SOFTWARE PACKAGE EVALUATION:
APPLICATION TO A PERSONNEL SYSTEM
FOR THE CITY OF BOSTON

by
SUSAN ELLEN PRYTERCH
S.B., Massachusetts Institute of Technology
(1976)
SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF
SCIENCE
at the
MASSACHUSETTS INSTITUTE OF
TECHNOLOGY
June, 1977

Signature of Author

Alfred P. Sloan School of Management, May 19, 1977

Certified by

Thesis Supervisor

Accepted by

Chairman, Departmental Committee on Graduate Students
SOFTWARE PACKAGE EVALUATION:
APPLICATION TO A PERSONNEL SYSTEM
FOR THE CITY OF BOSTON

by

Susan Ellen Prytherch

Submitted to the Alfred P. Sloan School of Management on May 20, 1977 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

The purpose of this thesis is to present a method for evaluating software packages and then to apply it to a personnel package for the City of Boston.

The thesis presents first the reasons behind purchasing software rather than developing it in-house and recommends a make/buy analysis before considering a package.

The method of evaluation presented is based on a comparison to a normative model. The approach to evaluation recommended is to define the problem, do a requirement study and then to evaluate the packages, the suppliers and their services.

To aid in evaluation a checklist of factors critical to success is provided along with a method of weighing the factors. A means for rating packages is also provided so that packages can be compared and a selection made among them.

The personnel package for Boston is discussed in light of their needs and requirements, the alternatives available to them and the constraints on selection.

The evaluation method was designed to be used for application software which already has a number of users, is fully operational and error free.

Thesis Supervisor: Stuart E. Madnick

Title: Associate Professor of Management
ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Stuart E. Madnick for his helpful comments and suggestions.

I would also like to thank John Dimicelli of Boston's Office of Management and Budget for the time and interest he devoted to my thesis.

I am particularly grateful to my roommate Deborah A. McComber for doing the typing, correcting the grammar and telling me that my thesis was not boring.
TABLE OF CONTENTS

Abstract 2
Acknowledgements 3

I. Introduction 5

II. Pros and Cons of Packaged Software 8

III. A Normative Model of an Ideal Packaged System 13
    The Approach 13
    The Ideal Package 16
    Vendor Support 22
    The Vendor 24
    Summary 26

IV. The Evaluation and Selection Procedure 29
    Evaluation 29
    Selection 32

V. Boston's Personnel System: General Remarks and Background 36

VI. The Evaluation and Selection Process for Boston 40
    Evaluation Criteria 40
    Screening 43
    The Alternatives 44
    Estimating the Consequences 46
    - Functional Requirements 46
    - Technical Requirements 47
    - Package Design and Design Requirements 47
    - Personnel/ Payroll Interface 49
    - Installation and Implementation Support 50
    - Maintenance 53
    - Costs 54
    Evaluation of Alternatives 59
    Comparison of Alternatives 60
    Constraints 60

VII. Critique of the Methodology and Its Application 65

VIII. Conclusion 72

Footnotes 75

Bibliography 76

Appendix A 78

Appendix B 89
CHAPTER 1
INTRODUCTION

The fastest growing employer in the United States today is the government. The major portion of the budget is spent on personnel expenses. As a result, the operating costs of the government have increased at a tremendous rate since World War II. This rapid rise in costs can be attributed to an increase in demand by the citizens for services in the form of federal programs administered by state and local governments.¹

The financial crises of the nation's cities and the rapid growth in costs has brought the management practices of the cities to public attention. There is a definite need to efficiently manage our cities' resources.

The city of Boston, in an attempt to address the problem, decided that better control of its funds would be achieved through better management of its personnel. Boston has an annual operating budget of approximately 250 million dollars. Seventy-five percent of the budget represents salaries paid to the city's 14,000 employees.

The personnel function represents more than just a record keeping activity. A personnel/payroll system provides the data needed for resource development, manpower planning, and resource allocation. In addition, a personnel system simplifies compliance with such government regulations such as EEO.

Once the city identified a need for a computerized personnel system, a decision on the source of the software had to be made. Three alternatives were considered. They were in-house development, contracting out and
purchasing a software package. Buying a software package for a personnel system is a viable alternative. Personnel management is common to every organization and is a function easily understood. Therefore, there are several packages on the market with varying amounts of detail.

It is easy to develop a checklist of technical and operational factors which indicate a good product. In fact, much literature has been written on the subject of software evaluation in the form of that type of checklist. There are several problems with just having a checklist. First, no package will meet needs exactly. Trade-offs and concessions will have to be evaluated and made. Weights must be assigned to each factor which reflect their importance to the organization. The criteria by which packages are judged cannot be generalized to fit every situation, since needs, requirements and the environment of every organization differs.

For most applications there is more than one available package and the problem becomes how to rate the packages so they can be compared. There is no Consumer Report for software packages. All the research must be done by the buyer. Generally, the best and probably only source of information is previous buyers of the package.

Purchase of a software package can be viewed as a decision making process. Each package is an alternative solution to the problem. The focus of this thesis is what to look for when buying a software package, identifying the critical factors needed to make the decision, and how to determine the relative importance of each in order to compare packages and reach a decision.

Chapter 2 looks at why behind purchasing a software package. Some
reactions of people who did and reactions of vendors to their clients are presented.

Chapters three and four analyze the decision process of package selection. Chapter 3 takes up the issue of what is a successful system and what determines an acceptable solution. Chapter 4 contains a methodology for evaluation and selection.

With these two chapters as background the situation at Boston is explored and a set of evaluation criteria is developed. Based on the criteria, their alternatives are examined and a plan of action is proposed. First, a decision is made in the absence of system constraints. It is based just on the normative model reflected in the evaluation criteria. This decision is then modified to take into account the factors which constrain selection.

The final chapter examines the methodology, for its use and operational effectiveness. The outcome of the decision made in Chapter 5 is analyzed for its reasonableness and feasibility.
CHAPTER 2
PROS AND CONS OF PACKAGED SOFTWARE

Software can be divided into two categories: systems and application. Systems software are those programs designed to make the hardware run efficiently. This category includes operating systems, database management systems, assemblers, compilers, input/output schedulers and control routines, simulators, etc.

Application programs perform data processing or computational tasks. Typical applications are accounts payable/receivable, inventory control, personnel management, matrix inversion, etc.

There are three ways to obtain software: develop it in-house, contract it out or buy an "off the shelf" package. Until recently, there were only two major sources of software. One source, the computer manufacturer, supplied for the most part only systems software. For application software the organization had to rely on its in-house staff.

In 1969, IBM unbundled its software, paving the way for an emerging software industry. Previously, software was used as a marketing device and was thrown in free with a hardware purchase. The software, in reality, was not free. The cost was incorporated in the price of the hardware. Since it was used primarily for marketing, the software was not technically as good as it could be and was designed to create a need for more of the manufacturer's hardware products. Unbundling meant that now the software had a price attached to it. This gave the buyer an incentive to examine other alternative sources for software, thus creating a market for companies specializing in software development. Sales of software
packages have been growing at a rapid rate. Packages benefit both the buyer and consumer by providing a higher return to the developer and a lower cost to the customer than custom work.

In addition to software houses which provide custom and/or packaged software, packages can be obtained from user groups and software brokers. In a user group, members pool together programs they have developed and offer their use to other members free of charge. Quality varies and installation support and good documentation are often sacrificed.

Software brokers are intermediaries between the software house and the buyer. The extent of service varies. Besides searching for and screening packages for the buyer, some brokers are fully staffed to provide support and maintenance for the packages they sell.

Choosing among in-house development and the other sources involves comparing the advantages and disadvantages of each alternative. The ultimate decision of what option to use is based on a make/buy analysis.

Opting to buy a package rather than developing it in-house is considered because of one or a combination of the following:

- the economics of "do it yourself"
- urgency of need for an operational program
- insufficient staffing
- insufficient expertise.

The right software package can save time and money, and in some ways can be more practical than in-house development. The determination of right package is not trivial however. It involves correctly defining the problem and identifying the needs and requirements. People who understand
the problem, the business, software and hardware are needed in order to make the proper selection.

The high cost of in-house development, the lack of qualified systems people and the length of the time span before a company has an operational system have led companies to increasingly consider packaged software for routine needs.

In a study done by Currie, Cooper and Lybrand, Ltd. it was found that 29-36% of computer staff, 15-21% of computer time, and 28-33% of total computer costs go into systems development and maintenance. Not all companies can support a large in-house staff, and there is a limited number of competent software analysts. These two facts in conjunction with a large commitment to systems development and maintenance indicate the best allocation of resources would be to use in-house staff either to develop software to meet unique needs or for those projects with a high return. For routine needs a good vendor can provide the specialization and experience needed to produce a high quality package. For the software house, software development is their business not just a staff function. Since the software industry is highly competitive the vendor must hire the best in order to stay in business. There is also no point in every company inventing its own wheel.

Buying a package can avoid the pitfalls of in-house development. The cost factors of in-house development are easy to define, but estimating them is difficult. They are based on forecasting the time needed to design, program, test and debug. These forecasts in general underestimate the situation. As a result there are over-runs in schedules and
unanticipated costs. A package is a proven product whose details and costs are known in advance. If the package has been used before, it will already be tested and debugged. The package is available now and, therefore, cuts down the risk that by the time the installation is over and the is fully operational, that it will not meet changed organizational requirements.

Economically, a package is cheaper than in-house development, for the simple reason of economy of scale. A software vendor can spread development costs over a greater number of users. Buying a package eliminates the costs associated with time and personnel commitment needed for development. A package can also save on installation costs. If a package has been installed previously, the developer may be able to provide a plan for installation and implementation based on his experience. Such a plan could make the installation process go faster and more smoother.

Developing software in-house involves a large lead time before it is fully operational. All design, programming, debugging, and testing must be done by the staff. The lead time associated with the purchase of a package includes searching and evaluating the package and making the necessary modifications. This lead time can be considerably less.

Of course, there are disadvantages associated with the purchase. Due to the intended universal use of the package it will not meet all the needs of the organization, and some sort of modification will have to be done. These modifications may mean that the package will not run efficiently. Since the in-house staff did not develop the package they will not be as familiar with it. This can cause problems when and if future
modifications and updating is needed.

The benefits derived from a software package presuppose that a thorough search and evaluation of packages and vendors has been done before the purchase is made.

In one study vendors were asked to list the top three issues their customers complained about. The four mentioned the most were documentation, maintenance and updating, package not satisfying needs and installation. The customers were not concerned about performance or efficiency. When the users were asked they mentioned documentation, efficiency, better performance and maintenance the most.4

Dissatisfaction with the product can be classified into one of the following areas:

1) It does not satisfy needs or meet the requirements
2) product quality
3) vendor service.

The sources of dissatisfaction for the most part can be found to be the fault of the buyer. Most vendors when questioned felt that in general the customers were neither well informed nor had sufficient expertise to buy software and that they were concerned more with price than with value.5 Failure to identify needs before buying the package will cause dissatisfaction with the package's performance. Dissatisfaction with the product's quality or vendor's service can be partially eliminated by carefully and properly evaluating both before any purchase is made.
CHAPTER 3

A NORMATIVE MODEL OF AN IDEAL PACKAGED SYSTEM

The Approach

The need for a computer system application begins with the recognition of a problem and the realization that it cannot be solved through revision of existing manual procedures. The extent of the need is determined by comparing what is currently being done, with what would be done if the system existed. This comparison should reveal how inferior, if at all, the current system is. On the basis of this comparison, a decision can be made on how much it is worth to the organization to improve it.

