IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4606

Mathematical Statistics in SW Engineering Education

Jaroslav Král, Michal Zemlicka
Dept. of Software Engineering
Faculty of Mathematics and Physics, Charles University, Prague
Malostranské nám. 25, 118 00 Praha 1, Czech Republic
Phone: +420 221 914 263, Fax:v+420 221 914 323
{kral,zemlicka}@ksi.mff.cuni.cz

ABSTRACT

The methods of mathematical statistics play an increasing role in SW engineering. Mathematical statistics is necessary to understand the functions of many applications as well as to mature SW development methods (proposed by CMM [CMM]). Mathematical statistics is the kernel of the analysis of software metrics. The analysis of software metrics must be used during the evaluation of the properties and impacts of software systems. The problem with mathematical statistics is that computer science students dislike it and are unable to use it. They like the simple Boolean (or black-and-white) world of computers too much. The importance of mathematical statistics and the attitude borrowed from experimental sciences is not even recognized by lecturers. The way to change this situation is discussed.

INTRODUCTION

SW systems grow, are more complex and support continually more complex functions using larger and larger data bodies. It is hardly possible without sophisticated methods of data analysis based on mathematical statistics. The complicated software systems cannot be developed or customized without modern management methods using data of software metrics (see the standard ISO 9126 and the recommendations CMM [CMM]).

It is well known (see e.g. [SG]) that the deficiencies in requirements specification and management errors during development customization are the leading triggers of the failures of SW systems, especially of information systems.

According to the experience of the authors gained via the development of SW systems and teaching at several universities the root reason of the failures is the improper (if any) use of the methods of mathematical statistics during requirement specification (offering proper management methods, the use of software metrics).

The solution of such problem is quite difficult as the computer professionals are not trained enough in mathematical statistics – and it is not recognized as important (compare [IS 2002, MSIS 2000]) although it is important (see [Král86]).

THE ROLE OF STATISTICS IN INFORMATION SYSTEMS

The growing data bodies collected by modern information systems (IS) and/or accessible via WWW cannot be analyzed properly without methods of mathematical statistics. The tools based on mathematical statistics must be also used if we want to measure the impacts and/or the effects (e.g. the financial ones) of modern software systems, especially of information systems. Statistical analysis should be included into the methodologies of the evaluation and testing of experimental laws and dependencies between software metrics like the dependency between the system development effort and the system size.

The tools of statistical analysis offered by many enterprise software systems (like workflow systems, production planning, and OLAP) are as a rule quite weak, if any. The methods of mathematical statistics are difficult to understand and/or to use for computer professionals as well as for many users. Many computer professionals strongly dislike statistical methods (and - more generally - the attitude of experimental sciences).

Modern SW development and software maintenance and/or use (SW processes) must include the collection and the analysis of software metrics. The maturity of a software firm can be - to a high degree – characterized by the quality of the collection and analysis of SW metrics (compare CMM maturity degrees).

Many software development techniques (e.g. the decision whether to stop testing and to release the system for distribution/use, the detection whether a system is “worn out” and should be cancelled/reengineered, [Rajlich, KZ02b]) depend on a proper use of the tools of mathematical statistics and the philosophy of experimental sciences (compare [Král98, KT2000]). The example of the Critical Chain method by Goldratt [Gold] indicate that there can be very surprising application of mathematical statistics in project planning.

Mathematical statistics should play an important, may be the crucial, role in software engineering, software science, and software use. It is not, however, generally recognized.

TEACHING OF MATHEMATICAL STATISTICS

Mathematical statistics is a science requiring a lot of mathematical knowledge and skills. If lectures on mathematical statistics are included into a curriculum a lot of lectures on mathematical analysis and probability theory must be included as well particularly in the case when mathematical statistics is presented as an abstract mathematical discipline. This implies that we must sacrifice some software-oriented lectures (e.g. lectures on latest programming languages). It is quite painful but it is not the main problem.

