A Software Maintenance Survey *

Stephen W.L. Yip
Lecturer (CDS)
Dept. of Computing,
Hong Kong Polytechnic,
Hung Hom, Kowloon, Hong Kong.
e-mail: cssyip@comp.hkp.hk
tel: +852 766 7290, fax: +852 774 0842

Tom Lam
Provisional Airport Authority,
Hong Kong.

Stephen K.M. Chan
Dept. of Computing,
H.K.Polytechnic.

Abstract

We have decided to conduct a survey of the state of software maintenance in Hong Kong, as the software industry in Hong Kong and south China is expanding. The survey instrument is derived from the previous work of [7] and [5]. We sent out about 1000 questionnaires and received about 5% replies. Our results indicate that, in Hong Kong, *about 66% of the total software life cycle cost is spent on software maintenance.* The average application system is about *5 years old, consisting of 577 programs* and 308,000 lines of code. *Making enchancements* appears to be the most costly group of maintenance work (38% of all maintenance work undertaken), followed by *error correction* (16%). The most often cited problems in software maintenance are *staff turnover, poor documentation* and *changing user requirements.*

1 Introduction

As a consequence of the Asia Pacific economic growth, the software industry is becoming an important industry in Hong Kong and South China [13]. There are now large scale national software technology projects in mainland China, such as the Jade Bird and the Torch Plan projects [14]

1.1 Software maintenance

Software maintenance is defined as 'the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment' (ANSI-IEEE definition, [6]). The significance of software maintenance was first voiced in [7]. Following an extensive survey, Lientz [7] reported in 1981 that 70% of all cost in the life time of an average software product went to software maintenance. The sum of efforts in requirements analysis, design, coding and testing only accounted for 30% of the overall life time cost. Since then there have been a greater awareness of the significance of software maintenance, and a number of books ([10, 9]) have been published.

2 The conduct of our survey

We sent out about 1000 questionnaires and received about 5% replies. We have classified the business type of the respondents into eight categories. These categories and their distributions are shown in Table 1. Here we have included airlines and railway companies in the *Travel & Transport* categories. *Travel & Transport,* together with *Manufacturing* and *Government* make up the largest categories where their MIS departments have responded to our survey. These categories and distributions appear to be similar to the findings of other surveys [11] in Hong Kong.

Table 1: Distribution of survey respondents

Business Type	Count	Percent
Banking & Finance	5	10.5%
Commerce, Trading	5	10.5%
Education	3	6%
Government	8	17%
Manufacturing	8	17%
Property & Construction	3	6%
Servicing	2	4%
Stocks, Security, Insurance	5	10.5%
Travel & Transport	8	17%

*This research was supported in part by the Hong Kong Government funding body UPGC (University and Polytechnic Grants Committee).

0-8186-6960-8/94 $04.00 © 1994 IEEE

3 The design of our questionnaire

We have carefully studied the questionnaires used by Lientz [7] and Dekleva [5]. Then it was decided to adopt the questionnaire in [5] as our baseline survey instrument. This is because Dekleva's questionnaire is more concise and has absorbed previous work including that of Lientz [7] and Ball [1].

3.1 Major goals of our survey

There are a few figures of statistics that we regard as major goals to achieve in our survey. These figures are conceptually important as they help to project an overall picture of the state of software maintenance in Hong Kong. These goals are given below.

- An average ratio of cost (in terms of man years) between software development and maintenance in Hong Kong, assuming an average software product life time of 10 years.
- We are attempting to build a picture of how companies initially acquired their information system packages, whether the development was contracted-out or in-house . We are after information such as age and size of the software, the ranges and percentages of programming languages, development methodologies, and CASE tools used (if any).
- The distribution of maintenance activities such as analyzing change requests (or problem reports), consulting developers and documentation; designing, coding and testing the changes.
- The percentages breakdown of software maintenance work in Hong Kong, based on the well established categories of corrective, adaptive and preventive maintenance [7].
- Figures on a range of problems that are against the smooth conduct of software maintenance in Hong Kong. These may be human related problems such as high staff turnover due to lack of challenges and responsibility in maintenance work. There may be technical problems such as poor documentation , no proper development methodologies and unstructured programming languages used in the past.
- We like to look at the impact of technology, such as 4GLs (4th Generation Languages), CASE (Computer Aided Software Engineering) tools and OO (Object-Oriented) approaches to software maintenance.

