Deploying Software Estimation Technology and Tools: The IBM Software Solutions Toronto Lab Experience

Nikki Panlilio-Yap and Danny Ho

Abstract

Developing a software product involves estimating various project parameters. This is typically done in the planning stages of the project when there is much uncertainty and very little information. Coming up with accurate estimates of effort, cost, schedule, and reliability is a critical problem faced by all software project managers. The use of estimation models and commercially available tools in conjunction with the best bottom-up estimates of software-development experts enhances the ability of a product development group to derive reasonable estimates of important project parameters.

This paper describes the experience of the IBM* Software Solutions (SWS) Toronto Laboratory in selecting software estimation models and tools and deploying their use to the laboratory's product development groups. The combined efforts of the Software Engineering Process Group (SEPG) and the Tools and Technology Group (TTG) are presented and explained. The SLIM* and COSTAR* products, the software estimation tools selected for deployment to the product areas, are introduced, along with the rationale for their selection. This paper also describes mechanisms used for technology injection and tool deployment, and discusses the important lessons learned in the technology and tool insertion process.

1.0 Introduction

Developing a software product involves estimating project parameters such as effort, cost, duration, and reliability. Estimates are crucial to developing the project schedule and allocating the necessary staff and resources. Estimating is typically done in the planning stages of the project when there is much uncertainty and very little information. Nonetheless, estimation is very important to software development since it forms the basis for project planning and management. It is a cross life-cycle discipline that applies to all phases of the development life cycle. During the course of running the project, constant re-estimation is vital to assess the risks at various stages of the project. In some situations, the estimates have to be revised and the project has to be rescheduled.

*IBM, AS/400, OS/2, AIX and BookManager are registered trademarks of International Business Machines Corporation.
SLIM is a registered trademark of Quantitative Software Management, Inc.
COSTAR is a trademark of Sofstar Systems.
Checkpoint is a trademark of Software Productivity Research, Inc.
This paper captures the experience of the IBM SWS Toronto Laboratory in deploying software estimation technology and tools, and summarizes the key lessons learned. Topics include

- Tool selection in the early stages
- Technology injection through demonstrations, technical exchange sessions, and lab-wide education
- Availability of tools through the Software Lending Library and the Toronto Lab Common LAN
- Availability of information through the Experience Warehouse, the Window on the World utility, and the Software Lending Library.

Lessons learned from the experience described in this paper can be applied, in general, to the deployment of other technologies and tools.

2.0 Estimation Technology and Tools Deployment

The deployment of software estimation technology and tools in the IBM SWS Toronto Laboratory consisted of three major stages as illustrated in Figure 1. Activities associated with each stage are shown; each stage is described in the following subsections.

2.1 Understanding - The Early Stage

The Software Engineering Institute (SEI) self-assessment conducted by the IBM SWS Toronto Laboratory in 1991 revealed a critical need for software estimation techniques and tools. Probably the best tools for estimation are those that use models based on historical data from one’s own organization or environment [1, 4]. In the absence of an internally developed tool based on historical data from the IBM SWS Toronto Laboratory or from similar IBM laboratories that develop multiple software products across multiple hardware platforms, it is logical and practical to use one or more commercially available estimation tools. Some of these tools have underlying models based on thousands of software development projects from industry. These tools typically use input on the size of the product to be developed, project constraints, characteristics of the development team, complexity of the product, and characteristics of the development environment. The Software Engineering Process Group (SEPG) and the Tools and Technology Group (TTG) combined efforts to find software estimation techniques and tools that would be appropriate to use in the laboratory.

The Tool Evaluation and Introduction Process described in Ho [7] was adopted in conducting pilots and early experiments. At the outset of the process, information was gathered from different sources, such as magazines, technical papers, conference proceedings, etc., to identify potential candidates. Once several promising tools and vendors had been selected, the vendors were requested to send detailed information or demonstration diskettes of the tools for evaluation. Pilot experiments with some software-development projects were also conducted by obtaining trial licenses or borrowing tools available at other IBM Canada Ltd. sites. Eventually, the SLIM and COSTAR tools were formally purchased or licensed.
### Figure 1. Stages of Software Estimation Technology and Tools Deployment

#### Understanding
- Gather preliminary information
- Obtain trial license
- Conduct pilots
- Perform technical assessment and evaluation

#### Installation
- Recommend methodology and tool
- Demo tool: 1-1, group, public forums
- Educate users through technical exchange
- Provide consulting to interested parties

#### Adoption
- Provide broad-based education
- Make tool available through
  - Software Lending Library
  - Common LAN
- Capture information and experience in
  - Experience Warehouse
- Provide lab-wide consulting service

---

#### 2.1.1 Criteria Used in Tool Selection

Several criteria were used to evaluate software estimation tools. Required basic features include the ability to:
- Give accurate estimates
- Perform automatic recalculation whenever some parameters are altered

Break down the estimates into different phases of the development life-cycle
Support different software sizing methods.

