaboration

<table>
<thead>
<tr>
<th>Question</th>
<th>Variables</th>
<th>N(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you allowed your spouse or partner access to the application or website, I would use the application or website more?</td>
<td>Strongly disagree</td>
<td>5 (15.6)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>1 (3.1)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>8 (25.0)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>9 (28.1)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td>If your spouse or partner contributed towards rewards I would use the application or website more</td>
<td>Strongly disagree</td>
<td>3 (9.4)</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>2 (6.3)</td>
</tr>
<tr>
<td></td>
<td>Neither agree nor disagree</td>
<td>9 (28.1)</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>8 (25.0)</td>
</tr>
<tr>
<td></td>
<td>Strongly agree</td>
<td>3 (9.4)</td>
</tr>
</tbody>
</table>

4.0 Conclusion and future directions

Financial management remains a challenging proposition for many households. The findings of this study suggest that households continue to rely on trivial strategies such as mental budgeting techniques which are prone to errors and omissions. Whilst households are not fully satisfied with their current strategies, there exists apprehension towards the adoption of more sophisticated or robust techniques. This partly stems from the perceived limitations and distrust of current technology in handling sensitive personal data. But it also stems from a lack of motivation for engaging in a potentially consuming process with perceivably few rewards. Gamification can make the process of budgeting more gratifying by attempting to induce fun into the process. Mechanisms that provide rewards, progression, autonomy and collaboration can enhance the experience of household budgeting and can be strong motivators for change.

The findings of this study are consistent with existing studies [4, 5, 9] which suggest that gamification has the potential to motivate household financial management. As future work this study proposes to investigate the application of a gamified budgeting application for household budgeting.

4.0 References

16 Butt P, Students’ Perceptions of Game-Based Learning, proceedings of ICICTE 2016, pp. 151-158, Rhodes, Greece, 2016.
Proposing a Weighting Function for Adjusting the Global Information Technology Report Networked Readiness Index Framework

Peter K. Oriogun
Lead City University, Ibadan
Department of Computer Science
Oyo State, Nigeria
p.oriogun@lcu.edu.ng, p.oriogun@gmail.com

Abstract

The Global Information Technology Report (GITR) has since 2002 been publishing the Networked Readiness Index (NRI) ratings for a number of countries under the auspices of the World Economic Forum. A number of authors have suggested that the credibility of the NRI is called into question by the non-transparent manner in which the authors report the sources of the data and the methodology that was followed to collect the raw data. Furthermore, that it is clear that there is no fixed formula for the economic policy that suit every individual country, but various widespread procedures normally share some common characteristics. This paper offers a weighting function for adjusting the current NRI final computation based on existing framework. The author claims that computing of the NRI rankings based on this new improved weighting function will minimize the so called ‘digital divide’ alluded to in this paper. It is argued that computing of the NRI rankings based on the author’s proposed weighting function would be more acceptable to the NRI community, by adjusting the current computed final NRI ratings for the benefit of all the economies deemed capable of being members of the GITR NRI community.

Keywords: NRI (Networked Readiness Index), GITR (Global Information Technology Report), WEF (World Economic Forum), ICTs (Information and Communication Technologies)

1.0 Introduction

In this paper, the author having studied the World Economic Forum’s Global Information Technology Reports (GITR) on Networked Readiness Index (NRI) from its inception (2001 – 2002) to date (2016), is of the opinion that the reporting of sub-Saharan Africa countries in particular has not been particularly encouraging. This 2001 – 2002 report claims that 75 countries in the first edition represents more than 80% of the World’s population and more than 90% of its economic output [1].
This first report was based on the original NRI framework by the Center for International Development (CID) at Harvard University, which concentrated on ‘Network use’ and ‘Enabling factors’. The 2002 – 2003 report [2] cautioned that the 82 countries considered in the NRI analysis has limitations due to availability of data from reliable sources. They strongly argued that ranking other countries in future will possibly pose a serious challenge, and suggest that ‘any overall rankings should be done with this taken into consideration’. However, [3] explained that, the 102 countries involved in the 2003 – 2004 report limits the number of variables that can be considered because the methodology adopted imposes a 65% observation rate for each variable over the 102 countries, consequently, variables with fewer observations have been removed. The impetus for this paper stems from these cautionary notes from learned authors of different Global Information Technology Reports on the NRI data to date. Consequently, the author of this paper is suggesting a ways to improve the NRI rankings of all the countries involved in general, based on a new improved formula that is predicated on current model with some adjustments as explained in this paper through the author’s proposed weighting function to be used in conjunction with the NRI ranking.

2.0 The Global Information Technology Networked Readiness Index (NRI)

Initially, a country’s NRI was defined to be the degree to which a community is prepared to participate in the Networked world, however, in the 2001-2002 report, this definition was modified to include the community’s potential to participate in the Networked World in the future. In the same paper, it was pointed out that a single measure such as the NRI is too restrictive and limited in terms of understanding how a country’s national environment affect the adoption of Information Communication Technologies (ICTs). The 2002 – 2003 report further refined the NRI definition such that individuals, businesses, and governments are stakeholders within the community by including the potential and preparation of a community within its immediate environment. The 2003 – 2004 report claims that NRI is a community’s degree of preparation to participate in and benefit from Information Communication Technology development. Another GITR author, Mia (2006) suggest that the NRI measures the tendency for a nations/economies to take a competitive advantage of the opportunities offered by ICT and establishes a broad international framework formulating the enabling factors of such capacity. In the 2013 report [4] argued that in order to make any marked impact on ICT readiness, access and usage is of highest priority for developing economies given the need to narrow the so called ’digital divide’ Bilbao-Osorio (p.5).

3.0 Analysis of Recently Published NRI Data of 6 Sub-Saharan Africa Countries

In [5] statistical analysis of 6 sub-Saharan Africa countries on the basis of continuous availability of the Networked Readiness Index (NRI) data for these
economies since 2003. The author noted the huge difference in the NRI data in the Global Information Technology Report of 2005 and 2006 respectively compared to the rest of the reports to date; consequently the author relied heavily on the NRI data of the past 9 consecutive years (2007 to 2015) as the basis for the prediction of the next 9 consecutive years (2016 to 2024). It is evident from the author’s analysis of recently published NRI Data for 6 sub-Saharan Africa countries that the current framework and analysis of the Global Information Technology Report NRI rankings published by the World Economic Forum needs to be adjusted with what the author is referring to as the Oriogun NRI Weighting Function in this paper.

For Botswana, in Figure 1 and Figure 2 the author observe the marginal difference in the regression equation fitted for the 2016 – 2024 prediction compared to when the author used the actual NRI ranking data for 2016 to predict the next 9 years, 2017 - 2025 (predicated on the previous 9 years 2008 – 2016 inclusive).

\[ Y = 3.53111 - 0.024167X \]
\[ R^2 = 0.470428 \]

\[ Y = 3.524444 - 0.02383X \]
\[ R^2 = 0.45292 \]

Figure 1: Botswana 2016 – 2024

Figure 2: Botswana 2017 – 2025

For Mauritius in Figure 3 and Figure 4 we observe the marginal difference in the regression equation fitted for the 2016 – 2024 prediction compared to when the author used the actual NRI ranking data for 2016 to predict the next 9 years 2017 – 2025 (predicated on the previous 9 years 2008 – 2016 inclusive).

\[ Y = 4.09 - 0.064X \]
\[ R^2 = 0.750183 \]

\[ Y = 4.148889 - 0.066333X \]
\[ R^2 = 76726 \]

Figure 3: Mauritius 2016 – 2024

Figure 4: Mauritius 2017 – 2025
\begin{align*}
Y &= 3.397778 - 0.012388X \\
R^2 &= 0.0165
\end{align*}

\begin{align*}
Y &= 3.433333 + 0.012833X \\
R^2 &= 0.12676
\end{align*}

For Namibia in Figure 5 and Figure 6 the author observe the marginal difference in the regression equation fitted for the 2016 – 2024 prediction compared to when the author used the actual NRI ranking data for 2016 to predict the next 9 years 2017 – 2025 (predicated on the previous 9 years 2008 – 2016 inclusive).