4 Survey findings and analysis

The results obtained from the returned questionnaires are analyzed according to the goals listed earlier.

4.1 Information on application systems

The statistics about application systems, are presented in Table 2. These are given in terms of age, number of programs per system, number of data elements and executable lines of code. The initial cost of development and estimated cost of maintenance. We find that the average application is about 5.1 years old and has taken 11.5 man years to develop. The average system consists of 577 programs, 308,000 lines of code and 1438 data elements. In Hong Kong, the average cost of software maintenance is 66% of the total software life cycle cost.

Table 2: Statistics on application systems

System characteristic	Count	Mean	Min.	Max.	Standard deviation	Median
System age (years)	47	5.1	0.5	17	7.3	5
No. of programs	44	577	4	10000	912	400
Lines of code	33	308,561	1000	3,000,000	633,000	80,000
No. of data elements	36	1438	50	10000	1879	1000
Initial sys dev cost (in man years)	40	11.5	0.25	100	32	10
Maintenance cost	33	66%	17%	97%	47%	74%

In addition to the questions used in [7] and [5], we have included questions to find out more information about these application systems. Results indicate that about 67% of applications were developed 'in-house'. The popular programming languages used are COBOL (30%), RPG (21.5%), C (4%) and various 4GLs (22%). These information are presented in Table 3 below.

Table 3: System development

Development approach	Count	Percent
In-house dev.	30	67%
Bought-in as packages	10	22%
Dev. by contractors	5	11%
Use of 4GLs	10	22%
Use of CASE	8	17%

4.2 Maintenance work types

In our questionnaire, maintenance work is classified into six different types : enhancement, correction, performance tuning, answering questions, adaptation (new H/W, S/W) and re-engineering. The findings are presented in Figure 1. It can be seen that *making enhancements* (39.7%) is by far the largest type of software maintenance work performed in Hong Kong. The second largest type is *making correction* (15.7%). These figures are similar to those reported in America [5].

In addition to work types, we have also categorized maintenance activities. These are :

- Studying change requests : including consulting users and studying change request and user documents.
- Designing changes : including consulting developers and studying user needs.
- Coding changes : including studying code and documentations.
- Testing changes : including installation.
- After changes, including documentation updates, user training.

The percentage breakdown of the above activities, as reported in replies to our questionnaires, are presented in Figure 2. The largest groups of activities being *coding* (28.3%) and *design* (23.4%).

4.3 Organization and management

It is interesting to find out how software maintenance work is organized in relation to software development, in different MIS departments in Hong Kong. The survey result indicates that almost half of the respondents report that staff carry out development and maintenance work simultaneously. Less than 30% of replies indicate that there are separately managed software development and maintenance group in their companies. These findings are listed in Table 4 below.

Table 4: Organization of maintenance

Organization approach	Count	Percent
Separate managed group	13	27.5%
Same grp, separt. assignmt.	5	10.5%
Rotating assignments	6	13%
People simult. dev & maint	23	48%

The management of software maintenance is probed by a set of questions requiring only an answer of yes or no to each (as shown in Table 5 below).

Table 5: Maintenance management

Questions	Yes (%)	No (%)
Has the responsibility for overseeing maintenance process in your MIS dept. been assigned to one person ?	19	81
Are there systems in your dept that have to be maintained by individual because no one else understands the program logic?	38	62
Do you follow a formal procedure for determining priority of maintenance requests ?	68	32
Do you require in your dept. that older systems conform to programming standards ?	40	60
Do your MIS dept. keep the data on the history of change requests?	85	15
Does your dept. employ a formal method for determining when programs should be re-written ?	35	65