Some desirable and advanced features are the ability to:
- Track project actual data
• Conduct re-estimation if needed
• Perform what-if analysis to experiment with different parameters
• Be extensible to include user-specific parameters
• Be adaptable to user-specific development environments.

2.2 Installation - Making the Selected Tools Available

2.2.1 The Selected Tools

The final decision in the choice of software estimation techniques and tools depended on the results of the pilot experiments. Both the SLIM and COSTAR products satisfied the basic requirements and possessed some desirable features for good software estimation tools. Both tools produced good pilot results.

The amount and complexity of input required for these tools is not nearly as cumbersome as that required for some other commercially available tools. In addition, the underlying theory of the SLIM and COCOMO models is well-known and published in the public domain. Two models were adopted because neither one gives 100% accurate estimates. The use of more than one model may make up for some of the shortcomings of each one.

The latest version of the Checkpoint* tool has recently been identified by the Software Solutions Division as a Common Quality Tool for the division, extending the scope of software estimation to project tracking and product benchmarking. This makes one more model and tool available to validate estimates derived from the other tools and the development teams' experts. At the time this paper was written, however, there were insufficient pilot results for the Checkpoint tool, and we shall not mention it further.

2.2.1.1 The COSTAR Tool: The COSTAR tool is a DOS-based estimation product by Softstar Systems [8, 14]. It implements the CONstructive Cost Model (COCOMO) [5]. COSTAR 3.0 is currently deployed in the laboratory. The tool estimates the duration, staffing level, and cost of a project. Line of code (LOC) is used as the basic input. Various components can be refined within the effort equations for estimation, which can be performed in a structured manner using subcomponents. Estimates are provided for two models, intermediate and detailed, with accuracy within 20% of actual values, 70% of the time. The output consists of a development summary and a variety of reports.

2.2.1.2 The SLIM Tool: The SLIM tool is a software product designed to help software managers, financial managers, and analysts estimate the time, effort, and cost required to build medium and large software systems. It is a metrics-based estimation tool developed by Quantitative Software Management, Inc. (QSM) of McLean, Virginia. It embodies the Software Life-cycle Model developed by Putnam [10, 11], using validated data from over 3000 projects from industry. The projects are stratified into nine application categories ranging from microcode to business systems. The category into which most of the IBM SWS Toronto Laboratory products fall is system software.

The tool can be customized to a specific organization. Its project milestones can be specified and historical data can be used for calibration.

The tool has a rich set of what-if capabilities for exploring opportunities to increase delivered product quality and reduce development costs. It generates project plans that optimize resource use. It includes features that enable the assessment of time, effort, and
cost risks. Input to the tool consists primarily of product size estimates, process productivity parameters, and management constraints. After a time-effort-cost solution is selected, the tool generates a set of detailed implementation plans associated with that solution. A more detailed description of the tool and its capabilities can be found in [6], [9], [12], and [13].

2.2.2 Demonstrations and Technical Exchange Sessions

Demonstrations and technical exchange sessions are useful for introducing new technology and tools to the laboratory. During the course of injecting software estimation techniques and tools, the SLIM and COSTAR products were demonstrated on different occasions:

- To individuals in one-on-one sessions
- To development teams in group sessions
- In public forums such as:
  - The annual Centre for Advanced Studies Conference (CASCON) organized by the Centre for Advanced Studies (CAS)
  - The Tools Exposition (TOOLS EXPO) organized by the Tools and Technology Group
  - The IBM Worldwide Software Development Conference (SDC) sponsored by the SWS Director of Quality and the Corporate Software Quality Excellence Council.

These occasions have given some people an increased awareness in the area of software estimation and allowed others to gain in-depth technical knowledge.

2.2.3 Limited Consulting

In addition to the demonstrations and technical exchange sessions in-depth consulting was offered to a number of projects whose personnel showed a commitment to learning and using the selected software estimation techniques and tools. The consultants sat down with project managers, planners, and other key project personnel, and walked them through the software estimation process with the aid of the selected tools. The consultants also provided analysis and interpretation of the estimates and tips on their use.