A system begins with identifying the problem and the preferred outcome, so that possible solutions can be proposed. The preferred outcome is a normative model stated in the form of requirements which when satisfied resolve the problem. Before solutions can be identified, there has to be a study of the existing system, an analysis of costs and benefits, review of the alternatives and the determination of economic, technical and organizational feasibility. The proposed solution to be considered in this chapter is in the form of packaged application software.

The obvious goal in selecting which package to purchase is to obtain a successful system. Success as it applies to the organization must be defined and a method of measuring it developed. This involves understanding the problem, the organization and the functional areas involved.

An approach to the selection of software can be broken down into eight stages:

- define the problem
- state goals and objectives
- state goals and objectives
- determine needs and objectives
- perform a make/buy analysis
- develop an evaluation criteria
- screen available packages
- evaluate vendors and packages
- make decision.

The first three stages are the same whether software is done in-house or purchased. Step 4, the make/buy decision, is the point where the organization decides whether it is more feasible to develop in-house or to buy the software. The rest of the steps follow as a result of electing to buy a package. These steps form the evaluation and selection procedure which will be discussed more fully in Chapter 4.

The purpose of defining the problem is to clarify exactly what is to be done. It serves to put bounds on the scope of the application.

To identify the problem, descriptive and normative models should be developed and compared. A descriptive model represents how a function is currently being done. It should outline what information is collected, who uses it and for what purpose. The information flow through the organization should be tracked and flowcharted. A normative model depicts the ideal situation. It reflects the way the organization thinks things should be done.

Comparison of the two models will reveal discrepancies and indicate possible areas for change which will lead to improvements.

From the problem definition, goals are established. A goal is a
broad statement of what is needed to solve the problem. Objectives are specified based on the goals. They are a more specific statement of ends. Objectives are based on the reasons for the system. These reasons are obtained by investigating who wants the system and why. (Investigating who does not want the system and why can also be informative.) There are two classes of reasons for a system, efficiency and effectiveness. Efficiency is increased output for given input. It may be that a certain task is not being done as well as possible in relation to some predefined performance criteria, and that it could be done faster, more accurately or cheaper.

Effectiveness is the extent to which a process produces the intended or expected results. A system may be needed so that a function can be done better.

Determining the needs and requirements involves clearly defining the elements of the system based on the preferred outcome. A system consists of hardware, software, a database, procedures and operating personnel. Hardware is the computer, secondary storage and input/output devices. In this case, software is the application software and any needed data management software. System functions should be specified. They include file creation, file maintenance, report generation, on-line inquiry and information retrieval. Database, as it is used here, is broadly defined to be data stored on some storage media. Procedures make up the function which is being automated. Operating personnel are those people required to prepare input, run the system and process the output.

The user should take a black box approach to design, specifying
what has to be performed and not how. Preliminary design specs are for input into the make/buy decision. If purchasing the software, the buyer should be able to rely on the expertise of the vendor's design and development. If the decision is made to buy then detailed specs can be drawn up, later.

The black box approach entails defining what is to be the output, then defining what are the inputs and calculations needed to obtain it. The output should be specified in terms of volume, frequency of use, data fields, reports and calculations. Specification of volume in terms of number of records should be for present needs and to take into account growth.

The make/buy decision involves looking at available resources and alternative uses for them. By comparing the total cost of choosing each alternative, the minimum cost approach is selected.

Given that there will be more than one possible solution, a method of evaluation is needed to compare them. Using some criteria as a guideline, a grade is assigned to each alternative based on how well it satisfies the requirements of the preferred outcome. The criteria for evaluation differs depending on the type of problem, but it should contain those factors essential for solution to the problem.

The Ideal Package

The factors to be discussed are the elements for a selection criteria designed to reflect the ideal situation. In the case of a software package, ideal implies meeting certain operational, functional, and technical requirements. The package should be well designed and documented; be able
to handle all paper and information flow; take data and aggregate it into meaningful information; process day to day transactions; support related functions; and provide for effective and efficient operations. In addition, it should be easy to install, fully operational and error free. All needed vendor support should be available.

The quality of the software depends on the quality of the effort and talent that went into its design and development. Quality of the talent can be assessed by examining the credentials of the people involved in its design and development. Readily available, quality and complete documentation is a good indicator of the overall quality of the package and professionalism of the developer.

Documentation is needed to support installation and operation. It should be easy to understand and of sufficient detail so that the in-house staff can make its own modifications and do any necessary debugging. Documentation is essential. The designer is not part of the in-house staff and will not be around if questions and problems arise.

Documentation should be provided at four levels: system manuals, program manuals, operations manuals and user manuals. These manuals should be designed to meet the needs of the systems analysts, programmers, computer operators and end users. Table 1 is a checklist of things to be included in documentation.

The factors which identify a well designed package are universal to all software design whether the software was developed for in-house use or to be marketed. Certain factors like flexibility and modifiability gain additional importance if the package is being marketed. In order to
appeal to a wide range of users, the package must be adaptable to different situations. Functional and technical requirements cannot be generalized. They depend on the type of application and the environment in which it is being installed. Meeting these requirements entails taking needed input, performing necessary care, and processing and producing required output.

A well designed package is based on certain design goals. They are effectiveness, efficiency of operation, simplicity of form, ease of understanding, modifiability, flexibility, expandability and reliability.

An effective package is the result of having a good understanding of the function being modeled before developing the specs. The objective is to model the function not just automate the procedures.

A package should be designed to make efficient use of the resources it requires for operation. A package's original design may be very efficient but subsequent modifications can create inefficiencies in operation. Therefore, before a package is purchased the original use of the package, the equipment configuration and operating system should be investigated.

Simplicity of form and ease of understanding are needed so the software is easy to debug, modify and use.

The goals of flexibility, modifiability and expandibility are important for achieving generality and preventing obsolescence. Flexibility means that the user is not constrained to any special input/output format or method. A program which is expandable is one that allows for growth in number of fields, records and features so that present and future needs can be met. The structure of the software design should
TABLE 1 DOCUMENTATION CHECKLIST

Overall System Documentation

- General system description.
- System flow chart.
- Equipment configuration.
- Operating system and programming language specifications.
- File definitions and layouts.
- Input and output definitions and layouts.
- System scheduling and control procedures.
- Actual or estimated system timings.
- File protections and audit trail provisions.

Program Documentation

- Detailed description and detailed logic flow charts for each program.
- Significant program details such as tables, special constants, program switches, accumulators or counters.
- Description of special programming features, methods, modifications.
- List of all other programs, sub-programs, macros, library programs called by each program.
- Assembly or compilation listing, cross-reference listing, load map, memory dump.
- Test data and sample run showing inputs, outputs, console log.
- Actual or estimated run times.

Operations Documentation

- Input preparation procedures.
- General description and flow chart for each run.
TABLE 1 (continued)

- Operator set-up instructions.
- Description of all abnormal operating conditions.
- List of all normal and abnormal messages and halts, meanings of each and corresponding action to be taken.

User Documentation

- User oriented system description.
- Description of required forms and equipment.
- Source data preparation procedures.
- Error and exception handling instructions.
allow for minor changes without requiring major reworkings.

A downed system can stall the operations of an organization so obviously the package should be reliable. Reliable means minimum down time and good recovery. A reason behind buying a package and not developing it in-house is to get a tested product that is error-free and fully operational.

Achieving these design goals is facilitated through the use of modular design. Design is done by breaking procedures up into groups called building blocks or modules. Each module is independent and does not rely on another module to operate. These modules are linked in accordance with the logic of the mutual input/output relationship.

Modular design is a very viable technique to use for mass produced software. It permits a vendor to offer a wide range of functions without over-generalizing the package. The customer has only to buy those modules actually needed and has the option to buy others later.

In evaluating a package, design features should also be taken into account. Design features are number of fields in master file, type of file structure, input/output methods, etc. No package will have exactly the desired features. They may have more or less features or slightly different ones. Slightly different does not mean the package will not do the job adequately.

The design features should be evaluated in light of the organizational needs and requirements to determine if they are sufficient. If not, to determine how much modification is needed to make them so.
Vendor Support

Acquiring the software is a small part of the system. The system must be installed, implemented and maintained. Generally, vendors have a program of support which includes installation aid, training, conversion and maintenance. Support is assistance supplied by the vendor in the form of skilled manpower. The extent of the support depends on the complexity and price of the system, sophistication of the purchaser and the importance placed by vendor on various services.

The burden of successful implementation does not rest entirely on the vendor. It depends also on the amount of resources like the time and staff the client is willing to commit. Initial effort put into planning, conversion and training can prevent doing unnecessary maintenance, modification and revisions later. If the vendor has previously installed the package the purchaser can benefit from his experience in the form of a recommended installation and implementation plan.

Three types of conversion are involved in installation. The first is file conversion or file creation. Data must be collected and put in a form and format that the software can utilize. Initial gathering of data can put increased work loads on clerical and operating personnel. Resources should be allocated to take this into account.

The second type of conversion involves making the software programs operational on the client's equipment under his operating system. This is where the vendor's familiarity with the client's equipment and operating system is utilized.

In the functional areas the software application effects, changes
in procedures will occur. This is especially true if the changeover is from a manual to an automated system. Personnel have to be trained in the new methods of doing things.

Typical methods of training are manuals, classroom sessions, or "on the job." The method used depends foremost on what the vendor provides. The training offered should be examined to determine its exact nature, level and location. The type of training required depends on the complexity of the system and the qualifications of personnel involved in it. The four levels of personnel for which training may be needed are clerical, operators, programmers and systems analysts and managers.

In addition to initial training, some vendors provide refresher courses to keep their clients up on the state of the art. These sessions are usually seminars at extra cost.

Maintenance includes debugging and updating. Updating is supplying improvements and modifications of the software to clients at some additional fee. A written warranty of error correction is usually provided for some reasonable period of time.

The amount of on-going maintenance is based on some understanding between the buyer and the seller and is usually written into a contract. Maintenance contracts are usually on a renewable yearly basis. Service is provided at a set fee. There are three types of on-going maintenance arrangements. The first provides for both debugging and updating. The second type covers only error correction. In some cases no maintenance is provided. The user is supplied with a copy of the package and then charged for any assistance required.
It is important to take into account the capability of the vendor to perform maintenance and generate improvements on a long term basis. Another consideration is if the supplier will be able to upgrade the package if a new release of the operating system is put out or the equipment configuration is changed.

The purpose of evaluating the vendor is to determine the capability of the vendor to deliver both a quality package and quality service. Dissatisfaction with a vendor can be the result of a lack of understanding of what the buyer thought he was getting and what he was being offered. The buyer should present to the vendor a clearly stated set of objectives, evaluation criteria, a time schedule, detailed equipment configuration, budget constraints and any technical constraints.