Figure 1. Maintenance work Types – Percentages

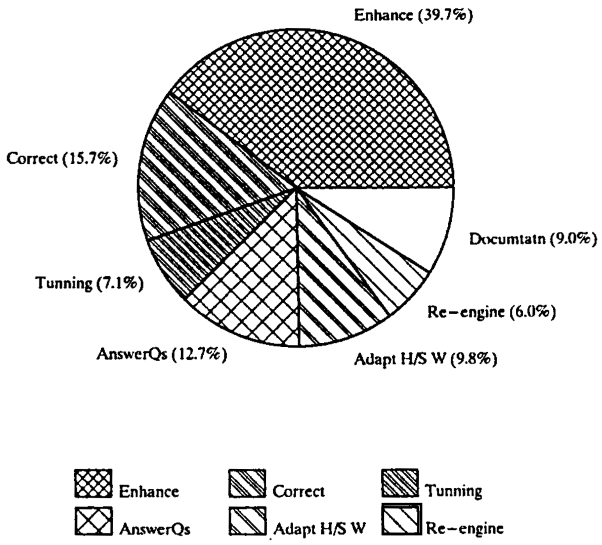
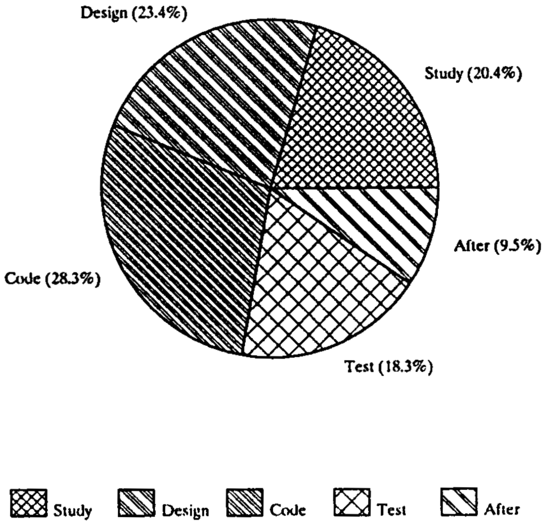


Figure 2. Software Maintenance Activities



The questions in Table 5 aim to find out if maintenance work is recognized as important in industry. Other questions are also asked on the procedures, methods, standards and records required for maintenance work. One of these questions reveals that about 38% of the MIS departments actually relie on some individual to maintain certain application programs where no one else in the same company can take over easily. This is undesirable, as staff volatility in Hong Kong is high. The results indicate that a proper infrastructure for software maintenance does not normally exist in Hong Kong.

4.4 Development and maintenance

Our survey also attempts to contrast the characteristics offered by software development and maintenance as opinned by respondents. This is shown in Table 6 below. For instance, only 2.1% of respondents think that software maintenance offers a higher personal satisfaction than development work. Whilst 83% say software development offers a higher satisfaction, with 14.9% of respondents report no difference in personal satisfaction between software development and maintenance.

Table 6: Characteristics of Dev. & Maint.

Characteristics	Maint.	Dev.	No dif.
Personal satisfaction	2.1%	83%	14.9%
Challenges	10.6%	72.3%	17%
Learning opportunity	10.6%	66%	23.4%
Career growth	6.4%	68.1%	25.5%
Exposu. to mangmt.	10.6%	59.6%	29.8%
Salary	2.1%	27.7%	70.2%
Staff turnover	53%	32.7%	14.3%

Table 7: Quality required of staff

Staff quality	Maint.	Dev.	No dif.
Experience	37%	39.1%	23.9%
People skill	19.6%	50%	30.4%
Problem solving skill	56.5%	10.9%	32.6%
Progrmng skill	21.7%	43.5%	34.8%
Design skill	4.3%	84.8%	10.9%
Knwldg of user busnes	17.4%	50%	32.6%
Intuition	33.3%	20.3%	44.4%

Staff quality required for development and maintenance work are given in Table 7. It is noted that maintenance work only reportedly requires a higher level of skill of staff in ***problem solving*** and ***intuition*** . Development work appears to require a higher level of skill in most areas listed in Table 7, with ***design skill*** being the most outstanding (84.8%).

4.5 Maintenance problems

One of the most important reasons for this survey is to discover the major software maintenance problems experienced in Hong Kong, so that they can be tackled. These major maintenance problems are classified and given in Figure 3. Here we have used the general categories for classifying maintenance problems as in Dekleva [5], for purposes of comparison.

It can be seen that the largest category of maintenance problems in Hong Kong is the ***'environment for performing maintenance'*** . Example of ***'environment'*** problems are :

- Lack of resources, manpower shortage is the largest problem in this category.
- Different programming styles used.
- Insufficient documentation.
- Problems with upgrades, data conversions.
- Lack of tools and graphical user interfaces for maintenance work.