\begin{align*}
Y &= 3.285556 - 0.009X \\
R^2 &= 0.105144
\end{align*}

\begin{align*}
Y &= 3.282222 - 0.019333X \\
R^2 &= 0.12676
\end{align*}

Figure 5: Namibia 2016 – 2024

Figure 6: Namibia 2017 – 2025

For Nigeria in Figure 7 and Figure 8 the author observe the marginal difference in the regression equation fitted for the 2016 – 2024 prediction compared to when the author used the actual NRI ranking data for 2016 to predict the next 9 years 2017 – 2025 (predicated on the previous 9 years 2008 – 2016 inclusive).

\begin{align*}
Y &= 3.285556 - 0.009X \\
R^2 &= 0.105144
\end{align*}

\begin{align*}
Y &= 3.282222 - 0.019333X \\
R^2 &= 0.12676
\end{align*}

Figure 7: Nigeria 2016 – 2024

Figure 8: Nigeria 2017 – 2025
For South Africa in Figure 9 and Figure 10 the author observe the marginal difference in the regression equation fitted for the 2016 – 2024 prediction compared to when the author used the actual NRI ranking data for 2016 to predict the next 9 years 2017 – 2025 (predicated on the previous 9 years 2008 – 2016 inclusive).

For Zimbabwe in Figure 11 and Figure 12 the author observe the marginal difference in the regression equation fitted for the 2016 – 2024 prediction compared to when the author used the actual NRI ranking data for 2016 to predict the next 9 years 2017 – 2025 (predicated on the previous 9 years 2008 – 2016 inclusive).
4.0 Transparency of GITR NRI Data Collection and Methodology

According to [6] a number of extraneous variables have been included that do not shed any light on ICT environment, readiness or usage, while others, that may have added greater robustness to the measure, are missing. Additionally, the credibility of the NRI is called into question by the non-transparent manner in which the authors report the sources of the data and the methodology that was followed to collect the raw data. When [7] investigated Thailand’s position in the Networked Readiness Index, they concluded that it is clear that there is no fixed formula for the economic policy that suits each individual country, but various widespread procedures normally share some common characteristics (p. 408).

A cautionary note was given in the 2002 – 2003 [2] report stating that: ‘countries ranked together can show very small variation in the index...Additionally small differences in the index may be outside the limits of statistical significance due to the fact that some missing observations were estimated using analytic techniques such as regression and clustering’(5). The same report went further to, explicitly outline the research challenges of computing the Networked Readiness Index as ‘Absence of key usage matrices; selection of countries; ensuring statistical significance; Data Estimation and Calculating the NRI’. They also alluded to the fact that computation of the ‘Networked Readiness is a complex phenomenon, and measuring countries NRI remains a significant challenge, and any framework or model representing NRI is a simplified representation at best, a simplified version of reality’.

It was explained in the 2014 report [8] that from a total of 49 sub-Saharan Africa countries, only 35 were included in the analysis, there was no particular reason supplied to explain the rationale for not including the 14 countries that were excluded (34). The 2015 report captured 143 economies, these economies account for 98.4% of the world’s GDP. In the same report, 5 countries (Benin, Bosnia and Herzegovina, Brunei Darussalam, Ecuador, and Liberia) were omitted, however, they were covered during 2014 GITR NRI report. The reason advanced in the report for the exclusion of these countries was because the survey data for these countries were not available. Sierra Leone was also excluded, although survey data does exist for that country, however, it was deemed that too many data points were missing for other indicators. Tajikistan was reinstated, however, there was no reason for the reinstatement in the report [9].

5. Oriogun NRI Weighting Function

The proposed weighting function is dependent on the current GITR NRI methodology and framework. However, it is assumed that the initial membership
rating will normalize all the anomalies and inefficiencies of the current position. In light of the observations documented in this paper regarding the inadequacies and lack of transparency of the Global Information Technology current NRI rankings due to the restrictive methodology, and, no fixed formula for economic policy that suit every individual country in order to truly promote readiness, access and usage, the author is suggesting a weighting function based on the current GITR methodology, such that any country that is deemed to be in a ‘ready state’ for membership, should be afforded certain privileges as a reward for this recognition. Consequently, the proposed weighting function recognises that the rating for each pillar of the GITR framework is currently set to between 1 and 7.

In the current GITR NRI framework, ‘ready state’ means that no country should score less than 1 in the current framework (although the 2005 and 2006 report differs greatly from this norm –without any reason to support the sudden change). On this note, it is suggested in this paper that if a country is deemed important enough, with some initial basic infrastructure in place to be a member of the GITR NRI community, the minimum rating that should be awarded must be 3.5 out of the maximum of 7 (already 50% of the ratings). It now depends on individual country/economy to prove themselves in terms of moving from 50% rating up to 100%, which we believe should be possible, or at least the opportunity is available. Consequently the proposed weighting function is as follow:

\[
\text{Oriogun NRI Weighting Function} = (3.5 \times \frac{\text{GITR NRI}}{7}) + 3.5
\]

Implementing the Oriogun NRI Weighting Function to existing data sets will truly make the index inclusive and respectable in terms of the position of the ranking of all the countries deemed to be at a ‘ready state’ to join the GITR NRI community will be respectable and reasonable to all concerned. The implementation of Oriogun NRI Weighting Function is shown in Table 2 whilst Table 1 depicts the actual GITR data from 2012 – 2016 inclusive. The relevance of these 10 sub-Saharan Africa countries is because they have all featured in the World Economic Forum Global Information Technology Reports from the third edition of the reports from 2004 to 2016 inclusive.

It is evident that the new improved Oriogun NRI Weighting Function based on the Global Information Technology Report NRI actual data looks more respectable on a scale of 1 – 7 ranking. Even though Chad has almost always consistently been at the bottom ranking of all the economies covered by the GITR since inception in 2002. Using this improved Oriogun NRI Weighting Function, its ranking on average over the past 5 years gives the country approximately 67% (4.68) of the maximum NRI rating achievable. If we consider the past 5 year average for Chad using the original GITR NRI ratings, we discover that the Chad would have approximately 34% (2.36) of the overall NRI rating.
Table 1: World Economic Forum GITR NRI (GITR Actual Data)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BOTSWANA</td>
<td>3.58</td>
<td>3.5</td>
<td>3.43</td>
<td>3.4</td>
<td>3.5</td>
</tr>
<tr>
<td>CHAD</td>
<td>2.55</td>
<td>2.53</td>
<td>2.22</td>
<td>2.30</td>
<td>2.2</td>
</tr>
<tr>
<td>KENYA</td>
<td>3.51</td>
<td>3.54</td>
<td>3.71</td>
<td>3.80</td>
<td>3.8</td>
</tr>
<tr>
<td>MALI</td>
<td>2.93</td>
<td>2.97</td>
<td>3.00</td>
<td>3.00</td>
<td>2.9</td>
</tr>
<tr>
<td>MAURITIUS</td>
<td>4.06</td>
<td>4.12</td>
<td>4.31</td>
<td>4.5</td>
<td>4.4</td>
</tr>
<tr>
<td>NAMIBIA</td>
<td>3.35</td>
<td>3.29</td>
<td>3.41</td>
<td>3.5</td>
<td>3.6</td>
</tr>
<tr>
<td>NIGERIA</td>
<td>3.22</td>
<td>3.27</td>
<td>3.31</td>
<td>3.2</td>
<td>3.2</td>
</tr>
<tr>
<td>SOUTH AFRICA</td>
<td>3.87</td>
<td>3.87</td>
<td>3.98</td>
<td>4.0</td>
<td>4.2</td>
</tr>
<tr>
<td>UGANDA</td>
<td>3.35</td>
<td>3.30</td>
<td>3.25</td>
<td>3.20</td>
<td>3.1</td>
</tr>
<tr>
<td>ZIMBABWE</td>
<td>2.94</td>
<td>3.17</td>
<td>3.24</td>
<td>3.1</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2: Oriogun NRI Weighting Function (Based on GITR Actual Data)

