

Scope Management: 12 Steps for ICT Program Recovery[®]

Carol Dekkers
Quality Plus Technologies, Inc.

Pekka Forselius
4SUM Partners, Inc.

The information and communications technology (ICT) world is “addicted” to dysfunctional behavior and the problem is spreading globally. The sad truth is that the parties in the ICT relationship (the customer and the supplier) are largely co-dependent on a pattern of dysfunction characterized by ineffective communication, fixed price contracts with changing requirements, and eroding trust. This article focuses specifically on the northernSCOPE™ 12-step process for ICT program recovery.

The ICT world’s dependence on dysfunctional behavior—specifically ineffective communication, fixed price contracts with changing requirements, and eroding trust—is devastating. Billions of dollars are being spent on rework and enhancements to (defective) software [1]. The latest CHAOS report says that a staggering two-thirds of ICT projects are deemed failures [2], with schedule extensions, cost overruns, and poor quality software more the norm than the exception to the rule. One can find daily news stories where ICT projects are reportedly over budget by several hundred percent, overdue by years, or cancelled after millions of dollars have already been spent.

Almost a decade ago, software suppliers recognized their role in this problem and started investing in process improvement. Similarly, customers focused on improving their technical knowledge in the hope that they could better direct suppliers to implement the right solution. Professional project management and software process improvement initiatives helped streamline the supplier side of ICT programs and projects. However, the issues at play are systemic and involve customers and suppliers.

Questions of how to improve this state of affairs—in an industry of advanced technology, bright project man-

agers, and leading-edge maturity models—led both Australia and Finland to individually investigate further, with results worthy of attention. Their formalized scope management approaches—southernSCOPE [3] and northernSCOPE [4]—have, within the first few years, reversed the trend of failed projects, posted increased ICT program success, and improved customer/supplier relationships. Both approaches are based on project management best practices

... the cost overruns for projects using the southernSCOPE method were found to be less than 10 percent whereas the industry norm was 84 percent. [9]

This article focuses on the concepts of the more recent northernSCOPE, and the differences between it and southernSCOPE will be noted as appropriate. Additionally, the new job role of a certified Scope Manager (CSM), as established by the European Certification & Qualification Association, is discussed. Scope management is not rocket science; however, managing scope is not a natural byproduct of project management.

Why Scope Management?

Introduced in “A Guide to the Project Management Body of Knowledge” (PMBOK Guide) as a knowledge area, scope management can be more important to project success than any of the other individual knowledge areas. As a case in point, 60–99 percent of all defects latent in production software can be attributed to the requirements phase [10]. While project scope management will not guarantee perfect requirements, the simple act of identifying scope delineates what is within the requirements and what is not.

Scope management effectively addresses five out of six of the most common reasons cited for ICT project cost overruns and uncontrolled project growth [8]:

1. Lack of user input.
2. Incomplete requirements.
3. Changing requirements.
4. Technology incompetence.
5. Unrealistic expectations.

Scope management is critical to successful ICT project completion. northernSCOPE places scope management dead center in the overall PMBOK Guide knowledge areas because it involves and interfaces with all eight areas (as depicted in Figure 1).

The PMBOK Guide definition of project scope management is “... the

“Scope management is central to software development and must be integrated especially in regards to time, cost, quality, and risk management.”

combined with customer-centric scope management, and illustrate the important role of scope management in building more resilient software that can stand the test of time¹.

Both initiatives examine and advance ICT programs through steps that initialize, scope, split into manageable sub-projects (as necessary), quantify size, cost (on the basis of currency per unit size), and manage and deliver through professional ICT scope management. The results of both approaches are profound: Success rates on ICT projects have skyrocketed and cost overruns have plummeted to levels unprecedented in the ICT industry. In 2005, the International Software Benchmarking Standards Group (ISBSG) proclaimed:

Figure 1: *Scope Management of northernSCOPE and the PMBOK Guide Knowledge Areas* [11]



[©] Carol Dekkers and Pekka Forselius. All rights reserved.
[™] northernSCOPE is registered in the U.S. Patent and Trademark Office to FiSMA.

processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully” [12]². Scope management is central to software development and must be integrated especially in regards to time, cost, quality, and risk management. There are no scope changes without possible consequences to schedule, budget, and quality, or increases to the risk level of the project. This is true vice-versa as well: If the schedule or budget must be tightened, it requires changing the scope or quality requirements. Otherwise, one increases the overall risk level of the project, including a reduction in quality of project outcomes [11]. The core process of a software development project is *developing the software*. It is not a management process, but an essential object process to be managed (as depicted in Figure 2).