The second largest category of maintenance problems is the *'management of maintenance'*. Reported problems in this category appear to be centred around the lack of management policy in both acquiring/requiring documentation, and the updating of them.

The third largest category of maintenance problems is *'personnel'*. This is mainly exhibited in the volatility and turn-over of staff. Apparently this is a common problem in Hong Kong since the late 1980s. The economy bloom in the 1980s has virtually made Hong Kong an 'unemployment free' zone. It is often heard that young computing professionals seldom stay in the same company for more than two years. Shortage of skilled staff is worsen with the large number of emigrations of Hong Kong citizen to other countries under the shadow of 1997.

Although the category of *'user-related'* problems only ranks 4th amongst other problem categories, *changing user requirements* is one of the most often quoted problems in software maintenance.

Figure 3.

Major Software Maintenance Problems in Hong Kong

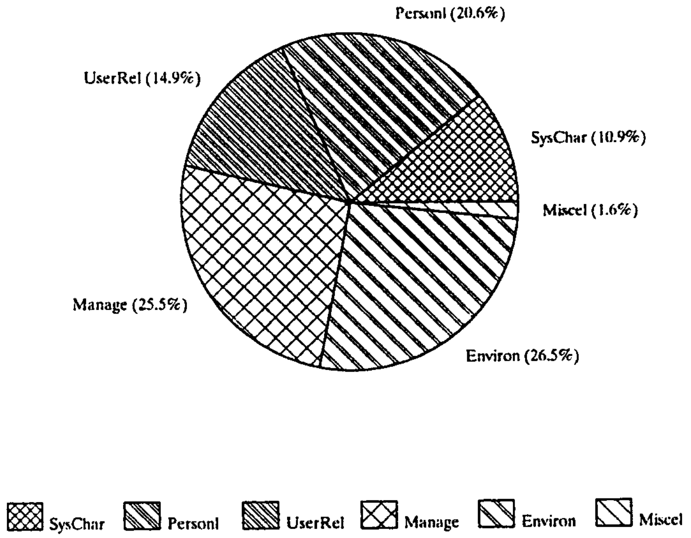
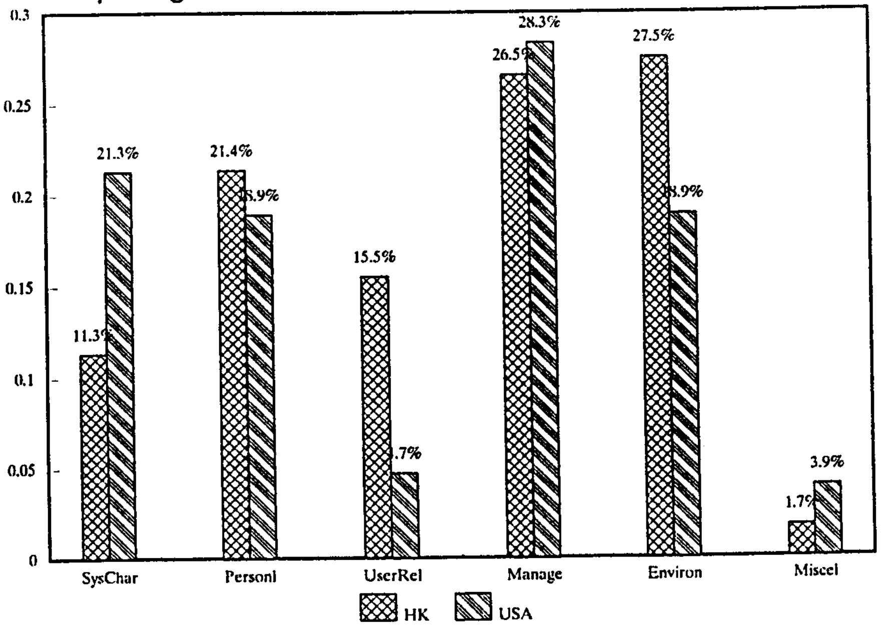


Figure 4.