The Vendor

Clues to quality service and a quality product are the credentials of the staff. Key personnel should have strong and concentrated experience in the specific problem areas addressed by the product. The staff should understand and have experience with the buyer's industry and environment. An indication of this, is the ability of the vendor to provide financial analysis of the impacts of the product on the buyer's environment. In addition to expertise, the vendor should be evaluated to determine if it has available staff, money and is geographically close enough to provide support.

The quality of the package rests not only on the reputation and experience of the vendor but on the person who actually did the design and development. As important as the credentials of the designer is his
availability. Since the in-house staff did not do the development, they may not be familiar enough with the software's workings to solve a problem should it arise.

The expertise and availability of the in-house staff will determine the amount and extent of vendor support required. If the reason the software was purchased was the staff did not have the time to commit to the project or the expertise to do it properly, it is possible that they will not be able to upkeep the software once it is installed. In this case, ongoing support will be necessary from the vendor. If the buyer must rely heavily on vendor support, it is essential to separate out the back yard garage operation from the serious supplier. A vendor should be able to supply concrete guarantees for maintenance, personnel training and installation assistance.

The software industry is a relatively new industry and suffers from infant industry problems. Cost of entry is very low and demand for the product is very high making it a highly competitive industry. There is a rapid turnover rate of both companies and products. EDP Performance Review annually lists all packages actively being marketed. Comparing 1976 to 1975, 57 packages were dropped and 61 new ones were added indicating that the industry is far from stable.

To insure a higher probability of obtaining on-going maintenance, support and updating, the companies financial status and commitment to the industry should be examined. The number of years in the business, the number of products and amount of marketing effort are indicators of commitment. A serious competitor should have a good handle on his costs
and, therefore, be able to give a good and firm pricing of his product.

Summary

In summary, a package should be well designed in that it meets certain design goals. It should be useful to the organization. Useful implies performing the necessary functions in an efficient and effective way such that the organizational goal are achieved.

A good vendor should have the following characteristics:

1) Good financial status – this indicates stability and increases the possibility that the vendor will be around to provide on-going maintenance and continual enhancement and updating.

2) A quality staff with a low turn-over – it is nice if the people who installed the package and trained the staff and, hence, are familiar with a company's operation are around if problems come up in the future.

3) Availability of staff – a downed system can paralyze a company's operations. The vendor should be able to provide quick on-site aid if difficulties should arise.

4) Commitment.

5) Reputation and experience – experience is a function of the number of years in the business, the number of products and the number of customers serviced. Talking to other customers is the best way to obtain an overall picture of the quality of a vendor's package and service.

6) Ability to supply all necessary assistance needed for successful installation, implementation and maintenance of the system.

The factors discussed in this chapter are presented as a checklist in Table 2.
TABLE 2 EVALUATION CHECKLIST

I. Package
   A. Design Goals
      - Effective
      - Efficient
      - Simple
      - Ease of understanding
      - Modifiability
      - Flexibility
      - Expandability
      - Reliability
   B. Design Features
   C. Credentials of Designer

II. Proposed System – meets:
   Organizational goals
   Functional needs and requirements
   Technical needs and requirements
   Interface with existing systems
   Operational considerations
TABLE 2 (continued)

III. Vendor

A. General
- Stability
- Years in business
- Financial status
- Geographical location
- Familiarity with function
- Familiarity with clients business
- Credentials of staff
- Availability of staff
- Experience with equipment configuration
- References

B. Services
- Documentation
- Maintenance agreement
- Training program
- Installation aid
- Project plan
  -- Time table
  -- Implementation
  -- Conversion
CHAPTER 4
THE EVALUATION AND SELECTION PROCEDURE

Evaluation

Decision is the selection among alternatives with the aim of achieving some goal. Each alternative has associated with it a set of consequences. Selection is made based on some criteria which ranks the alternatives and their consequences. The alternative selected is the one which leads to the most preferred outcome.

The likelihood of a particular alternative being chosen depends on:

1) relevant values and objectives of the organization
2) perceived alternative courses of action
3) estimated consequences of choosing that alternative
4) evaluating the consequences. 8

The selection of a software package is a decision making process. Alternative packages have to be evaluated and compared in order to make the best possible selection given the alternatives.

Relevant values and objectives of the organization with respect to software packages should become apparent from the analysis done in order to define the problem and to identify the needs and requirements. Relevant factors for the organization should be selected from the factor list and weighted according to their impact on system success. Some factors may be so important that they result in a "go-no-go" decision. These factors should be specially indicated.

Assigning weights is not an easy task and is far from precise. It involves understanding the organizations utility with respect to the
various aspects of software packages. As a result, it is very subjective.

Selecting factors and assigning them weights should be done before looking at the vendor's proposals, so that the decision is not swayed by the vendor's presentation. A vendor will highlight his strong points and make them appear to be the most important factors.

Alternative courses of action are the various packages on the market. Alternatives are found by searching the market for available packages and screening them for their suitability. The search is facilitated by publications such as the IPC Quarterly which lists and describes the packages currently being marketed. Trade journals also have lists, advertisements and directories for software packages and their suppliers.

The initial screening could possibly lead to a dead end in that there are no suitable packages available. If that happens, an alternative source such as in-house development or contracting out will have to be considered. However, if it is a common and routine application, there will probably be several alternative packages among which to choose.

Consequences for software packages are the options, features and services associated with the purchase of the package. Estimates of these consequences are inputs for the decision making process which allows one package to be compared to another.

The inputs are obtained by asking questions of the vendor or from his submitted proposal. Since the vendor's proposal is used to sell their product, it will be biased and, therefore, should be closely examined in order to determine what is theory and what will be the reality of practice. Unfortunately, there is no consumer report for application software.
packages. This makes it hard to establish the reliability of the vendor, his package and his service. If there is an in-house staff, they can examine the proposal for its reasonableness and feasibility. Clients of vendors are good references for the vendor and his product. Talking to users and visiting their installation to see the system in operation can give insight into how well the package functions; operational needs and requirements; and types and extent of vendor services. Appendix A contains a list of questions to ask vendors and their clients to aid in evaluating the different packages, services and vendors.

Estimating a consequence implies assigning a rating to each factor of each package. The rating should reflect the worth of that factor to the organization. A value can be represented as either a dollar figure or as a rating. There are problems with either method. Ratings are subjective and allow a lot of room for difference of opinion in their meaning.

Cost figures such as dollars are a unit of measure most people understand. However, obtaining an accurate dollar measure of worth is difficult. Two types of dollar amount can be attached to a factor. It can represent the value to the organization or the price of obtaining that factor. Price is easier to obtain but does not necessarily reflect value which is more important.

The price of the package reflects the costs and pricing policy of the vendor. A generalized industry breakdown is:

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>24%</td>
</tr>
<tr>
<td>Marketing</td>
<td>28%</td>
</tr>
<tr>
<td>Administrative overhead</td>
<td>10%</td>
</tr>
</tbody>
</table>
maintenance and installation 10%
replication 7%
profit 21%

In many cases the final price of the package is determined by spreading costs over the number of expected customers.9

The rating method presented here is a value rating. Each factor is assigned a score based on how well the factor as presented in the proposal matches up to the normative model. The ratings were based on a scale of 0 to 10 and is as follows:

10 - exactly
8 - adequately
6 - with slight modification
4 - with some modification
2 - with a lot of modification
0 - not at all or factor not presented in package

A zero rating on a "go-no go" factor means immediate rejection.

Evaluating the consequences entails summing up all the values so that there is a representative number attached to that proposal reflecting its worth to the organization. A score is obtained by multiplying the rating by the weight. This scales the rating so it reflects the importance of the factor to the firm. The scaled ratings are then summed up to get the total score. Once all the proposals have been evaluated, they can then be compared and a selection can be made.

Selection

The decision process of selection involves defining the acceptable
solution, evaluating the alternatives based on a cost/benefit analysis and then making the decision.

A package is considered acceptable if it has all the characteristics required to insure system success. To determine if a package is acceptable, its consequences are compared to the evaluation criteria. This is the purpose behind developing the checklists. Packages which receive less than acceptable ratings on certain factors should be examined to determine if they can be improved through modification done either by the vendor or in-house.

To take into account improvement through modification, factors are assigned a new rating and the cost of modification is added to the direct cost of the package.

The next step in the decision process is to do the cost/benefit analysis. The cost/benefit ratio is used to measure the value of each package so that alternatives can be compared.

Calculation of the cost/benefit ratio is done by taking the total cost associated with the purchase of the package and simply dividing it by the score of the package.

The direct cost of the package is the lease or purchase price. This price does not represent the total cost of the acquisition of the package, however. Installation costs in general will be the majority of the total cost. Total cost should include the indirect as well as the direct costs.

Possible indirect costs are evaluation, modification, training, installation, conversion, execution, maintenance agreements and debugging.
The top solutions should be compared in order to identify the major differences among them. An evaluation is then done based on only those factors associated with the differences. This sub-evaluation is used to assess the trade off of selecting one package over another. Based on the sub-evaluation the selection is made.

Package choice is not guided just by technical and economic considerations. The package must fit in with the organization and its environment. Package selection is constrained by the constructs of the organization. Constructs are organizational structure, available resources and the organization's method of doing things.

Typical constraints relevant to package selection are time, money, available personnel, equipment configuration, other systems, the organizational decision process and other organizational issues.

Evaluation should, however, be done in the absence of constraints. If constraints are satisfied before evaluation, an alternative could be eliminated before it is evaluated for its capability of doing the job. Constraints can generally be manipulated to accommodate a preferred alternative.

Evaluation was based on an ideal. Existence of the ideal is a major constraint on selection. No package will match all the requirements. Trade offs and concessions will have to be made.

With the best possible solution a gap will still exist between what is wanted and what is acquired. The object of selection is to minimize that gap. It can be narrowed in three possible ways: relaxing a constraint, downgrading a need or requirement, or through modification of
the package.

The best method is modification. It has the highest probability of closing the gap and, at the same time, improving the overall system. Modification does require commitment of additional time, money and expertise. If those resources are not available then modification is not a viable alternative. Extensive modification may defeat the purpose for originally buying the package.

Relaxing a constraint may allow a better package to be chosen or the best on to be improved through modification.

Constraints can generally be relaxed if no satisfactory solutions exist which meet all of them. This occurs if the value of obtaining the system is worth the extra cost of relaxing the constraint. In some cases, inability to obtain a satisfactory solution implies unrealistic constraints are being placed on selection and that they should be re-evaluated for their feasibility of obtainment. Organizations do, however, have limited resources, and there is a point where constraints cannot be relaxed.

Downgrading a need may imply just adjusting the ideal down to reality and feasibility and may not impair system success at all. A distinction should be made from the start what is essential and what is frills. If the system performs the job adequately without the frill, then it can be downgraded to that point. Downgrading is done either to meet a constraint or as a result of the infeasibility of obtaining some need. It should be done only if it does not impair system success or the system's usefulness.
CHAPTER 5
BOSTON'S PERSONNEL SYSTEM: GENERAL REMARKS AND BACKGROUND

The current financial situation of Boston has created an interest in a computerized personnel system. People are the major resource used in all of the programs provided by the city's various departments, agencies and offices. As a result personnel expenditures make up the major portion of the city's operating budget. Control of these costs would provide a good rein on the entire budget.