2.3 Adoption - Expanding the User Base

2.3.1 Broad-Based Education

To increase the penetration of software estimation techniques and tools within the laboratory, the SEPG and the TTG jointly developed a two-day course. Its objectives were to:

- Teach the underlying theories of the SLIM and COCOMO models
- Provide in-depth training on the SLIM and COSTAR tools
- Provide hands-on experimentation with the tools.

The benefits of the locally developed course were:

- The creation of local experts in the product areas who, in turn, spread the knowledge to their development project teams
- The deployment of more than one theory and tool
- Substantial cost saving: compared to bringing a vendor into the laboratory to teach the course
Tailoring of the course content to suit the local development environment.

The Software Estimation and Tools course was a good combination of theory and hands-on exercises. It was structured as follows:

- **DAY 1 - CONCEPTS, COSTAR, AND SLIM**
  - Motivational Introduction
  - Estimation Concepts
  - Lab Direction
  - LAN Access
  - COCOMO/COSTAR
  - SLIM Introduction

- **Day 2 - SLIM**
  - Overview and General Concepts
  - SLIM Theory and Calibration
  - Trade-off Concepts and Limits
  - Wrap-up

Feedback from the course participants was very positive and seemed to improve with each new session offered. This result reinforces the effectiveness of lab-wide education in technology and tool injection.

Appendix A shows a sample abstract for the course as it appeared on the laboratory’s EDUCATE system.

### 2.3.2 Tool Availability

One of the most important tasks in deploying promising tools is to make them available throughout the laboratory. The target users for the SLIM and COSTAR tools are primarily planners, project managers, and team leaders. Since the majority of the laboratory community is LAN-connected, the Toronto Lab Common LAN is utilized to make the tools generally available. The Software Lending Library is used to distribute the tools to non-LAN users.

#### 2.3.2.1 Software Lending Library

The Software Lending Library is a central location for lending software to the laboratory community for experimentation and pilot use. A user who signs out a software package is given two weeks to experiment with the software. When the software is returned to the library, an online survey is sent to the user to gather feedback on the tool.

The Software Lending Library helps the deployment of SLIM and COSTAR significantly. It provides a means for non-LAN connected users to experiment with the tool. The online survey captures valuable tools experience that will benefit the other users within the laboratory and help define the strategy for software estimation techniques and tools in the future.

#### 2.3.2.2 LAN Deployment

The SLIM and COSTAR tools are also available on the Toronto Lab Common LAN, which is shown in Figure 2. Ho [7] provides a detailed description of the LAN configuration. The Common LAN is basically a collection of OS/2 file servers, AIX file servers, and end-user OS/2 and AIX workstations, connected by multiple token rings.

The SLIM and COSTAR tools are first installed on the master OS/2 file server in the backbone ring and then replicated in total on each of the slave OS/2 file servers on the floor ring. End-user workstations access the tools through the slave file servers. A license control mechanism limits the number of users concurrently accessing the tools to the maximum license count. The mechanism also provides a means to electronically invoke the tools in a more automated manner, as opposed to traditional manual software distribution. The Common LAN increases the penetration of SLIM and COSTAR by making the tools easily accessible and available throughout the lab. There is also a cost-
saving benefit since acquisition of individual copies of software for each end-user is avoided. Furthermore, end-users are relieved from the burden of tools upgrade and maintenance.

Quantitative Software Management Inc. and Softstar Systems both have agreements with IBM Canada Limited that permit use of their software on the Common LAN operating environment. Usage is constantly monitored by the Tools Support Group to quickly recognize and correct license exhaustion.

As the direction of tool development switches to true exploitation of the LAN, the tools should make use of the client-server model in data sharing as well as computing. Servers should be used as a repository for both the software and data. Remote LAN data services should be increasingly utilized.

2.3.3 Information Availability

Availability of tools must be accompanied by availability of tool information and ease of access to the information. It is important to document tool information, formal tool evaluation results, pilot results and user feedback, and to keep these documents up-to-date, so that the knowledge can be shared. Tool information is accessible from the Laboratory Experience Warehouse, the Window on the World utility, and the Software Lending Library.

2.3.3.1 Experience Warehouse: The Laboratory Experience Warehouse is the central repository for a wealth of information useful to the software development community. It is the Toronto Laboratory’s version of an Experience Base used to store some forms of packaged experience as described in the Experience Factory concept proposed by Basili and his colleagues [2, 3].