Comparing Software Maintenance Problems in HK and USA



4.6 Use of 4GL, CASE and OO

In the returned questionnaires, there are only 8 (out of 49) that report the use of CASE tools, with 4 of them being SSADM tools. Ideally, the use of CASE tools should lower the cost of software maintenance. However, our findings indicate a higher than average cost of maintenance at all but one of these 8 sites that use CASE tools. It could be that these sites decide to use CASE tools because they have realized they are suffering from a high cost of software maintenance.

There appears to be a fair number of sites (22%) that are now dependent on 4GLs for their main application systems. Conceptually, 4GLs should help solve the problems of maintenance by allowing users to program at a higher level of abstraction with generated code that is more likely to confirm to programming standards [10]. Against theoretical predications, the average maintenance cost of these 4GL sites is reportedly higher than the overall average maintenance cost of 66%. We do not, at this stage, know the reasons for higher maintenance cost amongst 4GL users. However, there have been misgivings about the maintainability of application using 4GLs [3], with possible problems of monolithic 4GL code, modularization, defaults, explicit and implicit assumptions.

It is suggested that Object-Oriented systems can encourage software reuse; and will reduce the need of maintenance enhancement, by providing repositories of objects and methods to satisfy different users [12]. However, our survey has received no report of the use of OO (Object-Oriented) approaches from respondents. It is understandable that the usage of OO must be very low in Hong Kong, as the OO usage reported in the USA (round about 1994) is only about 5%[12].

5 Recommendations for better maintainable software

Our survey results indicate that there is still little awareness of the significance of software maintenance in Hong Kong. This is evident in the lack of proper environment and management for better maintenance, as discussed earlier. Considering that *personnel problems* (e.g. staff turn-over) are long-term and specific to Hong Kong , the problems of *environment* and *management* should be tackled first. A number of suggestions are listed below.

- There is a need to improve the requirements finding process, as *'changing user requirements'* is a frequently quoted problem. (It is also confirmed by our other finding that ***enhancement*** is the largest type (38%) of all maintenance work done in Hong Kong.)

- It is important to produce proper and up-to-date documentation during all life cycle stages including maintenance. A management policy on documentation is vital.

- In view of the high level of staff turnover, it is a good strategy to properly raise the significance and esteem of maintenance work.

- In order to reduce reliance on individual staff in software maintenance, team work and project rotation should be encouraged.

- The technology of OO approaches and CASE tools (e.g. automatic generation of test data and updated documentation according to source code changes) should be transferred from researchers to industry.

- Reverse engineering techniques of data-mining and program transformation [2] can be introduced to tackle problems of geriatric software.

  (Before implementing changes to a piece of software, the maintainer has to understand what the code does. In order to achieve productivity gains, software maintenance should be performed at a higher abstraction level than code level. Reverse engineering is the software maintenance process of acquiring high level abstract views from existing code [4]. Reverse engineering involves the identification or 'recovery' of the program requirements and / or design specifications that can aid in understanding and modifying the program. Program transformation is a technique that can be used for the automated transformation of low level code to high level formal specification.)

- Our survey reveals that there is a noticeable number (about 10%) of respondents are now engaged in the process of re-writing their main application systems from scratch. Advanced reverse engineering techniques, such as program transformation [2], should be useful. In any case, the concepts of designing software with maintainability in mind [8] should be introduced, in order to avoid future maintenance problems.

6 Conclusions and future work

The main conclusion of our survey is that the cost of software maintenance in Hong Kong is high (66%), and the significance of maintenance work is still not properly recognized.

The problem of software maintenance does not appear to be as severe as that in the USA as reported in [5] which surveyed the attendees of the 1990 IEEE Software Maintenance Conference. This can be explained in that America has a much longer history of software usage, and hence a larger backlog of geriatric programs wanting maintenance. The average age of applications in the USA as reported by Dekleva [5] is 9.4 years, and our finding in Hong Kong is only 5.1 years.

However, the nature of software maintenance problems in Hong Kong appears to be more human-related (e.g. staff-turnover, changing user requirements) than that of system characteristics (e.g. geriatric software) as found in the USA. This is shown in Figure 4.