Scope management is best carried out by an independent and knowledgeable scope manager trained in ICT project management, customer relations (communication), software estimation, requirements elicitation, functional size measurement (FSM), change management, and best practices. As a third party usually hired by the customer, the scope manager is an advocate to both the customer(s) and the supplier(s). The role is similar to a construction inspector/coordinator who provides project oversight, governance, measurement, communication, change management, progress reporting, and experience data collection. The European Certification & Qualification Association formalized the northernSCOPE-based CSM job role in October 2007.

northernSCOPE

The northernSCOPE 12-step approach to professional scope management was developed in the late '90s by the Finnish Software Measurement Association (FiSMA) [4]; several CSMs have been trained in how to utilize the approach, with 4SUM Partners and Quality Plus Technologies currently leading the way in this training [13].

The 12 steps are summarized in Table 1. While there may be multiple customers and/or suppliers (e.g., hardware, software, integration suppliers), the scope manager works with all of those affected.

Step 1: Scope Manager Retained, Customer-Driven High-Level Requirements

The northernSCOPE approach is initiated when the customer or software acquirer (or the software supplier) recognizes

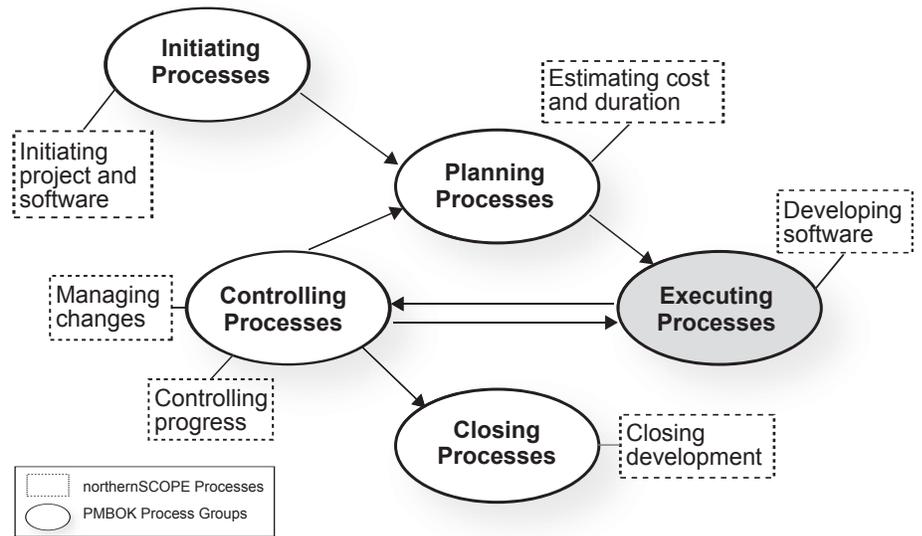


Figure 2: Mapping northernSCOPE Processes against PMBOK Guide Process Groups

the need for, and retains, a scope manager for a new ICT program³. We advocate using a CSM to ensure that there is a basic level of knowledge and experience with northernSCOPE.

The first task is a meeting with the customer (typically the program or project steering committee) to outline the roles and responsibilities of the scope manager, the customer(s), and the software supplier(s). This meeting allows the customer to ask questions and to clarify their role. The scope manager also reviews the high-level customer requirements for completeness and clarity, and discusses their envisaged scope and expectations with the customer.

Step 2: Divide Program Into Subprojects

Using the high-level requirements from step 1 and the program subdivision rules (see Figure 3, next page), the scope manager divides the program of work into appropriate subprojects. Note that this is

similar to subdividing a construction project into distinct subprojects, each of which is typically managed separately and involves unique tasks. This is important to do as early as possible, preferably before beginning one or more software development projects [14].