By invoking the Experience Warehouse (EW) utility, one can view sections dedicated to development processes (for example, Program Process Architecture: Plus, PPA+), Laboratory Standards, development tools, etc. The tool section consists of four matrices collecting information on tools under evaluation, under pilot (unsupported), supported (by the Tools Support Group) and rejected (not promoted). The tools within each matrix are grouped by development life-cycle, and the tool documents can be accessed through BookManager* hypertext links. (When a certain tool is selected within the matrix, details of the tool will be displayed.) The information includes some general description of the tool, formal evaluation report, and user feedback.

2.3.3.2 Window on the World (WOW): WOW is an online utility to retrieve information for quick reference. Information for tools under support is kept on WOW. Users can use the names of the tools to quickly retrieve information (for example, WOW SLIM, WOW COSTAR). General information on the tools include general description, operation, licensing constraints, installation, environment constraints, invocation mechanism, support, and license agreement.

2.3.3.3 Software Lending Library The Software Lending Library is another good place to retrieve tool information. Prior to deciding to borrow a tool for experimentation, the user is presented with its description. The user can also view the user feedback collected in the library’s repository. There is also an option for requesting the tool center of competence to contact the user and provide consulting.

The library also carries manuals of the tools accessible through the Common LAN. Although online help is usually available for the tools, the users may choose to borrow
Figure 2. The Toronto Lab Common LAN Architecture (Simplified)
2.3.4 Lab-Wide Consulting

As more and more project groups demonstrated a need, the SEPG and the TTG made their software estimation consulting services available to the laboratory's development community. Because of resource constraints at the laboratory level, most consulting was provided to the project groups through project personnel who had been trained on the use of the software estimation techniques and tools. This allowed the project groups to develop their own local experts. It also allowed the lab-level experts to provide service to more development project groups.

3.0 Level of Deployment and Future Directions

When this paper was written, five sessions of the Software Estimation and Tools course have been offered to the laboratory. Over 70 laboratory personnel covering all major subbusiness areas of the laboratory have been educated on the use of the software estimation tools. Several projects from each subbusiness area have experimented with or used the SLIM and COSTAR tools. Several others have also used the tools indirectly in consultation with the SEPG and the TTG. Client contacts have been established within and outside the laboratory. Five of the seven products submitted by the laboratory to the MDQ Malcolm Baldrige Assessment in 1993 stated the use of estimation models and tools as their initiatives to improve their overall estimation process and the accuracy of their project estimates.

Although the SLIM and COSTAR tools have been successfully deployed, much work still remains. In addition to the technology injection techniques discussed in the earlier sections (for example, demonstrations, lectures), users group meetings should be conducted periodically to update the users on the latest developments or breakthroughs. The group meetings will also provide opportunities for the users to exchange ideas and experience.

Another area that requires immediate attention is the technical assessment, evaluation, and recommendation of size estimation techniques and tools. Size estimates are critical inputs to software estimation models and tools. Other related activities that complement estimation are tracking and project management. The feasibility of integrating software estimation tools with project management tools should also be investigated.

As product development groups switch from the traditional approaches to object-oriented development, the models for software estimation are expected to change accordingly. It is unclear at this moment how the existing software estimation models can be used for object-oriented software development.

4.0 Lessons Learned

The experience we have described is based on over three years of solid work. The process we have followed can be applied in general to the deployment of other techniques and tools. Following are the key lessons learned from this experience.

- Deploying state-of-the-art technology and tools takes time. Users have to overcome a lot of barriers to become knowledgeable in the field. In addition to learning the methodologies and tools, the users have to learn about accessing the tools through the LAN, or installation of the tools (if not LAN connected). In some
situations, users may have to configure, install, or upgrade certain components of the operating system and learn about it prior to using the tools. These are overhead tasks the users must face before any true benefit in adopting the methodologies and tools can be realized.

- **Management commitment is very important.** Management support is needed for both the consultants and clients in terms of time allocation to tackle the overhead tasks, education, cost of software and hardware, etc.

- **The champions must be pro-active and proficient.** They must be in a position to give advice, provide consultation, and offer assistance. It is very important to conduct a thorough analysis, and point out both the strengths and weaknesses of the methodologies and tools to the clients.

- **Broad-based education is highly effective and rewarding.** We strongly encourage the same infrastructure in deploying technology and tools in other areas. Not only does it save the organization some money, but course participants also get more value out of a course taught by local experts using real development data collected within the laboratory, compared to one taught by a tool vendor. Vendor courses tend to teach limited theory and are confined to their product offering.

- **The collection of historical data is critical to process improvement.** There is a crucial need to continuously capture historical data on in-process project parameters. The estimated and actual values of the schedule, resource allocation, defects, etc. should be collected to improve the quality of subsequent estimation. Having this data is critical for calibrating commercially available estimation tools and tuning them to the development environment.