The next paper, being drafted, will be submitted to the Journal of Software Maintenance. It is planned to publish the complete survey data in this next paper. It will also include other findings and analysis that is omitted in this paper for lack of space, including more comparison between software maintenance status in Hong Kong and the USA. We will try to investigate the observations that sites using 4GLs and CASE tools actually incur a higher cost of software maintenance. (This will be done by conducting site visits and interviews.) We also plan to increase the sample size of our survey by sending reminders to companies that have not yet replied to our questionnaires sent earlier.

References

[1] R.K. Ball. *1987 Annual Software Maintenance Survey: Survey Results.* Software Maintenance Association, Vallejo, California, 1987.

[2] K. Bennett, T. Bull and H. Yang. *A Transformation System for Maintenance - Turning Theory into Practice.* Proc. IEEE Conf. of Software Maintenance, Orlando, 1992.

[3] N. Chapin. *Software Maintenance: a different view.* ACM SIGSOFT Software Engineering Notes, 9(1), pp41-42, Jan. 1984.

[4] E.J. Chikofsky, J.H. Cross. *Reverse Engineering and Design Recovery: A Taxonomy.* IEEE Software, Vol. 7, No. 1, 1990.

[5] S.M. Dekleva. *Software Maintenance: 1990 Status.* Journal of Software Maintenance, Vol.4, pp233-247, 1992.

[6] IEEE. *Standard glossary of software engineering terminology.* ANSI/IEEE Standard 729, IEEE.

[7] B.P. Lientz and E.B. Swanson. *Problems in Application Software Maintenance.* Comms. of ACM, 24(11), pp763-769, Nov., 1981.

[8] C.J. Locascio. *Software Engineering for the Future with Ada.* Reverse Engineering Newsletter in the IEEE Software Engineering Newsletter, 1993.

[9] D.H. Longstreet. *Software Maintenance and Computers.* IEEE Computer Society Press, 1990.

[10] J. Martin. *Software Maintenance - The problem and its solution.* Prentice-Hall, 1983.

[11] G. Mead Research Ltd. *Information Technology in Hong Kong - What makes software packages sell.* 1988, Hong Kong.

[12] L.J. White. *The future of Software Maintenance and Testing.* Proceedings of Mini-Conference on Software Technology (editor: S.Yip), Hong Kong Polytechnic, 15-16 June, 1994.

[13] S.W.L. YIP. *Software Quality and Maintenance in Hong Kong.* A research grant proposal submitted to the UPGC in Sept. 1993, and has been accepted for funding.

[14] C.H. Zhong. *Survey of Software Industry Development in China.* Proceedings of Mini-Conference on Software Technology (editor: S.Yip), Hong Kong Polytechnic, 15-16 June, 1994.

Appendix - Survey Instrument

Software Maintenance Survey
Please complete and return to:
Dr. Stephen Yip, Computing Dept.,
Hong Kong Polytechnic,
Hung Hom, Kowloon, Hong Kong.
Tel. 7667290, FAX 7740842, E-mail CSSYIP@COMP.HKP.HK

We would first like to ask a few questions about you. These are only for the purpose of categorizing your responses. Questions will not be used to identify you or your organization in any way.

Name: ____________________
Company: ____________________
Business: ____________________
(e.g. banking, education, government, insurance, manufacturing, servicing, etc.)

Telephone number: ____________________
Title or position: ____________________
Number of employees in MIS team / dept.: ____________________
Hardware / software environment: ____________________

How many people report to you (directly or indirectly): ____________________
How many years have you been in data processing: ____________________
How many years have you been in software maintenance: ____________________

For this section, please concentrate on a particular information system.
Please identify an information system maintained by your MIS team / dept. which :
(i) has been operational six months or longer;
(ii) represents a significant investment of time and effort by your team / dept.; and
(iii) is considered by management to be fundamentally important to the organization.

Name of application system: ____________________
How many years has this system been operational: ____________________
How was this system developed: ____________________
(e.g. in-house development, bought-in as a package, developed by contractors)

Provide as many of the following measures of system size as possible.
Initial development cost in man-years: ____________________
(If system was developed externally, please estimate man-year by dividing cost of purchase (or contractor charges) by an average annual salary of your own analysts.)