Presently there are no well-defined procedures for the current manual system. Lack of controls and procedures has led to abuse in the form of no show jobs; data which is conflicting, redundant, inaccurate or nonexistent; and a collection process which is slow, time consuming and involves a heavy amount of paper work. Preparing reports or obtaining any sort of aggregate data is also a slow and often tedious, time consuming process.

The offices which would be directly associated with a personnel system are in the Administrative Services Department. They are Personnel, Budget, and Data Processing. The head of Administrative Services has given the responsibility of acquiring a personnel system to the Office of Management and Budget (OMB).

OMB got support for the project from the Personnel Office, Office of Labor Relations, Department of Health and Hospitals and the Parks Department. Each felt that their operations would be improved by more efficient data processing and by the information which could be
generated from a personnel system. These various offices and departments were concerned with:

- the massive amounts of forms, data and data processing
- compliance with government regulations
- assessing the impact of labor agreements
- effective and efficient use of personnel
- assessing costs of programs
- manpower planning
- control and monitoring of program expenditures.

These problems can be associated with various functions of personnel management. And as a result lead to the proposed solution, a computerized personnel management system. Automation was proposed because of the heavy volume of transactions, the amount of data involved, and the need for highly accessible data in a usable form.

Personnel management can be divided into six basic functions: selection; operations; labor relations; monitoring and analysis; planning, budgeting and control; and support. Selection are the procedures of recruitment and hiring and involves maintaining an equal opportunity distribution. Day to day transactions are generally record keeping and form processing. This function includes processing attendance, transfers, promotions, and terminations. Labor negotiations are union associated procedures such as bargaining and grievances. Since personnel is a major resource of the city, monitoring and analysis is a very important function. It consists of analyzing jobs and personnel in order to upgrade both the quality of jobs and the work force so that
personnel is used in the most effective and efficient manner. Planning, budget and control includes manpower planning and projecting revenue requirements. Information obtained from personnel data is used to support such functions as wage and salary administration and benefit plans, and to comply with government regulations.

Based on these current problems the goals set by the OMB were:

1) To increase management control and to provide for the more effective use of personnel through the provision of timely and accurate information on positions, skills, experience, costs and benefits.

2) To expand the capacity for manpower planning through the ability to forecast personnel needs and revenue requirements.

In addition the state would like the city to take over its civil service record keeping function. With an automated system OMB felt it would be better equipped to do it.

Before developing a computerized system the present manual system had to be documented and revised. Documentation of present procedures was done in flow chart form. Data was tracked from its origin to final destination showing inputs, outputs and calculations. The purpose was to determine who used what information, how and for what reasons so that unnecessary and redundant information and procedures could be eliminated.

Data enters the system with the hiring of a new employee. It is obtained from application and medical forms and salary and appointment papers. Three types of information are kept. Identifying information includes data relating to appearance and personal characteristics such as
name, address, birth date, height, etc. Departmental information includes
department, job grade, and other data. Historical information is a
record of an employee's past job grades and salaries.

As a result of day to day operations this data is added to,
modified or deleted. The functions of monitoring, analysis and planning
aggregates the data into information that is outputed in the form of
reports or as answers to queries. The outputs are used for planning,
budget and control and to support the related functions.

OMB did an analysis of what would be needed to achieve their goals.
Functions to be covered by the personnel system were identified. Objectives
were stated as reporting requirements, system requirements, needed
calculations and procedures. These needs and requirements are listed in
Appendix B.

OMB elected to buy a package rather than do in-house development.
This decision was based on cost, availability of packages, time, and
lack of available in-house staff.

Personnel, as previously stated, is a common application, and
there are several packages on the market. OMB felt there would be one
which would adequately meet their needs without much modification.
CHAPTER 6
THE EVALUATION AND SELECTION PROCESS FOR BOSTON

Evaluation Criteria

Boston obviously would like a reputable vendor who can provide a quality package to meet all their needs and requirements and provide the services needed for successful implementation.

Boston in their RFP identified eight criteria for evaluation. These criteria with adjustment were used as the basis for determining those factors considered to be relevant according to organizational values and objectives.

The first adjustment made to the criteria in the RFP, was the elimination of the cost and timetable criteria. They do not reflect qualities of the system, rather they are constraints imposed on selection by the evaluating committee.

The first criterion was extended to include the systems capability to meet the functional requirements of a personnel system. Technical was interpreted to mean design features and flexibility requirements to mean design goals like modifiability, expansability and flexibility.

Product warranty was eliminated as a criterion. Warrantees are standard and although they are important, they do not contribute significantly to the success of the system.

The particular needs of Boston require the package design to be especially:

- flexible to fit the different requirements of the various offices, agencies and departments of the city;
modifiable to take into account the new needs created as revised procedures are implemented;
expandable in order to accommodate increases in the master file size as new departments are added to the system;
modular so that the system can be implemented in stages and the city can have the use of reports as soon as possible;
simple to use since the users will not have any programming or technical background.

Ideally the system should be designed so that it has the following features:
- a data base management system, preferably IMS, since the city already has that software;
on line updating and information retrieval;
- 1500 to 3000 characters/record;
security mechanisms to protect the confidentiality of the personnel data;
a flexible report generator that can provide one time and special reports;
a table file system;
accounting controls;
turnaround document;
- limits on range or input data;
- error correct/audit trail capabilities;
labor cost capability.

These features are described in detail in Appendix B.
The needs and requirements essential to achieving the first two goals are identified in the RFP (see Appendix B). They are the calculations and reports necessary to perform the Phase I and Phase II functions.

The personnel system must be able to interface with the city's currently owned payroll system. There are no expectations of purchasing a new system or making major modifications to the current one at present time.

Vendor support is essential since there will be minimal support from the Data Processing Department. The support is so minimal that Boston requests the vendor to supply an experienced programmer/analyst to service in place of a representative from the Data Processing Department.

A maintenance agreement should include updating and enhancements so that the system will be current with "the state of the art" and appropriate laws and government regulations. The city should be required to only have to do minimal maintenance.

Documentation is an important factor in evaluation for two reasons. First, quality documentation is a good indicator of package quality. Second, since the DP staff of Boston will not be involved in installation and implementation, documentation is essential for them to understand the workings of the system.

OMB ranked the criteria in their order of importance to the city. This ranking is reflected by the order of presentation in the RFP. The capability to meet the technical performance and flexibility requirements was considered to be most important.

Weights were assigned based on the ranking given by the OMB with
one exception. The payroll interface is a more important factor in determining the success of the system than design features. A workable system can be made on the basis of the design features of the alternative packages. The personnel system needs data from the payroll file to carry out some of the functions specified in the RFP. Therefore, an interface is essential to meet the needs and requirements of the system.

Screening

A list of possible sources of packages was generated from the IPC Quarterly. The General Accounting and Financial Reporting section contains 31 different personnel packages. In addition, there were 11 packages which performed some function related to personnel management such as job evaluation, government reporting, employment history or position control.

Personnel, payroll and labor functions are closely linked because of the types of data they require. Many packages, therefore, include functions from each of these areas and run them off a single data base.

Packages are set up as a "packaged deal" or as subsystems. For instance, Information Science Corporation offers a custom design approach, a straight package with all options or as subsystem packages containing only a few options.

Initial screening of these descriptions resulted in a list of 10 packages being selected as possible alternatives. The screening was based on a subjective judgement of the description, and whether or not it was written in Ans-Cobol for IBM equipment.
A RFP was submitted to each of the ten vendors. In addition, two companies contacted the OMB. Out of these twelve, two did not respond, five submitted proposals and five withdrew. Withdrawal was based on not having the resources available to commit to a project of that size or not being able to meet the requirements in the RFP.

The Alternatives

The alternatives available to Boston were IBM's Transfer System, the Wang Super Personnel, MSA Personnel Management and Reporting, InSci Human Resource Management and the ADMINS/ll data management system.

IBM does not market application software packages. They provide custom work for their clients. In this capacity they could not acceptably respond to the RFP. However, IBM in conjunction with various city and state governments has researched the accounting information, the information flows and defined the data bases needed for personnel and related functions. Based on this work, they have developed and aided in the design of a system for several state and local governments.

IBM's approach is to first do a system analysis or requirement study for the purpose of generating a general design for a personnel/payroll system. Existing systems would then be investigated for their ability to meet the city's requirements. Some existing system would then be modified to fit the requirements and transferred to Boston. The modification would presumably be done by IBM. With this approach IBM would be in a position to submit a bid.

Analysis would take eight to twelve weeks and would be free. However, IBM states that they must utilize their manpower as well as
possible and the analysis would not be given priority unless connected with some hardware, software or service purchase from IBM.

A possible existing system has already been proposed based on the requirement study done by Boston. The city of San Bernadino, California designed, wrote and implemented both a personnel and a payroll system based on the general design of another city. A lot of in-house work was done, because they transferred only 80% of design and none of the programming. The original programming was "monolithic" in design, and San Bernadino wanted a modular system. It took approximately eight people, a half of a million dollars, a year for personnel and one and a half years for payroll before the joint system was fully operational. The development of the two systems was concurrent but separate.

The ADMINS/11 bid was submitted by a consultant ADMINS/11 is a data management system available only on a PDP-11 mini-computer. The consultant proposes to design a system which will meet all the requirements stated in the RFP, subcontracting out only when appropriate. In addition, he will train the staff in the use of ADMINS/11. The ADMINS/11 approach has been used by five other cities and two city agencies.

Information Science Inc. specializes in software packages for personnel applications. They have experience with government applications, having done work for the cities of Philadelphia and Oklahoma. Of their packages, the one they propose for Boston is the Human Resource System. Its purpose is to reduce the clerical record keeping effort and provide timeliness, accuracy and completeness of reports.

Wang's approach was to provide their standard package with its
associated services. Any additional support or modification would have to be supplied by the city. A consultant was recommended, and his bid was submitted in addition to theirs.

MSA took the same approach as Wang.

Estimating the Consequences

Functional Requirements

The IBM, ADMINS, Wang and InSci bids stated that all the functional requirements in the RFP would be met.

IBM's analysis, transfer and tailoring of the San Bernadino system should give Boston a system which meets all its needs.

ADMINS/ll's file flexibility and report generator can also provide all the data analysis and summarization for manpower planning and forecasting in addition to standard reporting. The consultant plans to work closely with the Boston staff to design the necessary reports.

In addition to InSci's main module, Boston would need the Career Profile and Continuous Employment History modules. Modifications would have to be done in order to meet the position control and salary range requirements.

Wang's proposal states that its standard Super Personnel package will do all necessary reporting and calculations needed to meet the city's functional requirements.

The MSA proposal included not only the central modules but six other ones as well. Their proposal did not include the labor costing capabilities or the benefits function since they were part of MSA payroll package.
Technical Requirements

All the proposed systems with the exception of ADMINS/11 were compatible with Boston's currently owned hardware and software. ADMINS/11 was designed to run on a PDP 11. The minimum CPU required is the PDP 11/34. However, given the number terminals the city wants and its future requirements a PDP 11/70 is more reasonable. The city would be able to use its currently owned peripherals with the DEC equipment.