In order to have a complete picture of other countries with developed economies than those countries within the sub-Saharan Africa region, it is important to compare the rating of such economies to demonstrate that the disparity does not seem overwhelming when using the improved Oriogun NRI Weighting Function to adjust the World Economic Forum Global Information Technology Report NRI final computation from all the pillars within its framework. Table 3 shows 10 countries from the majority of the developed economies, using the GITR NRI published data, followed by Table 4, showing the ratings of these economies when implemented using Oriogun NRI Weighting Function.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGAPORE</td>
<td>5.86</td>
<td>5.96</td>
<td>6.00</td>
<td>5.97</td>
<td>6.00</td>
</tr>
<tr>
<td>FINLAND</td>
<td>5.81</td>
<td>5.98</td>
<td>6.00</td>
<td>6.04</td>
<td>6.00</td>
</tr>
<tr>
<td>SWEDEN</td>
<td>5.94</td>
<td>5.91</td>
<td>5.80</td>
<td>5.93</td>
<td>5.80</td>
</tr>
<tr>
<td>NETHERLANDS</td>
<td>5.60</td>
<td>5.81</td>
<td>5.80</td>
<td>5.79</td>
<td>5.80</td>
</tr>
<tr>
<td>NORWAY</td>
<td>5.59</td>
<td>5.66</td>
<td>5.80</td>
<td>5.70</td>
<td>5.80</td>
</tr>
<tr>
<td>SWITZALAND</td>
<td>5.61</td>
<td>5.66</td>
<td>5.70</td>
<td>5.62</td>
<td>5.80</td>
</tr>
<tr>
<td>USA</td>
<td>5.56</td>
<td>5.57</td>
<td>5.60</td>
<td>5.61</td>
<td>5.80</td>
</tr>
<tr>
<td>UK</td>
<td>5.50</td>
<td>5.64</td>
<td>5.60</td>
<td>5.54</td>
<td>5.70</td>
</tr>
<tr>
<td>LUXEMBOURG</td>
<td>5.22</td>
<td>5.37</td>
<td>5.60</td>
<td>5.53</td>
<td>5.70</td>
</tr>
<tr>
<td>JAPAN</td>
<td>5.25</td>
<td>5.24</td>
<td>5.60</td>
<td>5.41</td>
<td>5.60</td>
</tr>
</tbody>
</table>

Table 3: World Economic Forum GITR NRI (GITR Actual Data)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGAPORE</td>
<td>6.43</td>
<td>6.48</td>
<td>6.50</td>
<td>6.49</td>
<td>6.50</td>
</tr>
<tr>
<td>FINLAND</td>
<td>6.41</td>
<td>6.49</td>
<td>6.50</td>
<td>6.52</td>
<td>6.50</td>
</tr>
<tr>
<td>SWEDEN</td>
<td>6.47</td>
<td>6.46</td>
<td>6.40</td>
<td>6.47</td>
<td>6.40</td>
</tr>
<tr>
<td>NETHERLANDS</td>
<td>6.30</td>
<td>6.41</td>
<td>6.40</td>
<td>6.40</td>
<td>6.40</td>
</tr>
<tr>
<td>NORWAY</td>
<td>6.30</td>
<td>6.33</td>
<td>6.40</td>
<td>6.35</td>
<td>6.40</td>
</tr>
<tr>
<td>SWITZALAND</td>
<td>6.31</td>
<td>6.33</td>
<td>6.35</td>
<td>6.31</td>
<td>6.40</td>
</tr>
<tr>
<td>USA</td>
<td>6.28</td>
<td>6.29</td>
<td>6.30</td>
<td>6.31</td>
<td>6.40</td>
</tr>
<tr>
<td>UK</td>
<td>6.25</td>
<td>6.32</td>
<td>6.30</td>
<td>6.27</td>
<td>6.35</td>
</tr>
<tr>
<td>LUXEMBOURG</td>
<td>6.11</td>
<td>6.19</td>
<td>6.30</td>
<td>6.27</td>
<td>6.35</td>
</tr>
<tr>
<td>JAPAN</td>
<td>6.13</td>
<td>6.12</td>
<td>6.30</td>
<td>6.21</td>
<td>6.30</td>
</tr>
</tbody>
</table>

Table 4: Oriogun NRI Weighting Function (Based on GITR Actual Data)

It is evident from Tables 3 and 4 that the developed economies are actually competing with each other compared to sub-Saharan Africa economies. The latest (2016) NRI ranking notably had 2 economies, Singapore and Finland in joint 1st position, this was followed by Sweden, Norway, Netherlands, Switzerland and the United States of America in the 2nd position. The United Kingdom and Luxembourg were joint 3rd followed by Japan. In the past 5 years, the top five economies has been Singapore, Finland, Sweden Netherlands and Norway. Japan has moved from 10th and 9th position in 2012 and 2013 respectively to 4th position in 2014. Japan has over the past 5 years on average been in the 6th position overall in terms of its NRI ranking. This may suggest that Japan’s economy has grown the largest overall over the past 5 years.
Figure 13 shows the graphical comparison of the 10 sub-Saharan Africa economies (GITR NRI data), and the 10 developed economies from 2012 – 2016 inclusive.

**Figure 13:** NRI Comparison of 10 Sub-Saharan Africa Economies with 10 Developed Economies (Based on GITR Data)
6. Conclusion

It appears as clearly demonstrated in Figures 13 and 14 that the ten listed developed economies are already self-sufficient in terms of the key elements of the requirements of Global Information Technology Report on Networked Readiness Index. In other words, it is possible that from the framework adopted for the computation of the GITR NRI, the assumptions made, is such that these developed economies are the baseline economies, as such they are expected to have a particular underlying initial ratings (which has not been formally articulated in the framework), meaning that less developed economies such as sub-Saharan Africa countries are not judged on the same basis. It is the belief of the author, as echoed by those authors quoted in this paper that, the current computation of the NRI data need to be more transparent, and should be subject to scrutiny in terms of the level
to which the data gathered from each country is verified and validated within a specified international standard benchmark in order to have confidence in the reporting of the current NRI rankings. It is claimed in this paper that computing of the NRI rankings based on the improved Oriogun NRI Weighting Function would be more acceptable to the NRI community by adjusting the current computed final NRI ratings for the benefit of all the economies deemed to be capable of being members of the GITR NRI community.

7. References


158
Software Systems Engineering: A Journey to Contemporary Agile and Beyond, Do People Matter?

Raid AlQaisi¹, Eddie Gray¹, Bonnie Steves²

¹School of Engineering and Built Environment, Glasgow Caledonian University, Glasgow, G4 0BA, UK.
raid.alqaisi@gcu.ac.uk, e.gray@gcu.ac.uk

²The Graduate School, Glasgow Caledonian University, Glasgow, G4 0BA, UK.
b.steves@gcu.ac.uk

Abstract

This paper explores the evolutionary journey of the approaches and processes of Software Systems Engineering (SWSE), from the Traditional methods to Contemporary Agile methods and looks at elements and issues of importance for future development. It does this through a comprehensive search of the literature and review of the findings of an empirical study “Echoes from the Field” [1,2], which investigates the common practices and issues within the SWSE industry. Particularly, the paper identifies that the human-element is common across all the different SWSE approaches and processes current and past, and that people and their behaviour can have a significant effect on the success of the SWSE endeavour. In the literature, there is a focus on developing the approaches of SWSE to gain a successful outcome, yet very little focus on modelling the effect of human behaviour on the outcome. To understand the human-element effect in SWSE, established theories of human behaviour from the management and psychology disciplines, such as the Toyota Production System (TPS) or Lean, Knowledge Creation Theory (KCT) and Emotional Intelligence (EI), are adapted. The paper concludes that it is not really the approach or process that is the key to success, but rather it is the people who use these approaches that can make the approach succeed or fail. Finally, one of the authors reflects on his experiences of the changes in SWSE since he attended the first Software Quality Management (SQM) conference 25 years ago.