Number of programs: ____________________
Number of executable lines of code: ____________________
Number of different data elements (fields, atributes): ____________________

Of the total number of executable source language statements, what percentage is written in each of the following languages?
Ada _____ Assembler _____ BASIC _____ COBOL _____
C _____ C++ _____ FORTRAN _____ Informix _____
Pascal _____ PL/1 _____ RPG _____ SQL _____
Others (specify) _____

What is the total number of users of this system such as interactive terminal users, receivers of printed reports, data entry clerks, etc. ____________________

How many full-time people maintain this system: ____________________
Average working hours / day for these people : ____________________

How many part-time people and contractors maintain this system: ____________________

Assuming a software life time of 10 years, total man-years in maintenance : ____________________

Projected life time cost of maintenance in percentage : ____________________
100% * Total man-years in maintenance / (Initial development cost in man-years + Total man-years in maintenance)

Are there any methodologies / CASE tools used in the devel. / maint. of this system: _____
If YES, please specify: ____________________

Concerning maintenance and development, how is your MIS team / dept. organized:

a) Separate groups; each has its own manager □
b) Same group, but separate assignments (some develop new systems, others maintain) □
c) A person rotates between new systems development and maintenance □
d) People simultaneously develop new systems and maintain existing ones □

Comparing software maintenance and software development, which in your opinion offers more:

	Maint.	Devel.	No diff.
Personal satisfaction	□	□	□
Challenge	□	□	□
Learning opportunity	□	□	□
Career growth	□	□	□
Responsibility	□	□	□
Exposure to management	□	□	□
Salary	□	□	□

Comparing software maintenance and software development, which in your opinion requires more:	Maint.	Devel.	No diff.
Experience	□	□	□
Ability to interact with people	□	□	□
Problem-solving skills	□	□	□
Programming skills	□	□	□
Design skills	□	□	□
Knowledge of user business	□	□	□
Intuition	□	□	□

Comparing software maintenance and software development, which experiences more turnover:	Maint.	Devel.	No diff.
	□	□	□

Has the responsibility for overseeing maintenance process in your MIS team / dept. been assigned to one person ? Yes □ No □

Are there systems in your team / dept. that have to be maintained by specific individuals because no one else understands the program logic ?
Yes □ No □

Do you follow a formal procedure for determining the priority of maintenance requests? Yes □ No □

Do you require in your team / dept. that older systems conform to current programming standards ? Yes □ No □

Does your MIS team / dept. keep the data on the history of change requests ?
Yes □ No □

Does your team / dept. employs a formal method for determining when programs should be rewritten ? Yes □ No □

Is your team / dept. currently in the process of rewriting any of your main application systems ? Yes □ No □
If Yes, please specify: Name of application ____________________ Age _____
and HOW ____
(e.g. redesign, recode in another language, reverse engineering, using CASE tools.)

What percentage (%) of the total person-hours worked annually by systems analysis and application programming personnel is allocated to software maintenance: _____

What percentage (%) of software maintenance time in your team / dept. is spent on the following types of maintenance work:
Enhancements __________ Error correction ______________
Performance tuning ______ Answering questions __________
Adapting to new hardware / software environment ______________________
Reengineering, renewal, retrofit ____________________________________
Documentation __________ Others (specify) ______________

What percentage (%) of software maintenance time in your team / dept. is spent on the following maintenance activities:
Studying requests __________ (includes: consulting users, studying documents)
Designing changes __________ (includes: consulting devel./maint. staff, studying user needs.)
Coding changes __________ (includes: studying code and documents)
Testing changes __________ (includes: unit testing to regression testing)
After changes __________ (includes: giving training, documents changes)
Others (specify) ____

What are the three major software maintenance problems* in your MIS team / dept.:
1. ____
2. ____
3. ____

If you could change three things* about software maintenance in your MIS team / dept., what would they be:
1. ____
2. ____
3. ____
(* e.g. management, resources, personnel, system characteristics, user requirements)

Do you incorporate steps to improve the maintainability of new future systems that are being designed now ? Yes □ No □

If Yes, please specify ____
(e.g. use of application generators, CASE tools; use of structured / object oriented approaches; techniques / tools to allow maintenance changes to be made on a higher level of abstraction, e.g. use design changes to generate automated code changes.)

The Hong Kong Polytechnic is planning a more thorough study of software maintenance. We will also be organizing seminars on software testing and maintenance. Would you or someone else from your MIS team / dept. be interested to participate? Yes □ No □