Package Design and Design Features

The San Bernadino system is modular in design and can be brought up incrementally. It was designed to be very flexible so that changing needs and requirements could be met.

There is no documentation which presents a major problem in transferring. IBM is interested in the system and is willing to do the documentation. The project it can be done in sixty days with two people working on it.

The system is written in COBOL, is online, has an IMS data management system and a report generator.

ADMINS/11 was designed to facilitate the creation of new files and applications. The proposed approach is to bring up an operational system as fast as possible in order to meet immediate needs. Should needs be misinterpreted or changed, files and applications can then be easily modified.

Documentation is provided. The consultant will work closely with the staff so that they will be familiar with the design and operation of the application. If additional aid is ever needed, there are other
specialists in ADMINS/11 besides the consultant.

The system, being a data management system, is of course written in a data base management language and not in either COBOL or IMS. The language and system is supposedly very simple to master and should not present any problems. Other installations have people working with it who are neither computer technicians nor programmers.

The fields of a master file can be edited, added to, deleted or changed in format. The system can support 32 terminals and will also operate in batch. Simultaneous update and query are possible.

The system provides for confidentiality, error checks and data validation.

InSci is a modular package, designed for ease of use. It has a flexible report generator.

All specified documentation is provided with Standard Systems.

The data is structured in a sequential segmented master file with ISAM/VSAM tables. IMS will be not be available until 1978, but InSci will create an IMS data base.

The system operates in batch, but an online data analyzer can be obtained for an additional fee.

The Wang system written in COBOL is of modular design and is simple enough to permit the client to do most of the installation, modification and maintenance. The system is flexible. The master file has variable length records with fields that can be added enlarged or deleted.

The system is designed to run in batch off a sequential file.
Their approach is to bring up the system first in its present form, develop an IMS data base, and then interface them. The system could be modified for online inquiry but not online update.

The documentation supplied includes as part of the standard package a reference manual, user manual, training manual and operator's instruction manual.

MSA, a modular system written in COBOL was designed so that no special programming background would be required to use it. Documentation includes:

- a system manual containing input, output and file elements;
- a program manual to be used for technical clarification and in-house modification;
- a user manual to be referenced by clericals and computer operators for their day to day operations;
- a "how to" manual to train new employees.

The data is accessed sequentially under either VSAM or ISAM. It runs in batch but can also be adapted for online inquiry and update using CICS.

Personnel/Payroll Interface

IBM does not specify how they would treat the interface. There are two possible approaches they may take. First, would be to transfer both the personnel and payroll from San Bernadino, or they may just transfer the personnel and custom design the personnel/payroll link.

ADMS/ll could interface the systems in two ways. One is by transferring needed information on a computer tape. The other is a
hardware interface between equipment, which would provide automatic updating between files. ADMINS/ll has a updating between files. ADMINS/ll has a product function which causes two files to be compared and updated from one another or records to be inserted from one another.

InSci will design, program and implement an interface if Boston specifies what it wants. It proposes some sort of front end program.

Wang proposes using a combined data base. Linkage of files would be difficult. Wang will do none of the design and programming needed for the interface. It is not clear that the consultant will do it either, since it was not stated in his bid.

MSA made three proposals with respect to the to the personnel payroll interface. Two were to use some sort of front end procedure to link the files. MSA would not do the design or programming. The third proposal was to buy the MSA payroll system and convert over. An additional advantage of this would be that those functions dependent on payroll data would now be available to the city.

Installation and Implementation Support

IBM, as part of its transfer, would do the installation, implementation and training.

ADMINS/ll claims that a system meeting all the needs and requirements in the RFP will be fully operational in three months after the arrival of the equipment. It takes two weeks to install ADMINS/ll and a preliminary system to meet immediate operational needs. The three months is for defining the data base and the reports. Complete support will be provided. The consultant stated he will spend most of his time
training Boston's staff and bringing up the system.

Implementation for the InSci system is projected for six months. There is a fairly complete project plan for implementation, installation and conversion.

Twenty-five days of support are provided with the standard package. Support includes a project team of functional and technical experts. The functional experts work with Boston to define package needs. The technical ones coordinate the installation and the interfacing with the other systems.

Two types of training, on the job and classroom, are included in the support program. On the job training consists of the project team working with Boston's system team through all phases of study, design and implementation. This is to insure the system team understands why the system was designed, the detailed components of the system and technical, functional, operational and expansion capabilities.

The class sessions are held for the operators, users and general staff effected by the system. Seminars are provided for continual education so clients will be caught up on the "state of the art."

Wang's standard package includes support in the form of consulting, training and installation assistance. However, installation and training is minimal. Wang claims little assistance is necessary since the package was designed with the idea the client could install it.

Fifteen days are allocated for installation and training. Five days are used for a formal course directed at the supervisor level, user personnel and technical personnel. The rest of the time can be...
used at the choice of the customer any time over the first year.

The consultant provides supplemental support by tailoring the system to meet the city's needs and aiding in implementation. His proposed plan of action is:

- plan and manage system implementation
- review current methods and procedures
- review management information requirements and draft output
- define data elements
- design/modify data collection forms
- document new procedures
- install computer programs
- develop and perform tests
- train staff and implement new procedures
- construct new personnel data base
- documentation

Projected time is seven to nine months depending on amount of resources the city is willing to commit.

This same consultant was recommended by MSA to do the same support. The support supplied by MSA staff is forty hours of installation and training. Additional support can be provided at their standard consulting rates.

Training is a formal five day course held at the client's location. It consists of presentations made to technical and nontechnical staff and to direct and indirect users.
Installation time depends on the number of modules, the size of organization and the amount of assistance from the client. Projected time for installation and implementation of just the main module is three to six months.

Maintenance

IBM is not a package software vendor and does not provide those services associated with vendors. No maintenance agreement is expected from them.

Maintenance can be done by any ADMINS/11 specialist. There is a maintenance agreement for upgrading and debugging. For any major modifications or unusual problems, the consultant is available at the standard rate. No monitoring of changes in government regulations or tax laws is done. No updating of this form will be provided.

InSci's maintenance program is renewable on a yearly basis. It includes upgrading, adjustments, enhancements and rewrites. Upgrading occurs with technological advances or when a customer has a new need. An update, when done, is offered to all other customers of InSci. They have a staff which just monitors the latest government regulations and laws and provides the necessary updates for compliance.

Wang has a yearly maintenance agreement which is provided free of charge for the first year. It includes updates for new laws and regulations, maintenance, enhancement, documentation, updating and keeping up with state of the art IBM hardware and operating systems.

MSA also provides the first year of maintenance for free. After that, there is an annual charge. Maintenance is in the form of a
technical representative assigned to Boston to work out any problems they have.

Based on the previous descriptions of the different factors, ratings were assigned to each package. The ratings and scores are summarized in figure 1.

Looking at the evaluation matrix, the major stumbling block to reaching an acceptable solution is the payroll/personnel interface.

The basic position of InSci, Wang and MSA was that a linked file system would be difficult and inefficient.

It is reasonable not to expect to have to rework present systems every time a new application is added. However, in this case, the two functions, payroll and personnel, are related because of the considerable overlap in data requirements. Separate data bases would create possible integrity and redundancy problems.

MSA and Wang were the only two packages with unacceptable scores for the payroll/personnel interface. By either contracting out or through in-house development, an interface could be custom designed. An estimate of the cost, based on the cost to InSci, is $14,000.

MSA also received a low score for needs and requirements. To improve this score would involve the purchase of their payroll system. This costs $30,000. Purchase of the payroll system would also eliminate the interface problem. Figure 2 represents the revised version.

Costs

The types of indirect costs associated with the packages are the cost of additional modules, consulting costs and modification costs.
<table>
<thead>
<tr>
<th>Factor</th>
<th>IBM</th>
<th>ADMIN$</th>
<th>InSci</th>
<th>Wang</th>
<th>MSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Goals</td>
<td>wt rate score</td>
<td>rate score</td>
<td>rate score</td>
<td>rate score</td>
<td>rate score</td>
</tr>
<tr>
<td>flexible</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>modifiable</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>expandable</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>modular</td>
<td>2</td>
<td>10</td>
<td>20</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>simple</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Design Features</td>
<td>8</td>
<td>10</td>
<td>80</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>Needs and Requirements</td>
<td>15</td>
<td>6</td>
<td>150</td>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td>Payroll/Personnel Interface</td>
<td>12</td>
<td>8</td>
<td>96</td>
<td>8</td>
<td>96</td>
</tr>
<tr>
<td>Vendor Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>installation</td>
<td>10</td>
<td>8</td>
<td>80</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>training</td>
<td>10</td>
<td>8</td>
<td>80</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>conversion</td>
<td>10</td>
<td>8</td>
<td>80</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>maintenance</td>
<td>10</td>
<td>2</td>
<td>20</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>documentation</td>
<td>10</td>
<td>6</td>
<td>60</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Vendor Reputation</td>
<td>5</td>
<td>6</td>
<td>30</td>
<td>8</td>
<td>40</td>
</tr>
<tr>
<td>Total Score</td>
<td>100</td>
<td>736</td>
<td>946</td>
<td>844</td>
<td>806</td>
</tr>
</tbody>
</table>

figure 1.
<table>
<thead>
<tr>
<th>Factor</th>
<th>IBM</th>
<th>ADMINS</th>
<th>InSci</th>
<th>Wang</th>
<th>MSA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wt</td>
<td>rate</td>
<td>score</td>
<td>rate</td>
<td>score</td>
</tr>
<tr>
<td>Design Goals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flexible</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>modifiable</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>expandable</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>modular</td>
<td>2</td>
<td>10</td>
<td>20</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>simple</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Design Features</td>
<td>8</td>
<td>10</td>
<td>80</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>Needs and Requirements</td>
<td>15</td>
<td>10</td>
<td>150</td>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td>Payroll/personnel interface</td>
<td>12</td>
<td>6</td>
<td>72</td>
<td>8</td>
<td>96</td>
</tr>
<tr>
<td>Vendor Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>installation</td>
<td>10</td>
<td>8</td>
<td>80</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>training</td>
<td>10</td>
<td>8</td>
<td>80</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>conversion</td>
<td>10</td>
<td>8</td>
<td>80</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>maintenance</td>
<td>10</td>
<td>2</td>
<td>20</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>documentation</td>
<td>10</td>
<td>6</td>
<td>60</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Vendor Reputation</td>
<td>5</td>
<td>6</td>
<td>30</td>
<td>8</td>
<td>40</td>
</tr>
<tr>
<td>Total Score</td>
<td>100</td>
<td>736</td>
<td>946</td>
<td>844</td>
<td>854</td>
</tr>
</tbody>
</table>

*figure 2.*

56
Cost data was obtained from the vendors' bids. A lot of indirect costs such as cost of conversion were not provided and, therefore, were not included in evaluation.

IBM provides no cost data. The system cost San Bernadino a half of a million dollars. Presumably, it will cost Boston somewhat less, since they would be transferring most of it without modification. The personnel system is public domain, hence there will be no charge for the package itself. The system was developed as part of IBM's transfer program, with the understanding that the cost to another user would be free. Costs associated with this approach would be custom design fees paid to IBM.