Figure 3 illustrates the ICT program subdivision rules, while Table 2 (see next page) depicts seven possible (sub)project types. From the number of rules in Figure 3 and the possible combinations, this approach leads to a larger number of small projects. As with any approach, there are a number of pros and cons. One of the biggest pros is improved manageability so important to program/ project success. Neither the customer nor the supplier should resist this process because it divides an amorphous *big bag of work* into identifiable, manageable, and traceable parcels of work that are easier to mutually discuss and scope than is one large monolithic chunk. While not always

Table 1: The 12 Steps of northernSCOPE

Step	Concept	Participants
1	Scope manager retained, customer-driven high-level requirements.	Customer + Scope Manager
2	Divide program into subprojects.	Scope Manager + Customer
3	Scope manager does early FPs for each subproject and estimates total size.	Scope Manager
4	Scope manager and customer determine and analyze quality requirements.	Scope Manager + Customer
5	Customer issues request for proposal.	Customer + Scope Manager input
6	Customer selects supplier based on submitted unit cost per FP.	Customer + Scope Manager input + Supplier
7	Requirements specification developed.	Supplier + Customer
8	Scope manager baselines software size (in FPs) and product development.	Scope Manager
9	Scope manager sizes project changes and cost impact is evaluated.	Scope Manager + Supplier input
10	Scope manager quantifies progress.	Scope Manager
11	Project finishes and customer pays supplier based on FP delivered.	Customer + Supplier
12	Experience data collected and stored.	Scope Manager

1. If the program consists of ICT development and other development work, such as manual process development, re-organizing staff, or technical development, different types of work should be assigned to separate projects.
2. If you apply an incremental or iterative development approach, every increment or iteration should be assigned to separate projects.
3. Different types of ICT development work should be assigned to separate projects.
4. If the program must be stopped consciously for a long time (i.e., to wait for external decisions), the work before and after the break should be assigned to separate projects.
5. If two parts of either product or service development are of a similar ICT project type but differ from each other in the following ways, they should be assigned to separate projects:
 - Development technology
 - Development environment
 - Development team experience
 - Quality requirements of target result
 - Stakeholder dependencies
 - Risk level

Figure 3: *Guidance on Dividing ICT Programs Into Projects and Subprojects* [15]

the case, business projects where software is developed are typically more correctly referred to as programs [12].

Consider an analogy: A buyer of goods requests a fixed price to purchase a series of items—as many as can fit into a single standard-size shopping bag. Without specifically knowing details of the objects they wish to purchase, the constraints are the price (as many objects as possible that can fit), and that the objects must be able to be combined to provide an easy-to-prepare dinner for several people. As such, by discussion and division of what is needed (e.g., cooking utensils, ingredients, food preferences, etc.), both the buyer and the supplier can begin to better scope out the work in advance. Of course, when the supplier(s) are chosen for the work, then the job of planning each piece can be done (the requirements phase in software development) and the contents of the shopping

bag are agreed upon.

This step is a successful component of on-time and on-cost delivery in northernSCOPE⁴. The goal here is to create more manageable and definable parcels of work that can be more readily digested and understood by the business customer; however, it is not the intent to create miniscule projects either.

Table 2 lists seven distinct ICT project types. While there are many additional variations on these seven project types, these were the most common in practice based on the original work done with telecom and software industry companies in Finland. See [15] for further details.

The software development life cycle phases—such as requirements through to implementation—are not considered as being independent ICT project types, but rather as phases within each subproject itself.

Table 2: *ICT Project Types* [15]

ICT (sub)Project Type Classification	Description
1. Customer-specific new development project	Creates a completely new customer-specific piece of software.
2. Software product new development project	Creates a new software product. A software product is always developed to be used by more than one customer. A software product may either be standalone-packaged software or an embedded part of another product.
3. Software version enhancement project	Creates a new version of existing software. The existing software may be either customer-specific software or a software product.
4. ICT service development project	Creates a contract-based continuous or temporary ICT service. The service may be, for example, either software or hardware related, and consists of maintenance, support, help desk, or operating service.
5. Package software configuration project	The result of this project is an installed, parameterized, and user-configured software package.
6. Data conversion project	A project where data is moved from persistent data storage of one information system to persistent data storage of another information system. The software developed in a data conversion project is often <i>throw away</i> in that it is only used once. Even so, the pieces of conversion software may reside on one or more hardware platforms.
7. Software integration development project	Creates software that provides interface services between two or more information systems.