Keywords: Human-Element, Agile, Lean, Emotional Intelligence (EI), and the Knowledge Creation Theory (KCT).
1.0 Introduction

It is intriguing to view the evolution of Software Systems Engineering (SWSE) over the decades, from its early beginnings in the 1950’s to its current state. At first glance, it appears that the various processes and approaches to software systems development and maintenance are quite different from each other. Each SWSE process has different organisation of the four common basic process activities: specification, development, validation, and evolution, as defined by Sommerville [3]. Yet some of the issues with the different SWSE approaches are common across the different approaches, both past and current.

Such issues are almost constant in their existence across all types of software systems development processes over the past 60 years and are most often human-related. It is recognised in the literature, that the human-element in software development plays an integral part in the success of the software systems development endeavour. SWSE is, after all, a human-centric craft [1]. Cockburn [4] highlights that the fundamental characteristics of “people” have a first-order effect on software development, not a lower-order effect. Such a view on the importance of the human-element in SWSE was noticed, as early as 1971, in Weinberg’s [5] book. What is fascinating is that despite this recognition of the “people” factor, developments of new approaches remain focused on the process, rather than on the people. Sadly, this is the prominent view in the field, continuing to seek the silver bullet solution through the development of process [6]. This paper seeks to understand the effect of the human-element on the success of the software development and answer the question, “Do people really matter?”

Section 2.0 describes the evolution of SWSE approaches and processes over the last 60 years and highlights some of the common features and issues. Section 3.0 tackles how SWSE can be seen as a human-centric activity. Section 4.0 discusses some of the empirical findings of the research project “Echoes from the Field” [1,2] which explores with SWSE practitioners the common practices and issues found within the industry. It describes three key themes relating to people that affect SWSE practice and outcome. To understand the human-related issues, knowledge of the SWSE discipline is brought together with knowledge from other disciplines such as the Japanese management theories of Toyota Production System (TPS) also known as Lean, the Knowledge Creation Theory (KCT) and Emotional Intelligence (EI) theory from psychology [1]. Section 5.0 describes these theories and how they model the effects of human behaviour on people’s creative work.

Insight into the human-element effect in SWSE is drawn from adapting the related theories of TPS, KCT and EI to software systems creation, in Section 6.0. It is here that the question in the paper’s title, “Do people matter?” is answered. A deliberately contradictive viewpoint of the standard Agile vs Traditional approaches case is provided, with the inclusion of the human-element which is the common link in all processes and approaches. The paper finishes with the conclusions in Section 7.0. In a special Reflective Coda (Section 8.0), one of
authors, Mr Eddie Gray, gives his reflections on the commonalities and changes in SWSE since he attended the first Software Quality Management (SQM) conference 25 years ago.

2.0 Software Systems Engineering – Journey of Development

According to the literature, the journey of the development of SWSE began with what is known as the Traditional approaches such as code and fix [7], then moved on to the first published software systems development approach known as the phased concept in 1956 by Benington [8], followed by the well-known Waterfall approach by Royce [9] in 1970.

With the need for delivering complex software systems to meet requirements within time and budget constraints, various processes and approaches evolved such as the Spiral Model [7] in 1988 and the Dynamic Systems Development Methods (DSDM) in 1994 [10,11]. Such processes and approaches evolved to overcome the weaknesses of previous processes/approaches, to cope with the rapid evolution of systems, stakeholder demands, technology, and continuously changing environments [12].

From 2001 new approaches to software development began to appear in the literature, known as the Contemporary Agile approaches, that followed the Manifesto for Agile Software Development of 2001 [13]. Examples include Scrum [14], eXtreme Programming (XP) [15], Lean Software Development [16], and Disciplined Agile Delivery (DAD) [17]. In theory, the use of the Agile approach, enables individuals in the development teams to be empowered, more trusted, more liberated in their task selection, and most importantly, consulted in the software systems development process. Additionally, unnecessary documentation is reduced to the minimum [1,13].

What is interesting is the appearance of various methods before and after 2001 that propose a different view from either Agile or Traditional. For example, the Gilb’s Evolutionary (Evo) [18] software development method was proposed ahead of the Manifesto for Agile of 2001 as early as 1988. It focuses on early delivery of high value to stakeholders by promoting incremental iterative software development with an emphasis on clear requirements and delivering value [1]. Another example, in 2003 was the suggested balance of both Traditional and Agile approaches for a particular project in a particular organisation using a risk management tool [11].

One interesting recent change is the evolution of the Spiral Model into the Incremental Commitment Spiral Model (ICSM) in 2014 [19]. The ICSM design is based on the old spiral model design, but with the inclusion of symbolising systems thinking, best practices, and engineering practices. It covers the full system development lifecycle, starting from the exploration phase to the operation phase.
It has the aim of better integrating hardware, software, and human-interaction to cope with the rapid pace of change and to add value for the system users [1,19].

It can be argued that all processes and approaches have more or less the same principles and aims, i.e. to produce software systems on time, within budget, with minimal errors, that meet the requirements and add value to stakeholders. However, they differ in their operation [1,3,20,21].

Both Traditional and Agile approaches have successes and failures in delivering software systems. For example, both Traditional and Agile have the failures of technical-debt and scope-creep [1,22]. There is no silver bullet [6] solution; however, software systems engineers are advised to search and find the best silver bullet to develop a certain software system in a certain time and context [1].

A review of the literature shows that most, if not all, publications are proponents of certain approaches trying to advocate the use of a particular approach over other approaches. For example, Sutherland [23] in his book about Scrum claims that “The way we work is wrong; this is the solution.” [23, Back cover]. Whereas, AlQaisi [1] argues, along the ideas of Cockburn (2000), … [Software Engineering] SWE approaches generally try to approach software development on an abstract level view at the process level, rather than at the holistic level to consider the wider sociotechnical environment. Even when taking the holistic view, the focus remains on the mechanistic SWE approach as a first order component rather than having the people as a first order component.”[1 p34].

The one common element between all methods is the human-element [1]. As can be noticed from the literature, there is still a great deal of focus on the process/approach view and how to implement it, with no direct consideration, at least explicitly on the human-element dynamics [1]. The following section discusses how SWSE is human-centric.

3.0 Software Systems Engineering is Human-Centric

Software systems engineers are the innovators and creators of the software systems. Software development is human-centric and highly dependent on the human knowledge, innovation, and judgement, and to a large extent is influenced by the psychology and emotions of the engineers [1,2,24]. The findings of the empirical study “Echoes from the Field” [1] indicate that the human-centric view is especially important when using the Agile approach for software systems development. Yet, the literature on SWSE has only limited publications researching into the importance of the human-element in software systems development [1,2,25,26,27].

As early as 1971, Weinberg [5] argues the importance of the psychology of the engineers in software systems development. For example, he describes how social factors cause different levels of performance amongst software systems engineers.
Another example is the effect of the environment on individuals and teams. What is most interesting is that his speculations and ideas are still valid and are true reflections of contemporary SWSE practices and deficiencies [1]. Boehm in 1991[28] discusses how people and their initiative make projects succeed “Good people, with good skills and good judgment, are what make projects work.” [28 p.41]. Cockburn in 2000 [4] characterises software systems engineers (People) as a first order non-linear component in software systems development before the process/approach. He argues that it is not the methodological equation of how to develop software, it is the effect of “People” on methodologies [1,4]. DeMarco and Lister 1988-2013 [25] discuss how major issues in software development are human related, such as false assumptions, poor specification of requirements, and lack of top management support, not technical. Therefore, such issues are complicated and hard to solve [1]. Nevertheless, identifying, acknowledging, and dealing with them increases the chances of success. The human-side appears to matter much more than the technical side for a successful outcome [1,2,4,25].