- **Understanding how collected data will be used is essential.** There is resistance on the part of many software developers to capture estimates as well as the actual values of project parameters. They are afraid of how the numbers or measures will be used or misused by management or other groups. It is important to make them understand that the collected data will help managers identify strong points and bottlenecks, and help them set realistic goals for future software development projects.

### 5.0 Acknowledgements

Chris Smith gave wholehearted support and encouragement for our efforts in the deployment of software estimation techniques and tools, and aroused the interest of the laboratory's managers in this key process area. Chris Alford, George Dadoun, and Ross Simpson continued to support our efforts. John Henshaw encouraged the writing of this paper and provided many suggestions for improvement. Ann Gawmaa provided valuable editorial improvements. The authors thank all of them.
Appendix A. HTL226EC Course Abstract

**TITLE** SOFTWARE ESTIMATION AND TOOLS
**COURSE#** HTL226EC
**LOCAL#** HTL226EC
**STATUS** Active
**DURATION** 16 Hours

**DESCRIPTION** This course will cover the following topics:
- Overview of estimation in software development
- How to use two commercially available estimation tools:
  - COSTAR, based on Barry Boehm's CONstructive Cost Model (COCOMO)
  - SLIM, based on Lawrence Putnam's Software Life-cycle Model

The course will include theory as well as hands-on exercises. The first day will cover the software estimation overview, the COCOMO model and the use of COSTAR, and some of the theory behind SLIM. Concepts like productivity index (PI) and manpower build-up index (MBI) will be introduced. The second day will cover an in-depth look at SLIM's capabilities. Hands-on exercises will familiarize the participant with SLIM's input requirements and output such as schedule and resource allocation plans, reliability estimates, risk analysis, etc. They will also demonstrate the power of SLIM's what-if capabilities.

**OBJECTIVE** At the end of the course, the participant should be able to:
- Demonstrate high-level knowledge of some software estimation techniques
- Use COSTAR and SLIM to come up with reasonable estimates for project schedule, resource allocation, product reliability, etc. that can be used to validate their development team's bottom-up estimates.

**REMARKS** Use EDUCATE to enroll in this course.

**PREREQUISITES** There may be prerequisite reading.

**AUDIENCE** Project planners, project managers, team leaders, staff responsible for estimating project parameters in software development.

**FORMAT** Classroom (Traditional Stand-up class)

**SPONSORS** Toronto Lab SEPG & TTG

**CONTACTS** Nikki Panlilio-Yap
Danny Ho
References


About the Authors

Nikki Panlilio-Yap works in the Software Engineering Department, Group Technical Staff, at Loral Federal Systems (formerly IBM Federal Systems Company) in Bethesda, Maryland. She is presently on a leave of absence from the IBM SWS Toronto Laboratory where she has been part of the Software Engineering Process Group (SEPG) since its inception in August 1990. She joined IBM Canada Ltd. in July 1989 and worked in Utilities Product Evaluation for the AS/400* system. Before joining IBM, she had several years work experience in the government, academic, and commercial sectors of the computing industry. She obtained her Bachelor of Science in Chemical Engineering from the University of the Philippines, Master of Arts in Computer Science from Duke University in Durham, North Carolina, and Master of Science in Computer Science from the University of Maryland at College Park. She was a Fulbright Scholar and a World Fellowship recipient of the Delta Kappa Gamma Society International. She is a member of the Honor Society of Phi Kappa Phi and the IEEE Computer Society.

Nikki Panlilio-Yap can be reached at Lord Federal Systems, 6600 Rockledge Drive, Bethesda, Maryland 20817 U.S.A. Her e-mail address is nikki@fs.loral.com on Internet.

Danny Ho works in the IBM Microelectronics Toronto Laboratory as a team leader in infrared wireless development. He joined the IBM SWS Toronto Laboratory in 1989 and worked in the Tools and Technology Group for four years. His areas of special interest are software estimation, object-oriented software development, complexity analysis, tools delivery mechanisms, and tools platforms. Prior to joining IBM, Danny worked as a Software Engineer in the Communications Division of Motorola Canada Limited and was responsible for the analysis, design, and implementation of wireline and radio frequency communication systems and protocols. Danny received his Honours Bachelor of Science in Computer Science with Electrical Engineering and Master of Science in Computer Science from the University of Western Ontario. He is currently a member of the Association of Professional Engineers of Ontario.

Danny Ho can be reached at IBM Canada Ltd., 844 Don Mills Road, North York, Ontario M3C 1V7, Canada. His E-mail address is danho@torolab2.net.ibm.com on Internet.