As the formal lectures on mathematical statistics must (should) be preceded by lectures on mathematical analysis and probability theory, they cannot be presented not earlier than third year of undergraduate curriculum (UC). It is quite unpleasant as it implies that we are unable to use mathematical statistics in software-oriented lectures in the first and second year and partially in the third year of UC. The result is that the students are not trained to use the methods of mathematical statistics during training basic software development skills. In fact it makes the lectures on mathematical statistics to a high degree useless due to student mental barriers and the lack of good inspiring examples. If the students learn computer-oriented lectures only they tend to consider all knowledge, problems, and skills not strictly related to computers useless and boring. We call such a narrow-minded attitude *hacker syndrome*. Hacker syndrome disqualifies computer professional for many upper level activities like the requirement specification with collaboration with users, teamwork, planning and audit, etc. Hacker syndrome disqualifies students from doing management. Even the lower management levels (like small team leadership) need a non-black-and-white way of thinking. The syndrome is very difficult to cure. Prevention is better. The prevention can be achieved via lectures on mathematical statistics and experimental sciences during the first study year. It would have some other advantages discussed below.

WHY COMPUTER PROFESSIONALS OFTEN DISLIKE MATHEMATICAL STATISTICS

Computer professionals often dislike mathematical statistics and are not able to use it. Some reasons for it are discussed in the previous paragraph. A further reason can be (with some simplification) characterized as a too strong emphasis on mathematical statistics as a mathematical discipline.

Software professionals tend like the black and white way of thinking. They dislike “gray” cases. In other words they are not happy if they must/may apply attitudes of experimental sciences like physics and to some degree economy and some humanities with experimental features like sociology or psychology.

We are convinced that the lack of knowledge and of paradigms of experimental sciences is a large drawback for computer professionals due to the following facts:

- It limits the application capability maturity principles from CMM.
- It is an obstacle for the application of modern management principles in the SW project management.
- The modern SW architectures like SW confederations [KZ02a, KZ02c] depend on the ability of developers to understand the functions of components and the functions often contain some function/knowledge of mathematical statistics.
- Optimality problems of SW confederations (i.e. peer-to-peer networks of applications/services) should be solved with the help of the tools of mathematical statistics.

It is interesting that in the Czech Republic many top information technology professionals are physicists.

In the long term the introduction of experimental/statistical paradigms could increase the adaptability of SW professionals in the changing world and to increase their chances to take part in well-paid and prestigious activities like requirement specification or management. The teaching of mathematical statistics can induce students to learn some experimental sciences (physics, sociology) to gain the paradigms common for all experimental sciences. It will increase the ability of the students to take part in the early stages of software development and to work and maintain the system during the system use. Last but not least it increases the ability of the students to work outside software firms. It can prevent the hacker syndrome and equip our students with useful knowledge and skills.

WHEN AND HOW TO TEACH MATHEMATICAL STATISTICS

The teaching of mathematical statistics should be based on a good theoretical background. But it implies that mathematical statistics cannot be taught early enough. If it is the case, the hacker syndrome can appear. It then results into resistance of students against mathematical statistics. This blind alley can be resolved in the following way:

We propose to teach mathematical statistics in two stages. The first step can be based on very elementary knowledge of probability theory and mathematical statistics (sometimes not too deeper than the knowledge obtained at grammar schools) an on the analysis of typical problems of the analysis of software metrics. Examples are: When to stop testing, when a system is to be reengineered, server capacity estimation, error prone components, software quality control, optimal project size, etc. Some tools from CASE systems like Together [Tog] can be used here. The teaching of the application of mathematical statistics should be accompanied by a lecture on some experimental science (sociology, financial controlling, experimental physics). Such lecture should include application of mathematical statistics on experimental data. The students can use a statistical software package.