Step 3: Scope Manager Does Early Function Points (FPs) for Each Subproject and Estimates Total Size

Using the high-level customer requirements for each subproject from step 2, the scope manager performs an initial function-size estimate (where appropriate). Note that the function size measures only the functionality of the software; equally important are the quality and technical requirements (as outlined in step 4). The function size of the program software is the sum of all of its subprojects. Currently, there are five different ISO/IEC standard methods for FSM. The most commonly used method within the northernSCOPE concept is FiSMA 1.1 (ISO/IEC 29881), but all the other FSM methods can be equally applied. However, it is important to specify the used method when publishing the measurement results as the same method should be applied throughout the program. All five FSM standards are detailed in [16].

Depending on the completeness of the functional user requirements (remember that this is prior to a supplier engagement), this estimated function size gives the scope manager and the customer a *ballpark* idea of how big the ICT program could be. If the functional requirements are too vague, then the customer must work to at least identify the business processes to be supported by the software. If a customer cannot state their needs, then a supplier cannot provide an appropriate solution—especially before a request for proposal (to engage a supplier to do work) is issued. Even when a supplier is contracted to perform the requirements elicitation work through to software development, a customer may not be able to fully articulate their needs without the assistance of such suppliers. However, it is always up to the customer to make the final decision about what they need and what they are willing to pay to satisfy those needs.

Several subproject types (and also work such as support and fixes) are inappropriate for sizing with FSM. These include technical upgrades, maintenance work, etc. FPs are a square foot type of measurement used to quantify the functional user requirements for software; without them, the work effort must be estimated using another method, such as an hourly rate or historically based estimate. It is important to communicate to the customer about which subprojects can and cannot be sized using FSM.

Step 4: Scope Manager and Customer Determine and Analyze Quality Requirements

This step is unique to northernSCOPE and examines the quality requirements for

each subproject based on the ISO/IEC 9126 quality model. It is well-established that the non-functional requirements for software can dramatically increase the work effort and cost of software development; however, these requirements are often not identified until too late in the project to respond effectively. According to Barry Boehm: “A tiny change in NFR [non-functional requirements] can cause a huge change in the cost.” [17]. He went on to cite the tripling of a \$10 million project to \$30 million when the response time (of an NFR) went from four seconds to one.

Step 5: Customer Issues Request for Proposal

In this step, the customer—with the assistance of the scope manager—prepares the request for proposal and evaluation criteria for the program and then issues it to a set of software suppliers. The *laundry lists* of the high-level functional and quality requirements (apportioned by subproject) are included as attachments to the request for proposal in order for suppliers to submit unit pricing estimates.

Step 6: Customer Selects Supplier Based on Submitted Unit Cost per FP

The scope manager assists the customer to evaluate the supplier responses and provides high-level input regarding the validity of their rates (based on industry rates). The customer selects one or more suppliers to meet the program needs. More specifically—for those parts of the work where the development or enhancement of software functions are involved—the pricing is provided by the supplier(s) in dollars per FP. For other parts, such as migration or operational upgrades, the pricing is provided in dollars per hour (or other appropriate units). The scope manager can provide a reasonability gauge for the cost per FP values by comparing the submitted rates using published project delivery rates in hours per FP (converted to dollars per hour).

Step 7: Requirements Specification Developed

This is the first step in northernSCOPE where the scope manager does not play an active part. The customer works directly with the supplier(s) to develop and *flesh out* the requirements for the subprojects.

Step 8: Scope Manager Baselines Software Size (in FPs) and Product Development

The requirements documents for those subprojects for which functional size is appropriate are the input for this step. The