The ADMINS/11 bid involves the purchase of a DEC PDP 11/70. Consulting costs are based on forty man months at an average rate of $260 per day. Total Cost is

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDP 11/70 CPU</td>
<td>$50,000</td>
</tr>
<tr>
<td>software</td>
<td>35,000</td>
</tr>
<tr>
<td>consulting</td>
<td>20,800</td>
</tr>
<tr>
<td>maintenance</td>
<td>3,500</td>
</tr>
</tbody>
</table>

$109,300

The InSci package costs $195,950. The breakdown is given in figure 3. Yearly maintenance is an additional $3,850. The total is $199,300.

Wang's standard package costs $37,000. Consultant services are based on 80 man days at $250 per day. This estimate will vary depending on the amount of commitment of Boston's DP staff and other staffs.
## INSCI COST DATA

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STANDARD</strong></td>
<td>$38,500</td>
</tr>
<tr>
<td>Employee Profile and Personnel</td>
<td>28,500</td>
</tr>
<tr>
<td>Career Profile</td>
<td>5,500</td>
</tr>
<tr>
<td>Continuous Employment History</td>
<td>4,500</td>
</tr>
<tr>
<td><strong>MODIFICATIONS</strong></td>
<td>$48,750</td>
</tr>
<tr>
<td>Password</td>
<td>2,000</td>
</tr>
<tr>
<td>Position Control</td>
<td>4,500</td>
</tr>
<tr>
<td>Salary Range</td>
<td>1,500</td>
</tr>
<tr>
<td>Custom Designed Turnaround Document</td>
<td>1,000</td>
</tr>
<tr>
<td>CRT Screen/Display</td>
<td>2,500</td>
</tr>
<tr>
<td><strong>PAYROLL INTERFACE</strong></td>
<td>$14,000</td>
</tr>
<tr>
<td>3,500</td>
<td>9-12,000</td>
</tr>
<tr>
<td><strong>INSTALLATION</strong></td>
<td>$47,650</td>
</tr>
<tr>
<td>Standard Package</td>
<td>16,000</td>
</tr>
<tr>
<td>Payroll</td>
<td>1,200</td>
</tr>
<tr>
<td>Modules</td>
<td>6,450</td>
</tr>
<tr>
<td>Data Processing Representative</td>
<td>24,000</td>
</tr>
<tr>
<td><strong>OTHER</strong></td>
<td>$10,000</td>
</tr>
<tr>
<td>Creation of IMS Data Base</td>
<td></td>
</tr>
<tr>
<td>Data Analyzer</td>
<td>37,000</td>
</tr>
<tr>
<td><strong>TOTAL COST</strong></td>
<td>$195,950</td>
</tr>
</tbody>
</table>

*figure 3.*

58
Total cost is:

- software $37,000
- consulting 20,000
- maintenance 6,000
- interface 14,000

$77,000

MSA system involves several modules. The same consulting services as in the Wang bid will be provided by the same consultant. The cost breakdown is:

- payroll $30,000
- general ledger 3,000
- central 18,000
- position 6,000
- life to date 5,000
- benefits 5,000
- applicant 5,000
- maintenance 3,700
- consulting 20,000

$95,700

Evaluation of Alternatives

Cost/Benefit Calculations

ADMINS/11 with equipment $115.5

without equipment 62.7

InSci 236.7

59
Comparison of Alternatives

The top solutions are ADMINS/11, Wang and MSA. The major differences among them are the purchase of new equipment and how the personnel/payroll interface is handled.

The problem in evaluating the differences is how to evaluate the benefits associated with acquiring either new hardware or a new payroll system. New equipment is necessary for ADMINS/11 to function successfully, but is not a critical factor for Boston to obtain a successful personnel system. The payroll system also has benefits associated with it which do not directly impact the personnel system.

The two MSA alternatives were very close. A cost/benefit was calculated based just on costs and benefits directly associated with each alternative. Buying the payroll package incurs a cost of $30,000 but, at the same time, increases the scores of interface and needs and requirements factors. The front end program was estimated to cost $14,000. Based on this, the ratios are 122 and 86 respectively, indicating that using an interface program is a better alternative.

Constraints

The constraints on the system are cost, time, expenditures on equipment, available DP personnel, the bidding process, and payroll/
personnel interface.

No specific upper bound was set on cost. Expected cost of the system was estimated at $100,000. OMB was concerned primarily with cost as it reflected the capabilities of the package. This was approximated using the cost to benefit ratio. Based on the ratio, Wang, ADMINS/ll and MSA all fell within the same range.

Time is a critical factor because the city operates under changing management. Each new administration brings with it a new set of personnel with new ideas, values and products.

A deadline was set for two reasons. First, results are needed as soon as possible in order to justify the cost, time and personnel being committed to the system. Opponents to the system would like to see resources used elsewhere.

Another reason was to obtain an operational system before requirements or "state of the art" has changed.

Given that a satisfactory solution could not be obtained within the original deadline, it could be extended. The need for the system was not immediate enough to warrant downgrading any needs to meet a deadline.

There was no intention of purchasing any additional hardware except for terminals, CRT screens and other required input/output peripherals. The budget did not allow for additional CPU's. The present equipment configuration had to be utilized.

The only alternative constrained by the present configuration was ADMINS/11. The constraint could be relaxed if the purchase of a PDP was
justified by future growth in computing requirements, or if the city expected to eventually switch over to distributed or decentralized processing.

The in-house DP staff stated that they were already fully committed to other work and would not be able to devote any time to design, development, installation or implementation of a personnel system. This lack of aid will considerably increase consulting cost.

The bidding process is an important legal constraint designed to provide equal opportunity in obtaining government contracts. There are two basic processes, the closed bid and the no-bid award letter. For a closed bid, specs are submitted to the contractors, and conferences are held. The lowest bid is automatically accepted unless the bidders are deemed unqualified. The award of the contract is announced publicly.

With the no-bid system the city can reject any or all of the bids. The no-bid is used in those instances where a list of specs cannot clearly be defined. To waive the close bid and use no-bid, a letter must be obtained from the mayor.

In both cases, a request for proposal (RFP) is submitted, and the contents of the awarded contract are made public.

The bidding process is a constraint on the system since it limits the number of alternatives that can be considered. Only those vendors which respond to the RFP, clearly and completely, in the specified format and by the closing bid date will be considered. Failure to comply with any of the bidding requirements is grounds for disqualification.

The bidding constraint eliminates all the alternatives except Wang
and InSci. IBM could not respond to the bid since it provided only custom work. ADMINS/ll was unable to meet the requirements in the RFP because it operated on PDP's rather than on IBM equipment. MSA's bid was incomplete and, therefore, it was disqualified.

In general, it was felt by Boston's evaluating committee that the bids were unresponsive. It was suggested that this implied that what was wanted and what was needed from the vendor was not clearly defined.

Relaxing the bidding constraint involves rejecting all current bids, writing a new RFP and asking for new bids. This can be done if an alternative presents itself as the best solution. Sending out a new RFP binds the time constraint. Once the new bids are in, an evaluation must be done again, based this time on the new RFP.

As previously mentioned, there was no intention of buying a new payroll system. Any personnel package purchased must be able to interface with the current payroll system.

The general opinion of the Evaluating Committee was that the inefficiencies, the out-of-dateness and the lack of controls of the current system would lead to its eventual replacement. Given this knowledge, replacing the old payroll system with a joint payroll/personnel system becomes a viable alternative. Another possibility would be to run the new personnel system and the payroll concurrently but separately. The personnel system could then be modified in increments to take on payroll associated functions, leading to the eventual phasing out of the old payroll system.
In conclusion, the package with the lowest cost to benefit ratio meeting all the constraints was the Wang Super Personnel System.
CHAPTER 7
CRITIQUE OF THE METHODOLOGY AND ITS APPLICATION

The purpose behind developing the methodology was to provide a set of heuristics which would systematize the selection of software packages. The methodology was designed to minimize the effect of organizational biases and constraints, which may limit the generation and consideration of alternatives. To achieve this aim, it was developed using a rational actor framework based on a normative model which reflects only those factors which directly impact system success. "The rational decision problem is reduced to a simple matter of selecting among a set of given alternatives, each of which has a given set of consequences: the alternative whose consequences are preferred in terms of a utility function which ranks each set of consequences in order of preference."\textsuperscript{12}

Two possible sources of problems with this methodology are that the rational actor model relies on the assumption of perfect information in order to work and that the methodology is attempting to apply structure to an unstructured process. Selection of a package is not an exact science. The process is unstructured because there are no hard fast rules or formulas for determining the "right" package. In general, there is no right answer. Right depends on how the problem and organizational goals, values and constraints are defined.

Organizations do not have the time or the money to generate all possible alternatives. There is also a problem of uncertainty. The consequences of buying any package are not fully known until the
decision has been made and the package and installed.

According to Simon, the problem solving process goes through three phases: intelligence, design and choice. The intelligence phase is problem recognition and identification. Design involves generating alternatives and examining the implications of those alternatives. Choice is the selection among the alternatives on the basis of comparing outcomes across several alternatives. 13

Intelligence was done by developing normative and descriptive models and comparing them. The methodology does not control the intelligence phase. It suggests it. The outcome of this phase provides the basis of defining the weighted evaluation criteria. As a result, the methodology is dependent on a good initial systems analysis in order to eventually make a good decision.

Properly defining the problem is difficult. Problem definition is subject to what Allison calls the political process. There is no single right answer, and there are several people involved in the decision making, each with his own interests and values. As a result, there will be compromise in order to reach consensus. Certain issues will be explored thoroughly and others neglected or glossed over.

Design was the process of screening, estimating and improving. The key objective in this phase is to do everything with respect to the normative model and not to let organizational constraints and biases effect the process. People will have preferences for certain solutions and will structure the constraints so that all but the predetermined solution is eliminated before they are fully explored.
In many cases, alternatives have benefits which are not considered because they are not directly associated with the particular application but may make it a feasible and reasonable choice. A good illustration of this is the ADMINS/ll alternative. It was eliminated because it was not compatible with their present equipment configuration. In light of projected growth and new applications as well as the possible consideration of distributed processing, ADMINS/ll may be a very viable alternative. It provides additional computing power and has the flexibility to design other applications in addition to personnel and payroll.

The choice phase in the methodology consists of evaluating the alternatives, looking at the major differences and calculating the cost/benefit ratio.

The methodology is successful in practice since it does provide a systematic way of approaching the decision and eliminates some of the bias of the decision maker so that all alternatives can be fully explored and possibly lead to the generation of new and better ones. First, it forces the decision maker to think about what is needed for a successful system before he can develop and weigh the evaluation criteria. The evaluation criteria are based on the factor checklist. This list contains only those factors which reflect qualities of the package and the proposed system. Bias is eliminated since the constraints are not looked at until a decision is made based on the normative model. Also, by using the matrix the qualities and characteristics of each package are clearly laid out for comparison.
The MSA alternative presents an example of how the methodology leads to fully exploring and generating a new alternative. After MSA was scored, examination recalled that it was weak in the areas of personnel/payroll interface and in meeting the needs and requirements. First, an attempt was made to try and improve the method of interface. This, however, did not lead to a solution of the needs and requirement problem. Since those needs not met were dependent on the MSA payroll system, purchase of that system was considered. As a result, there are now two alternatives. One uses a front end program to interface the two systems, and the other entails converting over to a new system.