The second stage can be based on more formal lectures on mathematical statistics discussed above. The formal lectures should again contain enough computer and software related examples. The mathematical statistics should be presented with the aim how to use it, not as an abstract mathematical discipline. This schema is about to be applied at some faculties of Masaryk University in Brno, Czech Republic.

CONCLUSIONS

The teaching of probability theory and mathematical statistics tend to be crucial part of the computer science education. Besides the points mentioned above (application in SW development, implementation of useful functions in applications, a greater adaptability of students, etc.) it forms a good background for the computer science research like probabilistic algorithms, simulations, relevance of software metrics, effectiveness of new software methods, quality of the software (e.g. reliability, effectiveness) etc. Moreover it can prevent the symptoms of the hacker syndrome or hacker like attitude. The importance of statistical methods is not pointed enough in the curricula [MSIS 2000, IS 2002].

The main aim is to change the attitude of the students to statistical methods as well as to experimental sciences. Note that while the trends in many professions increase the extension of the use of computers, for the computer professionals the trend could be “less computers, more common sense”. Note that computer professionals often do not collect the data needed for later statistical research or metrics. People even do not care about generation and usage of log files and other easily available sources.

REFERENCES

[CMM] Carnegie Mellon University, Software Engineering Institute (Paulk, M.C., Weber, C.V., Curtis, B., & Chrissis, M.B.), *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley Publishing Company, Reading, MA, 1995.

[MSIS 2000] Feinstein, D.L., Kasper, G.M., Luftman, J.N., Stohr, E.A., Valacich, J.S, & Wigand, R.T.: MSIS 2000 – Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems. ACM, 1999.

[Gold] Goldratt, E.M.: *Critical Chain*, North River Press, Great Barrington, MA.

[IS 2002] Gorgone, J.T., Davis, G.B., Valacich, J.S., Topi, H., Feinstein, D.L., & Longenecker, H.E, Jr.: *IS 2002 – Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*. Association for Information Systems, 2002.

[Král86] Král, J.: Software Physics and Software Paradigms. In: Kugler, H.-J. (Ed.) *Information Processing 86*. Elsevier Science Publishers, B.V. (North-Holland), 1986. pp. 129-134.

[Král98] Král, J.: *Informaní systémy*, (Information systems, in Czech) Science, Veletiny, 1998, 356 pp.

[KT2000] Král, J., & Töpfer, P.: Education of software experts for a changing world. In: Pudlowski, Z., J. (Ed.): *Proceedings of 2nd Global Congress on Engineering Education*. Wismar, Germany, 2000.

[KZ02a] Král, J., & Zemlicka, M.: Component Types in Software Confederations. In: Hamza, M.H. (Ed.) *Applied Informatics*. ACTA Press, Anaheim, 2002, ISBN: 0-88986-322-9, ISSN: 1027-2666. pp. 125-130.

[KZ02b] Král, J., & Zemlicka, M.: Software Confederations and the Maintenance of Global Software Systems. In: *Software Maintenance and Reengineering*. Budapest, 2002. pp. 61-66.

[KZ02c] Král, J., & Zemlicka, M.: Global Management and Software Confederations. In: Khosrowpour, M. (Ed.): *Issues & Trends of Information Technology Management in Contemporary Organizations*. Idea Group Publishing, 2002. ISBN: 1-930708-39-4.

[Pressman] Pressman, R.S.: *Software Engineering: A Practitioner's Approach*. 5th edition. McGraw-Hill, 2001.

[Rajlich] Rajlich, V.: A staged Model of Software Evolution. In: *Proceedings of 6th European Conference on Software Maintenance and Reengineering (CSMR 2002)*, IEEE Computer Society Press, 2002.

[Som2000] Sommerville, I.: *Software Engineering*. 5th edition. Addison-Wesley, 2000. ISBN 0-201-42765-6.

[SG] http://www.standishgroup.com/chaos.html.

[Tog] http://www.togethersoft.com A Together company homepage.