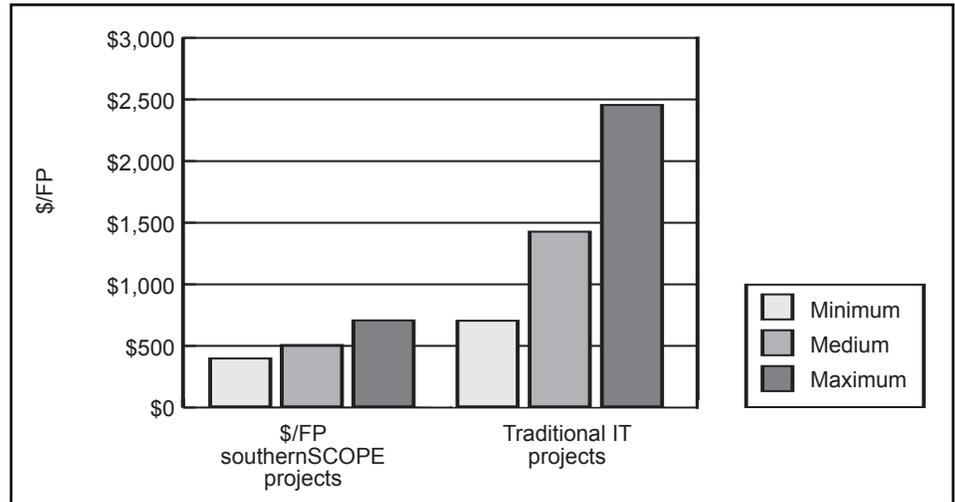


Figure 4: Cost of southernSCOPE Projects Compared to Traditional IT Projects [15]

scope manager reviews the requirements and measures the baseline functional size in FP for each. This step is similar to finalizing the size of a floor plan for a building³, and becomes the base against which any changes and progress are tracked.

Step 9: Scope Manager Sizes Project Changes and Cost Impact Is Evaluated

As changes are proposed for the project—by either the customer or the supplier in agreement with the customer—the scope manager collects and then records the data. A cost estimate is made of the cost and schedule impact of such changes at the point in the project where they are proposed and accepted (based on the unit cost[s] originally quoted by the supplier). A formal change management process facilitates this step.

Step 10: Scope Manager Quantifies Progress⁴

As each project progresses, the scope manager receives documentation from the supplier(s). Using the baseline(s) from step 8, the scope manager records the progress and prepares a formal status report for the customer. Usually this occurs on a monthly basis or on a sprint basis in Agile development, but the exact time frame of this reporting is established with the parties after step 7.

Step 11: Project Finishes and Customer Pays Supplier Based on FP Delivered

This is the second and final step of northernSCOPE where the scope manager is not directly involved. The customer pays the supplier(s) based on the FP delivered for each subproject and on whatever other mechanism where units of payment are used for the non-FP countable sub-

project(s). From the customer and supplier points of view, the program of work is now complete and the project manager closes the project(s).

Step 12: Experience Data Collected and Stored⁴

The scope manager finalizes the project by recording the data collected during and at completion of the project(s). Actual values for work effort and related project variables are recorded along with relevant project attributes.

It is worth noting that there are several concepts introduced in the scope management processes that are not traditionally included in ICT projects. These include:

- Analysis and classification of requirements into independently managed projects (or subprojects).
- Functional size measurement of the software requirements document and project scope.
- Baselining the project metrics.
- Estimating the project effort, duration, and cost based on historical *project actual* values.
- A feedback loop to estimate—and then incorporate and track—accepted changes into the existing project documents.

Through careful attention to project scoping and its management throughout the project, customers and suppliers alike can better specify, build, and acquire quality software products. After the completion of the entire project (i.e., after all steps of southernSCOPE are completed), then the results are tallied and the overall project is compared to those projects where no scope management was involved. Since the introduction of the concept, practitioners involved in southernSCOPE report substantially lowered costs per software functional size (Figure 4).

COMING EVENTS

March 1-4

Ground System Architecture Workshop
Manhattan Beach, CA
<http://csse.usc.edu/gsaw>

March 3-5

Secure IT Conference 2010
Los Angeles, CA
www.secureitconf.com

March 9-12

23rd Annual IEEE-CS Conference on Software Engineering Education and Training
Pittsburgh, PA
<http://cseet2010.dei.uc.pt/index>

March 23-24

GovSec/U.S. Law 2010
The Government Security Expo & Conference/U.S. Law Conference & Exposition
Washington, D.C.
<http://govsecinfo.com/events/govsec-2010>

April 12-14

7th Annual International Conference on Information Technology
Las Vegas, NV
www.cis.gvsu.edu/~nandigaj/SoftwareEngineeringEducationTrack.html

April 26-29

22nd Annual Systems and Software Technology Conference



Salt Lake City, UT
www.sstc-online.org

April 27-30

MARTS 2010:
The Capacity Assurance Conference
Chicago, IL
www.martsconference.com

COMING EVENTS: Please submit coming events that are of interest to our readers at least 90 days before registration. E-mail announcements to: marek.steed.ctr@hill.af.mil.