By not being able to obtain perfect information problems arise in the application of the methodology which can lead to a sub-optimal solution. The utility function for the organization which is represented by the weighted evaluation criteria can be fairly adequately but not accurately defined. It is also difficult to obtain the correct and sufficient information from the vendor and eventual users of the system needed to properly assess costs and benefits associated with each solution.

The methodology does not process well those factors which are also constraints. That is, it cannot distinguish between how something should be achieved and what should be achieved. An example of this was the personnel/payroll interface. The requirement was stated as how the two systems should be interfaced rather than we have a payroll system and want a personnel system. What should be done? By asking what instead of telling how, room would have been left open for vendors to
suggest alternative ways of approaching the problem.

The method does not distinguish between different methods. There is no way to determine if the approach taken by one vendor is better than that of another if they both do the job.

Application of the methodology to the purchase of a personnel package for Boston resulted in the Wang Personnel package being selected. The Wang system could adequately do the job within the constraints. This especially true since a consultant will be used to supplement the services provided by the vendor. The consultant will be used to manage the project. It is not clear if there is someone of sufficient authority and expertise at Boston City Hall to assume this role. There are no real outstanding advantages that Wang has over the other alternatives. It was able to meet the needs and requirements without needing any modification to the basic system.

There are problems with the assessing of the costs and benefits which may change the value of this choice to the organization. Cost of consulting to Boston could be alot higher. The original estimate was made based on the assumption that there would be a considerable amount of time commitment made to the project by Boston's DP staff and other essential personnel. This, however, is not the case.

The design and programming of the interface must be contracted out if the consultant cannot do it. This may involve four different parties having to work together.

If indirect benefits and costs are taken into account then ADMINS/11 becomes a strong possibility as a very feasible and reasonable
alternative. First, the system is very flexible with respect to change. Procedures could be completely changed very easily to reflect new needs and requirements. This is not the case with the other packages. The ease of programming of ADMINS/ll permits changes to be done by non-technical or non-systems personnel.

Flexibility to change is very important. It came out in an Evaluation Committee meeting that there was a strong desire to put in a system, any system, and then improve on it. Getting an operational system up and then improving on it was the approach taken by the ADMINS/ll consultant. The structure of ADMINS/ll lends itself to it.

Present information collection and control procedures are very poor. No software package will be successful if effort is not expended to revise and improve these procedures before the purchase is made. Once an analysis of present procedures and revisions are made, it is possible that the problem as well as the needs and requirements may have changed.

ADMINS/ll provided a lot of additional capabilities. Besides being used for personnel, it could be used to design and implement other types of applications. The other packages are strictly personnel.

Additional computing power of a PDP 11/70 comes with the selection of ADMINS/ll. The benefit of this additional capability is not clear since no information was given by the DP staff on the present utilization of the IBM 370/158.

Another possibility which was not considered was the purchase of a Wang payroll system. Given the present state of the current payroll
system, it would be good if the personnel system could be brought up concurrently with the analysis of the payroll system. The new payroll system would then be phased in. ADMINS/11 would allow a joint functioning of the current payroll and the new personnel system without much modification or adaption while analysis of the payroll was being done. The other alternatives involve the design and programming of an interface before this could be done.

In conclusion, the methodology presented cannot make the decision; it can only aid in the making of it. It supports best the design and choice phases. Before purchase the pros and cons of the package indicated by the method should be carefully explored.
CHAPTER 8

CONCLUSION

Software could easily be classified into a Gorry-Morton-like matrix. Across the top of the matrix instead of management activities would be software classifications like systems and application. Each function could then be categorized as structured, semi-structured and unstructured. Structured software could be interpreted to mean functions which have well-defined rules for operation. Semi-structured software implies the function's procedures cannot be made to conform to a standard, but the function is a common enough one that the general design has been fairly well outlined and agreed upon. Unstructured software is either highly specialized or based on an unstructured function. In general, it would have limited application and, hence, very few users.

The method of evaluation presented was designed to be used within range of semi-structured software. It could be modified to be used in the other ranges, but is not as essential for structured or as effective as unstructured.

Most types of structured software has been around for awhile, especially in the case of system software. As a result, there are fairly good publications such as the Auerbach Reports and Datapro Reports which compare and rate structured software packages. Since structure implies not much variability of what is being done, the emphasis is on efficiency rather than effectiveness. An unstructured function has so few users that it would be unusual if there even existed a package which performed the function. The problem with few or no
other users of a package is that there are no references by which to evaluate the vendor and his proposed package. It would be difficult then to fill out the evaluation matrix.

As software goes from structured to unstructured, the risk of not getting a package which will meet the needs and requirements is increased. No matter what type of software package is bought, it still makes sense to define the problem, do a requirement study, evaluate the package and its supplier and do a cost/benefit analysis of the alternatives.

Although the present methodology and evaluation criteria is far from a perfect method of reaching a decision, it does provide a systematic way of examining all the alternatives and realizing the consequences of each. It serves to eliminate alternatives obviously out of line with the rest without allowing the decision makers bias to enter in.

Constraints should not be allowed in any way to effect the screening process. Organizations will change their positions. Software selection has all the characteristics of Allison's model of political process. There is no right answer and the people involved with the decision making process have conflicting interests and values. What they think they want will be the result of compromise and tradeoffs in order to reach a consensus. It is, therefore, important to base evaluation in a normative model of the application and to maintain a rational actor type framework within which to make the decision.

Based on my experience at Boston City Hall, the following suggests itself as a future area of research. The vendor's presentation of his
proposal is very important in biasing the decision. InSci's proposal was very complete and well-organized and, therefore, swayed the evaluators to favor it more on the presentation than the content. As the software industry becomes more stable and, hence, even more competitive, emphasis will be placed on the marketing of the product and not on the product itself. A study could be done on the effects of marketing on the decision making process of software selection.
1. Financial Information System, Executive Overview, IBM Publication.

2. Punj, p.33.


4. Punj, p.44.

5. Punj, p.44.


7. Punj, Chapter 3.


11. Facts regarding the various alternatives were gotten from the suppliers' bids.


BIBLIOGRAPHY


Blee, Micheal, "How to Succeed With Software Programs", Data Systems, October 1974, p. 18-19.


Menkus, Belden, "Computer Software: Make or Buy?", Administrative Management, April 1976, p. 22-23.


SKINNY, International Computer Programs, Inc.


APPENDIX A

This Appendix contains two parts. The first is a set of questions to be answered in order to evaluate a package and its supplier.

Part II is a set of questions to be asked of other purchasers of the package.
PART I

Needs and Requirements

1. How and will package meet present and future needs?
2. Does it have required input format?
3. Does it produce required output reports?
4. Does it do all required calculations?
5. What options does it have? Are they necessary?
6. Can it handle present and future volume?
7. Is it flexible enough to handle changes in procedures, reports and inputs?
8. Can new features be added easily?
9. How easily and successfully will it interface with present systems?
Vendor

1. How long has company been in business?
2. What is their financial status?
3. How many experienced staff?
4. What is their staff turnover rate?
5. Who will work with you at what capacity and to what extent?
6. Will the two staffs get along?
7. How familiar is the company with your industry?
8. Who are their customers?
9. What kind of guarantees can they provide?
10. What kind of staff commitment can they give you?
Technical

1. Will package run on present equipment taking into account presently owned:
   - central processor
   - input/output devices
   - offline and supporting equipment?

2. With what modifications and at what cost?

3. What additional equipment is needed if any?

4. What is the minimum equipment requirement needed to run package?

5. With what equipment configuration will it run most efficiently?

6. How much storage is needed?

7. What programming language is used?

8. Is vendor familiar enough with your equipment, OS, and other software to be able to maintain, modify and update the package?

9. How will it effect present and future equipment requirements?

10. Will package run under present operating system? With what modifications and at what cost?

11. How will it interface with present system?

12. Can vendor give a demonstration on your equipment configuration?
Package Cost

1. What is the direct cost of the package?

2. What does it include? For how long?

3. What services are considered extras? At what cost?

4. What are the costs of:
   - making modifications
   - training personnel
   - maintenance
   - enhancement
   - documentation
   - conversion
   - installation
   - computer time?

5. How was the price of the package arrived at?

6. What financial arrangements are available for payment?
Design

Development
1. Who actually did the design? What are their credentials?
2. For what purpose was the package originally designed?
3. For what machine, OS, and programming language was it designed?
4. Is it done with modular design?
5. Were design goals, machine efficiency or ease and flexibility of use?

Features
1. Does master file contain all necessary fields?
2. What type of file organization and data management is used?
3. What are the input and output options?
4. Are programming techniques complicated or simple?
5. What are the editing features?
6. What type of data validation is provided?
7. What type of errors are checked for?
8. Are error diagnostics clear and concise?
9. Are console messages meaningful?
10. What provisions are made for file protection?
11. How is privacy protected?
12. Are any special forms required?
13. Does it provide for master file growth?
14. What additional features and capabilities does it have? Do they justify increase in cost, size and complexity of system?
Operational Considerations

1. Is the package operational?
2. Has the package been tested?
3. Is it running error free? For how long?
4. Is it efficient and reliable?
5. Has there been any down time due to software problems?
6. How difficult is it to restart?
7. How many people are needed for input and output?
8. What sort of manual set-up is required?
9. What is set-up time for a run?
10. What are the numbers and levels of people required to run it?
11. How easy and time-consuming is data gathering and input?
12. How fast is response?
13. What is normal turn around time?
Support

1. How far away and available are the people qualified to do the support?
2. What type of support will the vendor supply and at what arrangements and costs?
3. Will the vendor provide on-site technical support?
4. Will vendor provide training?
5. What is the nature, level, and location of training course?
6. Will vendor help with conversion and to what extent?
7. What modifications will be done and at what cost?
8. Will bugs be corrected and at what cost?
9. How long will support last?
10. What type of maintenance is provided at what cost and for how long?
11. Will updating be done?
Installation

1. How difficult is installation?
2. How long will it take?
3. What will it cost?
4. How many people will be involved?
5. What is the extent of the manufacturer's aid?
6. What type of training will the people need?
7. Will a recommended procedure be provided?
8. What changes will be required in systems, and procedures?
9. Should there be a gradual changeover with parallel operation?
10. What will be the increase in work demands?
11. How will system be sold to the users?
PART II
User Questionnaire

1. Which package and additional modules did you buy?
2. What modifications were needed?
3. How many people control input and output?
4. How many keypunch operators do you use?
5. What is your system configuration?
6. Do you have any comparable run times?
7. How often do you process?
8. What is normal turn-around?
9. Are you using the vendor's maintenance service and is it satisfactory?
10. What is your in-house maintenance requirement?
11. Was the software available from the vendor on the proposed date?
12. Were vendor representatives available when needed during implementation? Did they perform satisfactorily?
13. Did the implementation consume more time and costs than was expected?
14. Was the education training offered by the vendor sufficient for satisfactory operation of the software?
15. Did the software interface with the operating system as expected?
16. Did the vendor documentation prove adequate for understanding and operating the software?
17. Have the vendor representatives been available for routine systems problems and debugging? Were their services satisfactory?
User Questionnaire (continued)

18. Has there been downtime due to software problems?

19. In your opinion, does the software require more operator intervention than expected? More data control such as EAM requirements, clerks, etc.?