In 2001 Constantine [26] highlights that change in people’s behaviour is slower than technological change. Consequently, the people side of technology shows almost stable issues compared to the issues of technological change. He argues that projects are still running late, over budget, and not delivering the required software system, where people are still facing the same organisational and cultural challenges. Good software, successfully delivered comes from the people, not from the technical tools. [1,26].

In 2012 Broza [27] discusses the human-side of Agile, and how to build the personal leadership skills of the Agile team. One interesting idea by Broza is his model of the evolution of the 3P’s (Product, People, and Process). He states the focus on the three P’s differs through time. For example, in the early manufacturing theories, the most important P was the process; the second most important P was the product, leaving the people the least important of all. Conversely, in the contemporary Agile software development, the focus has shifted the order of importance of the 3P’s for successful outcome to (People, Product, and Process) [1,27]. The argument is that focusing on the process and product over people does not work in a knowledge based and therefore people based field such as SWSE.

The following section will discuss some of the findings of the empirical study “Echoes from the Field” based on investigation of the current practices in the SWSE industry.

4.0 Brief Discussion of Some Findings from the “Echoes from the Field” Study

This section summarises some related findings from the empirical study “Echoes from the Field”. Further details on the study design and implementation can be
found in [1,2,24]. Figure 1 shows the thematic map of the inductive thematic analysis of the findings.

![Thematic Map](image)

Figure 1: The Thematic Map of the Inductive Thematic Analysis from the Study “Echoes from the Field” [1, p171].

The study [1] showed there is a gap between real-life experience and SWSE theories. Most of the current software development methodologies/processes focus on the technical processes rather than on the people who implement and execute these processes, with the assumption that the processes will always be implemented accurately by the people. Such a view neglects the fact that people are different as well as emotionally and psychologically influenced [1]. Thus, process implementation varies between individuals, teams, and organisations; as well as across time with the same individuals, teams, and organisations. This is an important consideration when implementing and practicing SWSE [1].

In the study [1], software systems engineers describe their experience of practicing SWSE as being members of co-located autonomous teams taking ownership of their own environment. The following three key themes relating to people were identified to affect SWSE practice and outcome: 1) Teams work autonomously 2) Varying depth of the implementation of Agile 3) Apprenticeship style knowledge exchange and training.

Theme 1: **Teams Work Autonomously**. Most of the software systems cannot be delivered by any one person single-handedly. The successful delivery of a software project requires the collaborative group effort of a project team [1]. The
participants in the study preferred to work within an autonomous team with ownership, strong internal bonds and high level of synergy as well as lively group dynamics. Such autonomous teams were suggested to be both more Agile and more productive. Some participants described that at the team level, there are more informal and *ad hoc* communications. Some argued that using Agile as a development approach increased group dynamics, team autonomy, synergy, and team unity; whereas with the old approach, namely Waterfall, the environment felt like everyone was competing against one another [1].

Theme 2: “Varying Depth of the Implementation of Agile”. An interesting aspect of the experience of software systems engineers practicing Agile SWSE, was the confusion between the theory and practice of the Agile development approaches. For instance, some teams would apply Agile close to the theory, whereas others found this challenging.

It is interesting to note that implementing and practicing Agile methods differs between organisations as well. For example, organisation Alpha in the study [1, 2] is an older company that has moved to Agile from an older traditional SWSE approach such as the Waterfall. Whereas, organisation Beta is a young start-up company with slightly less experienced software engineers compared to Alpha. The younger software engineers at Beta started software development with Agile mainly using Scrum. As a result of their different history, Alpha had more issues implementing Agile than Beta did; however, both organisations preferred Agile over other processes/approaches. Both organisations perceived the results of using Agile as a success [1].

Theme 3: “Apprenticeship Style Knowledge Exchange and Training”. According to the participants, autonomous teams were generally based on co-locating within ‘physical proximity’, *i.e.* at the same physical space on the same site [1]. This co-location resulted in a rich interaction, both formal and informal, between people that was mostly face-to-face. Such an environment is conducive to learning and exchanging knowledge between individuals. A related matter pointed out by the participants was the effect of the training budget-cuts on the group learning. Interestingly, software systems engineers and managers found solutions to the situation, by being creative and practising a form of apprenticeship that shares knowledge and experience together. In the example provided by the participants, junior team members shadowed more experienced team members to gain their training and learning [1].

The following section describes theories from related disciplines that can be adapted to improve SWSE practices.

**5.0 Related Disciplines and Adapted Theories**

It has been shown that the human-element of SWSE should be treated as of first order importance when dealing with software development and maintenance
[1,4,5]. Yet, in the SWSE literature, there are no theoretical explanations or justifications of why the human-element potentially helps success or causes failure.

Fortunately, there are well-established theories in the fields of industrial production, management and psychology that can provide greater understanding of the complexity of human behaviour and can help explain the importance of the human-element in SWSE development. What follows is a brief summary of the most relevant theories.

5.1 Toyota Production System (Lean)

Known as Lean or the ‘Toyota Production System’ (TPS) [29], its adoption and use is wide spread around the globe in various industries and businesses. According to Liker [30], the “Toyota Way” has what is called the 4P’s: Philosophy, People/Partners, Process and Problem solving; for organisational excellence and for creating learning enterprise [30,31]. Toyota culture is based on “human systems” that are put in place to infuse its founding principles of excellence, mutual prosperity, and trust with all people inside and outside the company. In brief, people are the heart and soul of the “Toyota Way”. The core of TPS is the relentless pursuit to eliminate waste or “Muda” [29,30]. Toyota invests long term in systems of people, technology, and processes that combine to realise high customer value. TPS can be summarised as a culture of respecting people, continuous improvement (Kaizen) and eliminating waste. It is a system that is designed to provide people with the tools they need to continually improve their work [30]. Toyota is serious about long-term thinking philosophy. The Toyota problem solving process is based on the Deming’s Plan-Do-Study-Act (PDSA) cycle [29,30,32].

5.2 Knowledge Creation Theory

The Knowledge Creation Theory (KCT) by Takeuchi and Nonaka [33] is based on the notion that knowledge is created and can be passed on by establishing trusted autonomous teams with cross-functional areas of expertise. Takeuchi and Nonaka state that knowledge creation should be at the centre of the organisation’s human resources “human-element” strategy. Companies are not machines, but living organisms that have their own fundamental purpose, collective sense of identity, and their own self-knowledge that shapes them [1]. Furthermore, Nonaka [34] describes two types of human knowledge. The first one is the explicit hard systematic objective and formal ‘quantifiable’ type that can be articulated in formal language i.e. codified in the procedures. The second type is the tacit subjective knowledge that is highly subjective and depends on the hunches, insights, and intuitions of the employees in the organisations. The second type is hard to articulate in formal language. Nonaka and Takeuchi [35] state that the interaction between tacit and explicit knowledge is the key dynamic of knowledge creation in business organisations. They argue that organisational knowledge creation is a spiral process in which tacit knowledge interacts with explicit knowledge repeatedly, creating the knowledge spiral.
5.3 Emotional Intelligence

Goleman [36] first fashioned the theory and term of Emotional Intelligence (EI) in his book entitled “Emotional Intelligence: Why It Can Matter More Than IQ”. He claimed that EI ability counted for 66% of the ability needed for all jobs, and 85% for leadership jobs. This concept was supported by research output from the industry [37]. EI can be defined as the ability to recognise emotions; “feelings” of oneself and those of others, for self-motivation, and for managing emotions well in oneself and in handling relationships [36]. EI encompasses personal (individual level) competencies that include self-awareness, self-regulation, and self-motivation; as well as, group (social) competencies that include empathy, people (social) skills, and handling relationships [1,36].