How to Apply Solid Scope Management for Success on ICT Projects

The five processes involved in the FiSMA Scope Management concept (Figure 2) are integrated concepts; however, all steps are not mandatory on every ICT project. The initiation and estimation steps are prerequisites for the other three components, and these subsequent steps are independent of each other. FiSMA recommends that organizations at least examine the last three steps for their applicability, but understands exceptions and lighter application needs exist [11]. Organizations that will benefit most from northernSCOPE are those representing business areas such as banking, insurance, and public administration because they are routinely involved in software acquisition and procurement. In these organizations, a core business is information management, and the business development centers on developing information systems and software. As such, traditional project management practices have proven to be insufficient for ongoing project governance because scope management is not considered until there is a project already underway. Even then, the project manager has more critical tasks to manage than those related to scope.

In addition to software acquirers, professional software suppliers are in need of organizational-level processes to support continuous process improvement and organizational learning. All supplier organizations could benefit from applying the northernSCOPE processes at both the project and organizational levels. Can every company gain from implementing this concept? Our experience bears out that there are some small- and medium-sized suppliers whose process maturity is considered to be ad-hoc or initial, and would be better served by first concentrating on developing or improving core processes such as time recording and invoicing. Nonetheless, the division of projects into software projects to be independently worked can assist even the most disorganized or immature organizations in improving ICT project management.

Through international collaboration and consulting, we have found that northernSCOPE (and for that matter, software process improvement) just doesn't always succeed even with the best of intentions. Reasons for failure or partial adoption of sound ideas such as

scope management can run the gamut between lack of support or understanding to internal sabotage by organizations weary from *whiplash* changes imposed by management. It is useful to consider the critical success factors of process improvement before launching new concepts. Sources such as [18, 19, 20, 21] outline these critical success factors.

Conclusion

Through the results of northernSCOPE and southernSCOPE, scope management processes are a proven means of leveraging and augmenting professional project management on ICT projects. With the current levels of project rework in the vicinity of 45 percent of development effort, our industry surely needs to increase its ICT project success, and one proven way was expounded on in this article.

ICT program recovery can greatly benefit from a solid 12-step program, starting with an admittance of the facts, determination, trust, continuous control, and professional guidance and support from a scope manager. ♦

References

1. Christensen, David S. "An Analysis of Cost Overruns on Defense Acquisition Contracts." *Project Management Journal* 3 (Sept. 1993): 43-48.
2. The Standish Group. "New Standish Group report shows more project failing and less successful projects." Standish Newsroom – Chaos 2009. 23 Apr. 2009 <www.standishgroup.com/newsroom/chaos_2009.php>.
3. eGovernment Resource Centre – State Government of Victoria (Australia). "southernSCOPE – Avoiding Software Budget Blowouts." 6 July 2007 <www.egov.vic.gov.au/index.php?env=-innews/detail:m1816-1-1-8-s-0:n-832-1-0-->>.
4. FiSMA. *For Better Management*. northernSCOPE brochure. 8 Apr. 2008 <www.fisma.fi/in-english>.
5. Dekkers, Carol, and Pekka Forselius. *Increase ICT Project Success with Concrete Scope Management*. Proc. of the PMI Global Congress 2007 – Asia Pacific. Hong Kong. 29-31 Jan. 2007 <www.compaid.com/caiinternet/ezine/for_selsius-scope.pdf>.
6. Forselius, Pekka, and Risto Nevalainen. *The Role of SCOPE Manager as Link Between Engineering and Management Processes*. Proc. of the EuroSPI 2002. Nuremberg, Germany.
7. Morris, Pam. *Metrics Based Project Governance*. Proc. of IWSM/MetriKon.

Software Defense Application

The Standish Group's 2009 CHAOS report [2] states that software project success—meaning an on-time, on-budget, and a fully functioning product—has slipped to 32 percent. This, coupled with the abundance of sourced software development and enhancement projects in the DoD, should signal the importance of program scope management to defense software managers. The biggest benefit to the defense community is in increased customer communication and traceability of the original request through the software-intensive systems projects.