6.0 Understanding the Human-Element Effect in Software Systems Engineering Using Related Theories

In his PhD thesis, AlQaisi [1] took the established theories of human behaviour from other disciplines, namely TPS/Lean, KCT and EI, and adapted them in order to explain and provide a deeper understanding of the effects of the human-element on SWSE.

According to the findings of his study [1], software development practices require a high level of EI as in any other activities involving people. For example, software engineers described their feelings of excitement, motivation, frustration, confusion, and sarcasm; and how these related to practice, environment, the surrounding people, policies, freedom to create, and restrictions [1].

Emotions management and EI theory are important aspects to consider when understanding the effect of human behaviour and interaction in SWSE. SWSE is not an individual effort. It is a collective effort of a team of software engineers working and interacting together to achieve a common goal of developing successful software systems. The participating CEO of organisation Beta commented on how software engineers are very emotional people, the opposite of what people usually think about software engineers [1]. People are different and vary in their interaction and communications dynamics.

Understanding the variety of these human dynamics is vital for successful teams. The people dynamics and interactions are an integral part of the human behaviour theories KCT, TPS, and EI. People, including teams, interact together to create and share knowledge. Software systems development is a creative and a knowledge based field. Therefore understanding the effects of human behaviour on that knowledge creation using the theories of KCT, TPS and EI might explain why the Traditional and Agile approaches have both strengths and weaknesses that are related to the people factor which according to the studies outlined above is shown to be more important than the process factor.
Generally, in the SWSE literature, one typically finds listing of the strengths of the Agile approaches in comparison with listing of the weaknesses of the Traditional approaches. As such, all the traits focus on the process. Accordingly, in Figure 2 it is intentional that the diagram is constructed with an alternative focus, indicating that both Agile and Traditional approaches sometimes succeed and sometimes fail and that Traditional approaches have strengths and Agile approaches also have weaknesses. The key element in identifying routes to success is factoring in the middle Human-Element of Figure 2.

Figure 2: Agile versus Traditional View: Human-Effect within Software Systems Engineering.

Raising the consideration of the contradictive view as in Figure 2 is the intention of this paper. All processes and approaches have failures and successes; however, the common link is the people who create the potential for the success or failure of implementing any software systems development approach. What is interesting is that a simple literature search for Agile software systems project failure did not result in papers discussing such an important matter. Returning to the 3P’s example by Broza [27] in Section 3.0, Agile approaches shifted the order of the 3P’s to focus on people as in (People, Product, Process). However, in practice and in the literature according to the findings of the study [1,2], the focus is still on the process as a first order P. All evidence suggests that focusing on the process and product over people does not work in such a creative and knowledge-based field as SWSE.

7.0 Conclusion

It can be argued that Software Systems Engineering (SWSE) can be considered as an interdisciplinary complex field that is heavily reliant on the human innovation, creativity, and imagination, and affected by the psychology and emotions of the
software engineers [1]. The traditional view in the software engineering field, with software development approaches such as Agile and the Waterfall for example, is focused on the technical side. However, as discussed in [1,2,24] that narrowly focused view should be expanded to encompass the wider sociotechnical environment that includes not only the technical side, but also the human social side. By focusing only on the technical process, the context in which SWSE is practiced is missing. SWSE is human-dependent, and thus is highly reliant on the context [1]. The human-element dimension is central to SWSE and its successful practice.

All approaches and processes can potentially succeed or fail depending on people. It is not about Agile, nor the Traditional approaches, but rather the people who practice and use these approaches. Thus, the success of any approach, regardless of being Agile or Traditional is based on the software systems engineers - “People”. According to the findings of the study “Echoes from the Field” [1], it can be argued that it is important to acknowledge the human-element of SWSE as a first order component of successful practice and development.

Software systems engineers preferred as they described: to be co-located within autonomous teams that take ownership of their own sociotechnical environment and SWSE approach. Once people are trusted and have a sense of ownership, they will do their best to make it happen. So, it is the people who choose the appropriate process to develop a particular system, in a particular time, within a particular context; where the process could be changed across time and context. It is therefore the people’s ability to shape their sociotechnical environment and approach that makes the approach or process successful.

8.0 Reflective Coda by Eddie Gray

I am in a privileged position to have attended not only the 1st International Conference on SQM, in 1993 as a presenter but many more over the 25 years to the present day and can therefore reflect on past and recent trends and identify what has faded away and what has become more “popular”. The cover of the Proceedings of the 1st International Conference on SQM, 1993 pretty much summed up the “fashion” or state-of-the-art in SQM in 1993 with its emphasis on standards for software quality and software process improvement, ranging from ISO 9001 QMS through CMM to TickIT and TQM. Over the years these have all transitioned respectively to ISO 9001:2015, TickITplus and CMMI. How and why has this transition happened?
8.1 Into the Seventh Decade of Software Process Improvement Activity

Solomon says “There is nothing new under the sun”.

Software process improvement is not a new idea or aim. The first reference to improving the programming process appeared in 1951 [38]. To date there has been an intensive 66 year effort to improve the software process.

Various technical innovations have been introduced in the more recent decades, for example CASE tools, various programming paradigms, and formal methods and so on. The most notable innovations were met with great expectations and various ‘experts’ greeting each in turn as the panacea to solve most problems. Although growth in system size and functional complexity have been facilitated, through the years many techniques and tools have been tried and failed to deliver substantial global improvements to the software process. None have lived up to the real expectations. Technology alone then is clearly not enough.

Since the mid-1980s interest in the software process and its improvement has increased substantially, with the most significant contributing factor being the ongoing efforts of the Software Engineering Institute (SEI) at Carnegie-Mellon University, Pittsburgh, USA. Leading figures in this camp, such as Humphrey [39] and Curtis [40], have been directly responsible for increased industry awareness. Bill Curtis [40] of the SEI spoke of the need for PROCESS, PEOPLE and TOOLS (TECHNOLOGY), as three prongs of a concerted software productivity and quality effort.

Based on experience and evidence over the years, the trend has been to recognise more explicitly that IT systems and software systems are sociotechnical systems - they cannot be understood without a sense of the relationship between the social aspects (organisation and people) and the technical aspects (hardware and software) of the system. The social and the technical aspects of a system are inextricably linked. Although this is a strand of systems thinking that dates back to the early 1950s, it is becoming increasingly recognised, as evidenced in the earlier sections of this paper.

This trend is also reflected in the topic of my presentation at the 1st International Conference on SQM, in 1993, the Capability Maturity Model (CMM) which developed into Capability Maturity Model Integration (CMMI) for Development. The current CMMI model is divided into 22 process areas, of which 5 cover organisational aspects and many of the others cover management aspects.

The CMMI started life in 1987 as the Capability Maturity Model (CMM), a project at SEI. The CMM for Software was first published in 1991 and is based on a checklist of critical success factors in software development projects during the late 1970s and early 1980s. Its success led to the development of CMMs for a
variety of subjects beyond software. The proliferation of new models was confusing, so the government funded a two-year project that involved more than 200 industry and academic experts to create a single, extensible framework that integrated systems engineering, software engineering, and product development. The result was CMMI. The most important thing to understand about the CMMI-DEV is that it is a model. It is not a process or a prescription to be followed. It is a set of organisational behaviours that have proven to be of merit in software development and systems engineering. Why use such a model? What is its purpose? And how best should it be used? These are critical questions and are perhaps the most misunderstood issues with CMMI.