- Berlin, Germany. Nov. 2004 <www.compaid.com/caiinternet/ezine/morris-metrics.pdf>.
- Wright, Terry. "Controlling Software Development, A Breakthrough for Business." *Software Quality and Process Improvement*. Melbourne, Australia: Software Engineering Australia. Nov. 2000.
 - ISBSG. *Software Development Projects in Government ... Performance, Practices and Predictions*. 2005 <https://www.ifpug.org/about/SoftwareInGovernment.pdf>.
 - Insight, the Army Software Metrics Newsletter*. Summer 2002.
 - Forselius, Pekka. *Effective Scope Management Concept – A Key to Success*. Helsinki, Finland: FiSMA, Aug. 2003.
 - Project Management Institute. *A Guide to the Project Management Body of Knowledge – PMBOK Guide*. Newtown Square, PA: Project Management Institute, 2000.
 - Dekkers, Carol, and Pekka Forselius. *Certified Scope Manager based on northernSCOPE*. Training brochure. Dec. 2007 <www.qualityplustech.com>.
 - Forselius, Pekka. *Divide et Impera – Learn to Distinguish Project Management From Other Management Levels*. Proc. of the Software Measurement European Forum. Rome, Italy: May 2005.
 - Forselius, Pekka, et al. *Program Management TOOLKIT for Software and Systems Development*. Helsinki: Talentum, 2008.
 - Bundschuh, Manfred, and Carol Dekkers. *The IT Measurement Compendium*. Berlin/Heidelberg: Springer-Verlag, 2008.
 - Boehm, Barry. Preface to *Requirements-Led Project Management – Discovering David's Slingshot* by Suzanne and James Robinson. New York: Addison-Wesley Professional, 2005.
 - Dekkers, Carol. "Unleash the POWER to Improve." *Software Quality Professional* 3.1(Dec. 2000): 48-51.
 - DeMarco, Tom. "Mad About Measurement" essay in *Why Does Software Cost So Much?* New York: Dorset House, 1995.
 - DeMarco, Tom, and Timothy Lister. *Peopware: Productive Project and Teams*. New York: Dorset House, 1999.
 - Willman, Paul. "Organizational Change in the 1990s." Lecture. University of Limerick, Ireland: Oct. 1996.

Notes

- More about successful use of northernSCOPE can also be found in [5] and [6] while more on southern-

- SCOPE can be found in [7] and [8].
- Scope management is put in the middle of Figure 1, though [12] introduces knowledge areas in a different order originally. The order is not important, but here we want to emphasize the central role of scope management in software development.
 - At this point, customers may refer to this as the project; however, it is more

likely an ICT program with several projects. See step 2.

- This step is not a part of southernSCOPE.
- When a floor plan is initialized at the beginning of a home construction project, it becomes the master plan that the built area in square feet or square meters and any changes to the floor plan are measured against.

About the Authors



Carol Dekkers is CEO of Quality Plus Technologies, Inc. She consults clients on how to maximize project success through realistic estimating, software measurement, and scope management. Dekkers holds designations as a Project Management Professional, Certified Management Consultant, Certified FP Specialist, Information Systems Professional, and Professional Engineer (Canada). She is known internationally as an expert, a speaker, and an author in software metrics, functional size measurement, project scope management, and IT globalization, and has served as a visiting scientist with the SEI. Dekkers is the co-author of several books, including: "The IT Measurement Compendium - Estimating and Benchmarking Success with Functional Size Measurement," "Program Management Toolkit for software & systems development," "IT Measurement: Practical Advice from the Experts," and "Practical Project Estimation."

Quality Plus Technologies, Inc.
8430 Egret LN
Seminole, FL 33776
Phone: (727) 393-6048
E-mail: dekkers@
qualityplustech.com



Pekka Forselius is a business partner, CEO, and project management consultant at 4SUM Partners in Finland. Forselius developed the Experience Pro data-collection concept and was product manager for Experience Pro software. He is a researcher and developer of project management methods and concepts, including northernSCOPE Scope Management and the FiSMA 1.1 ISO/IEC 29881 Functional Size Measurement standard. Since 2000, he has been the primary representative of the national body of Finland to the ISO/IEC JTC1/SC7 standardization working group WG12, Functional Size Measurement. Forselius is also the president of the international benchmarking organization (ISBSG). He is co-author of several books including "Program Management Toolkit for software & systems development," "Practical Project Estimation," and "Applied Statistics for Software Managers." Forselius holds an executive MBA and a master's degree in informatics.

4SUM Partners, Inc.
Tekniikantie 14
Espoo, Finland FIN-02150
Phone: +35 85 05160416
E-mail: pekka.forselius@
4sumpartners.com