Best practices are intrinsically a matter of opinion since there is no known right way to develop or evolve software properly. When people get together and decide on best practice, it could be viewed that what is agreed upon is synthetic, from imagination to a significant extent, and indeed arising from their opinion(s) of software ‘engineering’ state-of-the-art in the 1990s. Perhaps it is not the best practice processes choices that are important here for success, so much as that people get together and agree, and are motivated to implement and make a success of their best practice choices.

9.0 References

4. Cockburn A R, Characterizing People as Non-Linear, First-Order Components in Software Development, the 4th International Multi-Conference on Systems, Cybernetics and Informatics, Orlando, Florida, USA, 2000
A Comparative Study of Cloud Services Use by Prospective IT Professionals in Five Countries

Chetan Sharma Kandel¹, Eleni Berki¹, Yan Zhao¹, Sunil Chaudhary¹, Margaret Ross², Geoff Staples³

¹Faculty of Natural Sciences, University of Tampere, Kanslerinrinne 1, Pinni B, 33014, Finland

²School of Media, Art and Technology, Southampton Solent University, East Park Terrace, Southampton, Hampshire SO14 0RD, UK
Margaret.Ross@solent.ac.uk

³BCS Quality SG Chair, British Computer Society, UK

Abstract

Individuals and organizations utilise the cloud technology and its services in various ways. Cloud-based services are becoming increasingly popular, while there is no adequate knowledge offered for their secure use in the education for future IT professionals. It is important to understand how security and privacy issues are perceived and handled by male/female users and IT professionals of different cultures. The authors aim at presenting and scrutinizing information about cloud services’ use by prospective IT professionals in five countries, namely China, Finland, Greece, Nepal, and the UK. In particular the authors, wanting to find out what are the future IT professionals’ conceptualisations and awareness, collected data from male and female IT students in higher education, who use (or not) cloud services. The authors further illustrate the research findings by proceeding to a comparative analysis considering different perspectives such as: gender, education background, national culture (values and culture), and IT-related knowledge. The final research outcomes reveal attention-grabbing information for future IT professionals’ skills, knowledge, and digital competencies. For the IT professionals and software quality engineering communities the latter comprise a body of realistic knowledge, worthy of note when designing curricula for security technology by accommodating practical and accessible solutions (e.g., cryptography-based cloud security) for developing and enhancing the IT professionals’ role.

Keywords: Cyber-security, Cloud-Based Services, Software Quality, IT Professionals’ Role, National Culture, Gender Issues, Higher Education.

175
1 Introduction and Research Rationale

Cloud services [1, 2] are becoming popular day by day for different purposes such as data storage, application deployment, test and development, data backup and many other purposes which often include very sensitive, private data. Therefore, it is crucial for the cloud users and IT professionals [3] to understand how secure a cloud-based service is; and if it is not secure, what might be the associated security problems [4] and what could be the feasible solutions. For an IT professional, in order to propose any appropriate solution [3, 4], it is important to understand the broad and international usage of the cloud-based services and associated technology.

In this work the authors present the data collected about cloud services’ use by prospective IT professionals in five countries, namely China, Finland, Greece, Nepal, and the UK. Thus, there is illustrated information on the use of cloud services but also the reasons for not using the cloud technology. The information presented here reveals the conceptualisations of a sample of future IT professionals of these five countries about use and security of the cloud applications.

It seems that there is a need to re-design the curricula in higher-levels of education in order to prepare the IT professionals to i) gain more security-related knowledge, ii) act professionally and ethically in the society, and iii) become more familiar with the international and broad cloud-based services. Additionally, cloud technology needs to be exposed so that cyber-security issues and security measures to overcome those issues are integrated into subject-taught knowledge. The research outcomes of this trans-national survey could also be useful for developing and wide-reaching the cloud security related knowledge and cyber-safety applications in every country, considering suitable solutions for different cultures (see [5, 6, 7, 8, 9]).

The paper is organised by first presenting the research rationale, questions and brief description of the research methodology, and then proceeds to data illustration, description and analysis’ comments and observations as far as the current data sets permit. Last, there is a session of conclusive remarks, limitations and future research considerations.

2 Research Questions and Methodology

2.1 Research Questions

In this research study, the authors aim at answering the following questions that are first unfolded in this paper:

- How many prospective IT professionals use cloud services and which are their background profiles?
- What is the percentage of male and female users in every country?
- Which are the reasons of not using cloud-based services and which are the profiles of the citizens who do not use them?

2.2 Research Methodology

A questionnaire was designed in Finland with the participation of all the members of our international research team on cyber-security. To fulfil the designing
objective for conducting surveys in a multi-cultural and multi-national research environment, was as follows: The doctoral and post-doctoral researchers and research assistants from many countries organized several brainstorming online and offline (face to face) sessions to determine the most appropriate and complete set of questions, and suitable ways to present them [10], such as translating the questionnaire when in need. We had a pilot survey and published the results [9] and afterwards we translated and distributed the questionnaire in the five countries we consider herein. This research study is international and ongoing. We are currently collecting more data from the UK and Australia, while data from African countries and North and South America is being planned and expected to take place in the near future.

3 Data Collection and Data Analysis

For this study, we collected survey data through printed and distributed to fill in questionnaires from five countries, three from Europe and two from Asia, respectively: Finland, Greece, UK, China and Nepal. We collected data from higher-levels IT students, who are the future IT professionals in these countries’ educational and socio-technical systems. Our current data set (data used herein) contains the largest numbers of data from Greece (three Universities) and Finland (three Universities) as shown in figure 1.

![Figure 1: Categorization of data sets by country](image)

The current dataset handled here contains 48% collected data from Greece. A further 20% of the data come from Finland, which is the second largest country for the collected data. Nepal is the third country for the data collected contributing with 12% of the data, from one University. We have currently collected equal amount of data (10%) from China (one University) and UK (one University) but more data is on its way from both countries.

The current percentages of male and female participants are shown in figure 2. The percentage of male participants is 71.59% whereas 28.41% is the percentage of female participants.
The following figures illustrate the current findings and a brief text description and analysis in comparison.

Figure 2: Total percentages of male/female participants in the current data sample

We categorized the male and female participants in all those five countries as shown in figure 3. As shown in figure 2, China has the largest percentage of male and the least percentage of female participants with 84.62% of male participants and 15.38% of female participants. Similarly, Nepal has the second largest percentage of male participants (83.87%) and the second least percentage of female participants (16.12%). UK has 76.92% of male participants and 23.07% of female participants whereas Greece has 66.93% of male participants and 33.07% of female participants. Last, Finland has the least percentage of male participants (66.67%) but the largest percentage of female participants (33.33%).

Figure 3: Categorization of data (Male and Female) per country
We collected data from 26 participants from China, out of which 22 participants were male and 4 were female as shown in figure 4, next. As shown in figure 5, among the male participants only 68.18% (15 in number) use cloud whereas 31.82% (7 in number) do not use cloud. Similarly, only 25% (1 in number) of female participants use cloud while the rest of 75% (4 in number) do not use cloud.

In Finland, we collected data from 54 participants consisting of 36 males and 18 females as shown in figure 6, next. Among the male participants, only 91.67% (33 in number) of them use cloud and 8.33% (3 in number) do not use cloud while all the female participants use cloud as shown in figure 7.
We collected data from 127 participants in Greece and the data set contains 85 male participants and 42 female participants as shown in figure 8. Figure 9 shows how many male and female participants use cloud or do not use it. In Greece, 80% (68 in number) of male participants use cloud and 20% (17 in number) of them do not use it, whereas 88.10% (37 in number) of female use cloud and the rest of them do not use it as shown in figure 8.

Figure 8 & Figure 9: Categorization of data collected in Greece

Figure 10 and figure 11 present the data collected in Nepal. A total of 31 participants gave the available data. The data commented here were collected from a set consisting of 26 male participants and 5 female participants. Among male participants 65.38% (17 in number) use cloud and 34.62% (9 in number) do not use cloud, whereas 40% (2 in number) of female participants use cloud and the rest of 60% (3 in number) do not use it, as the data illustration shows in figure 11.

Figure 10 & Figure 11: categorization of data collected in Nepal
Similarly, figure 12 and figure 13 present the data collected in the UK. The data collected in UK was given by 26 valid questionnaire respondents, comprising 20 male and 6 female participants, as shown in figure 12. Among the male participants, only 85% (17 in number) use cloud and the rest 15% (3 in number) do not use it. A notable 100% (6 in number) of the female participants use cloud.

![Figure 12 & Figure 13: categorization of data collected in UK](image)

3.1 Comparison of Male/Female Cloud Users Among Five Countries

As shown next, in figure 14, the male cloud users are more than the female cloud users in China and Nepal (Asian countries), whereas female cloud users are more than male cloud users in Finland, Greece and the UK (European countries). Notably, males who do not use cloud services are more than the percentage of the females (not using cloud) in Greece, whereas the opposite is true for other countries. For instance, the females who do not use cloud services are more than males (not using cloud) in China and Nepal. It is also noticeable that according to our sample all females use cloud technology and services in Greece and the UK.

The survey results also show that there are less IT female students in China and only one woman used cloud services. From a gender perspective, China has the lowest ratio comparing to the other four countries. Although the less women IT student phenomena are not only limited to Chinese students, additional recent research studies [9, 11] acknowledge different reasons. For instance, according to Sheng et al. [12] men are more interested in technology than women and fewer women have technical backgrounds [12, 13]. Other researchers [9, 11, 14] suggest to relate to the phenomenon differently due to the impact of females’ personalities,
meaning that it is important to also consider feelings such as anxiety, fear, worry, frustration, anger, envy, guilt, loneliness, jealousy …

There appear to be different reasons for not using cloud. We specified questions about this aspect in the questionnaire used. Some of these collected reasons for not using cloud in Finland, Greece, Nepal and the UK are plotted in figure 15: Thus, the answer ‘I never needed’ is the most common reason for not using cloud with a 50% percentage, whereas the answer ‘I don’t trust’ is the second most common reason. Similarly, other reasons include ‘I don’t know how to use them’, ‘no access’, ‘I never heard’ and other reasons.

Figure 14: Comparison of cloud users and non-users in five countries

Figure 15: Reasons for not using cloud technology
These reasons have also been categorised based on the countries of the data samples as shown in figure 16.

![Figure 16: Categorization of reasons by country](image)

‘Not having access’ is one of the popular reasons for not using cloud in Greece and Nepal. ‘Never heard’ about cloud is another reason for not using cloud in Nepal. ‘No need’ to use cloud is another reason in Finland, Greece and Nepal. The later has the highest percentage in Greece, where 31.82% of participants selected this reason for not using cloud services. ‘Not trusting’ cloud services is another reason that is mentioned in all countries. Last, ‘not having knowledge’ of using cloud technology is another reason that is apparent in Greece and Nepal.

3.2 Clustering

In order to view the similarities and differences between different countries, we performed K-means clustering taking the numbers of total participants, male participants, female participants, male and female cloud users and numbers of male and female not using cloud in all five countries. We performed clustering for several cycles. When we performed clustering for two clusters, we got one cluster containing only Greece and another cluster containing China, Finland, Nepal and the UK as shown in figure 17. In the second cluster, the distances between those four countries were quite different. The reason behind this distribution (Greece being in a separate cluster) is most probably the difference in the number of participants as Greece has much larger number of participants than other countries.

Future research plans might contain different clusters for analysis and comparison, since there will be more available data from more countries. A comparison, for example, according to Hofstede’s index of cultural awareness [5, 6, 7], between
collectivist and masculine values countries (Greece, Nepal, China) v individualist and feminine (UK, Finland) cultures countries could be possible.

Figure 17: Clusters plot

4. Conclusions, Limitations and Future Research

To carry out this research, we designed a survey using a standard questionnaire. No matter the time and resources limitations, the researchers were able to update the questionnaire on time, according to the discussions and brainstorming findings, and managed to contact the students of the aforementioned and other countries. A pilot questionnaire testing [9] was necessary to enrich the questionnaire with more questions and also made considerable changes in the format and structure of the initial questionnaire.

The participants were BSc and MSc level students majoring in computer science and related disciplines from five countries, of which two are developing countries (i.e. Nepal, China) and the remainder are developed countries (i.e. Finland, Greece, the UK). The countries that are considered here first are just by chance and because of convenience, since the first data arrived from these countries. However, our research team has dealt with cultural issues and software quality management
systems realisation in some of these countries, Greece, Finland and the UK in particular (see e.g. [7, 8, 15]). Notably, it is rather exciting for some researchers to come back to certain research issues and measure change/stability attitudes, if any, in given socio-cultural and geo-physical environments and observe their influence in formal and informal education for future IT professionals.

The authors are currently extending this study to capture a wider international context and cloud services users’ needs. Notwithstanding, larger sample sizes are obviously needed to capture the future IT professionals’ role, knowledge, skills and digital competencies worldwide. More data in the future will allow to compare and contrast these findings, considering the dimensions of national culture according to Hofstede’s research findings [5, 6]. In particular, we plan to consider the following: femininity/masculinity, power distance, collectivism/individualism, and time orientation. A future research question on the rise for future research is: What is the influence of national culture in the use and quality deployment of cloud-based services?

The authors further consider to illustrate the research findings by proceeding to a comparative analysis considering different perspectives such as: gender, education background, national culture (values and culture), and cloud technology relevant knowledge, e.g. computational modelling for trust assumptions and cryptography-based cloud security [16, 17, 18]. The later are important knowledge for prospective IT professionals because this type of knowledge could, in turn, assist in increasing trust in interaction by providing correct data encryption/decryption in cloud services’ use.

In order to have a successful realisation of the above plans a deeper in gender perspective and national culture approach are of paramount importance. These preliminary research outcomes together with the earlier surveys’ results [7, 9, 11] reveal attention-grabbing information for future male/female IT professionals’ skills, knowledge, and competencies in a given societal and geographical domain. Cultural awareness and suitable socio-technical education curricula can lead to IT professionalism and professional success in IT advising and consultancy suggestions. The research findings from an international study on cyber-security should comprise a body of minimum and realistic knowledge for the IT professional community. This is worthy of note when designing curricula of cyber-security technology by accommodating practical and accessible solutions for developing and enhancing the IT professionals’ role.

5. References

1. Amies, Alex; Sluiman, Harm; Tong, Qiang Guo; Liu, Guo Ning (July 2012). “Infrastructure as a Service Cloud Concepts”. Developing and Hosting Applications on the Cloud. IBM Press.

185


Authors Index

AlQaisi R ........................................... 159
Berki E ........................................ 105, 175
Butt P .......................................... 135
Chaudhary S ................................. 105, 175
Comley R ...................................... 77
Cristina A ...................................... 95
Dalcher D ..................................... 13
Domínguez-Mayo FJ ...................... 53
Escalona MJ ................................. 53
Estdale J ...................................... 65
Georgiadou E ............................... 77
Gray E ......................................... 159
Helenius M .................................. 105
Kandel CS ..................................... 175
Li L ............................................. 105
Lines KJ ...................................... 39
Martínez-García A ....................... 53
Mejías M ...................................... 53
Ojukwu D .................................... 123
Olivero MA ................................... 53
Oriogun PK .................................. 147
Paltalidis N .................................. 95
Parra-Calderón CL ....................... 53
Ross M ....................................... 175
Sánchez-Bejines JM .................... 53
Sievenpiper M ............................... 135
Smith IM ..................................... 39
Staples G .................................... 175
Steves B ..................................... 159
Suárez-Bote A ............................. 53
Turner S ..................................... 27
Valtanen J .................................. 105
White A ...................................... 77
Zhao Y ....................................... 175