20. How difficult is restart?

21. Were conversion times and costs more than expected in order to make the software compatible with existing modes of operations?

22. Are error diagnostics clear and concise? Is there need for additional error checking?

23. Is the package modular in design such that new features and routines can be added easily?

24. Is data input gathering and preparation relatively easy or is it complicated and time consuming?

25. Are the output reports sufficient to provide management with needed information?

26. Has the software system kept pace with growth in work force, increases in required management data, and any other pertinent information?

27. Generally, has the package performed as you expected?

28. Did you evaluate other packages?

29. Would you buy again from this vendor?
A. General Objectives and Requirements

1. Data Base
Given multiple users of the personnel system, the similarity of data requirements, the variety of information desired and the changing nature of information requirements, the structure which most recommends itself for this system is a data base management system or a flexible data base rather than a fixed format file.

2. Centralized, On-Line Updating/Information Retrieval
The City plans to establish a central I/O unit employing CRT's for the purpose of controlling and performing all updates to the master file. Eventually, CRT's will be placed in all City department personnel offices solely for the purpose of information retrieval. Nevertheless, the system should be capable of operating in a batch environment as well as on-line. If simultaneous update and inquiry is not a capability of the vendor's system it should be clearly stated in the response.

3. Linkage to Payroll System/Single Update Capability
The City is not contemplating the purchase of a payroll package at this time. Given that the City will continue to operate with its current payroll system it is essential that a link be established between the payroll and personnel system such that an entry or update to one would produce an automatic and corresponding entry or update to the other. This structure has overwhelming advantages over separate and unlinked personnel and payroll files for the following reasons:
* Duplication of information is eliminated.

Duplication of maintenance is eliminated. This is extremely important because it is almost a surety that if duplicate maintenance is required, one file or the other will contain different information and therefore the reports from one or both will be inaccurate.

* Computer time is saved because a single run performs maintenance and updates of both sets of data. Given the speeds at which modern computers operate, this savings is significant because the time to process files is minimal compared with the time to start and stop the processing function.

* It is simply much easier for people to administer a linked file system than it is to administer separate files.

* Given the financial reporting capabilities desired from the personnel system including budgeted and actual personnel costs, benefits analysis, salary projections and analysis, etc. it is clear that some operationally feasible method must be established for updating the personnel file from the payroll file as well as updating the payroll file from the personnel file for non-financial transactions.

The vendor should indicate by what mechanism the two files could be periodically compared to ensure the integrity of the data contained in each.
4. **Characters Per Record**
The average record of an employee could range from 1500 to 3000 characters. This includes both present and projected requirements.

5. **Extended Life Span**
It should reasonably be expected that any acquired system will be used by the City of Boston for an extended number of years. The system should be at the state of the art when acquired and should come from a vendor who has an established reputation for providing timely updates and enhancements.

6. **Ease of Use**
The system must be easy to use by Personnel, Budget, Labor Relations and all other user departments. No technical computer knowledge should be required in order to design and produce special reports. This criterion extends to all other facets of the Personnel system as well. Input forms, for example, should be multi-purpose and simple to read and use.

7. **Confidentiality**
The system must be capable of maintaining multiple levels of security in order to protect the confidentiality of personnel data. Respondees must describe the system they would employ to restrict users to only that information for which they or their department have clearance.

8. **Report Generator Capabilities**
The system must provide a flexible report generator that can provide one-time and special reports quickly and economically.
The report generator should be available at the user level and should provide basic arithmetic and logic capabilities as well as listing and cross-tabulation capabilities. In addition, the listing feature must be capable of generating multiple lines per employee and access all database elements. The report generator must be flexible enough to allow the user to manipulate the report formats and to sort output into desired sequence. The capability to produce simple charts and graphs is desired, but not a requirement of the system. The report generator must be able to execute under HASP via remote job entry facilities, under batch using IBM operating (OS) and via terminals using CICS as basic communications protocol.

9. Table File System
The system should arrange much of its information in tables such that changes can be affected in data across an entire category by entering a single element of change within that category.

10. Accounting Controls
The system should be capable of providing accounting controls on key data elements, e.g., Total Personnel - (Terminations + Leaves of Absence) = Today's Personnel Total.

11. Turnaround Document
The system should employ a custom designed (to City specifications) turnaround document for data entry and update. The City also wishes to explore the cost-benefits of designing more than one turnaround document for each department with
special data requirements not suited to a single generalized form. The City welcomes suggestions on alternative methods for dealing with this problem.

12. **File Structure**

The system must have a file structure which will allow for the recording of data and the reporting of information in a manner which corresponds with the City's organizational structure and the varied organizational structures of City departments. The divisions and sub-divisions of the Parks Department, for example, are different from those of the Department of Health and Hospitals, and each should have the capability to organize its personnel structure and data as it chooses rather than having to fit into a prescribed mold. At the same time, however, all desired information must be available to departments and to the City's Personnel Department regardless of structure.

13. **Limits**

The system must be capable of accepting limits and rejecting inputs that would exceed those limits. If the number of vacation days which an employee may have outstanding is set at 14, for example, and that number would be exceeded with the new year's credit for vacation time, the system should reject all or that part of the new credit which would be in excess of 14.

14. **Error Correct/Audit Trail Capabilities**

There will inevitably be some clerical errors in almost every computer run. The accepted system must display all errors per employee on a single listing with meaningful flags and messages for each error. The word "error" alone
is an unacceptable error message. There must be a more meaningful English description of each error. All errors and control totals must be displayed prior to printing. In addition, in order to further facilitate the error correction procedure, a full printout of the original incorrect transaction must be displayed on the error correct/audit trail report. Incorrect fields within the transaction must be highlighted (for example, by underlining with asterisks) whenever possible.

15. **Labor Cost Capability**

The system must be able to produce labor cost subtotals and totals by identifiable cost centers within departments or their subdivisions. These subtotals and totals should be capable of reflecting projected changes in salaries or wages.

**B. System Requirements**

The objective of this section is to itemize the detailed requirements for an automated personnel management and reporting system. The requirements section is broken down into the following subsections:

1. Personnel Reporting Requirements
2. Data Processing Requirements

1. **Personnel Reporting Requirements**

   a. The following list is a sample of the general reporting requirements desired from a personnel system:

   * To report vacancies by department, division, unit job classification, etc.
   * To project step rate increases by department, division appropriation code, etc.
* To provide immediate access to employee personnel/payroll records.

* To analyze the cost of filled and vacant positions:
  - budgeted
  - actual
  - year to date
  - projected estimate to year end

* To analyze the types and costs of the various forms of compensation available to employees.

* To analyze and monitor the amounts and use of overtime by departments.

* To analyze by department and city wide, sick leave and vacation time of personnel.

* To report on new employees, separations and all other transactions.

* To produce Equal Employment Opportunity reports.

* To analyze the insurance, retirement, tuition and other benefits available to employees and their cost to the City.

* To project personnel requirements through retirement and separation.

* To analyze the impact of employee groups on retirement funds.

* To produce salary frequency distribution reports.

* To produce reports on union membership, membership by bargaining unit and by grade and step.

* To provide job history data on active employees.

* To conduct summary analyses of personnel by sex, age, EEO categories, education levels, etc.

* To provide standard reports as required
  - Residency report
  - Civil service seniority report
  - Monthly tally sheet

* To report on the training provided to personnel and the attendant costs.

* To provide summary reports on personnel transactions by department.
b. Each CRT should be capable of displaying an employee's complete record. A hard copy should be available if desired.

c. Various levels of position control are desired:

* The position control file must reflect all positions involved in City service.

  Budgeted
  Vacant
  Civil Service (multiple categories)
  Part time/full time
  Federal (multiple categories)
  Subtotals and totals for all categories

* Each position should be identified by the type and amount of funding given it during the fiscal year.

  Budgeted
  Partially city funded
  Funded by other than city funds
  Dollars used
  Dollars budgeted
  Subtotals and totals for all categories

* Hires, transfers and terminations must be validated against this file before being paid by the payroll system.

The system must be able to apply changes to the above position information or deletion of positions in three ways:

* Specific positions only
* All positions in a department
* All positions in a class or group

d. The system should have the ability to transfer positions from one department to another. Position transfers may be of two types:

* From one department to another department
* From one division or center to another division or center within the same department.
e. The system must be capable of accepting position and classification information for a date in the future. The information would be edited and held until the effective date for processing. All additions, changes, and deletions to the classification and position information must be reflected on a list of transactions.

f. The system must have the capability of monitoring each position classification so that the number of authorized positions is not exceeded. Each position must be monitored so that it is not filled by an employee working more time than was allotted for the position. For example, a full-time position cannot be filled by more than two persons working half time. The system must allow for exceptions to this time rule, however, to allow for practices such as overlapping a new hire and a terminating employee in the same position for purposes of cross-training.

g. The salary range information must be available in a Table format that is easily updated by the Personnel Department. The capability must exist to add, change or delete ranges and steps. Changes to the salary of each step will be of the following types:

* Individual step changes
* Percentage increase to all steps
* Percentage increase to a range or group of ranges
* Dollar increase to all steps
* Dollar increase to a range or group of ranges
h. The system must be capable of changing all data base elements. A mass change in one or several data elements through the input of one card for each element is required. Furthermore, if a change in one element affects other elements these secondary changes should occur automatically.

i. The system should be able to carry and maintain each employee's level of education, years of experience in each job category, training courses completed, special licenses or certificates and an inventory of employee skills.

j. The system must keep a life-to-date history of selected personnel/payroll elements on each employee. As each new line of history is added the former entries should be bumped or shifted down one level.

k. The system should be able to take hourly, weekly or monthly salary rates and fringe benefit costs and project the annual costs.

l. The system must be capable of entering, tracking and maintaining a history of steps in grievance, disciplinary and discrimination processes.

m. The system should have the capability to report on Workers Compensation in the following manner:

1. Management Control Reporting
   a. Case File No. -date injury-OSHA code
   b. Injury Type (code or code triggered narrative)
   c. Incurred cost
   d. Payments made in reporting month
      - compensation
      - medical
   e. Payments to date (F/Y)
   f. Number of accidents F/Y to date (moving average monthly)
   g. Number of accidents previous F/Y to date (moving average monthly)
2. Accident Analysis Report (by department division cumulative)
   a. Injury code
   b. Cause code
   c. Cost by injury

3. Payroll analysis reporting
   a. Payroll by compensation code (department, division, cumulative)
   b. Paid time on leave (department, division, cumulative)

2. Data Processing Requirements
   a. The system must be compatible with the hardware and software operated by the City of Boston without modification to either. The City's Data Processing Department operates the following hardware and software:

   Hardware
   IBM 370/158 Model J1 CPU (3 megabytes of storage)
   24 3330 Model 11 Disk drives (4.8 billion bytes)
   8 3420 Model 5 Tape drives
   2 3211 Printers (P11)
   1 1403 Model N1 Printer (HN2 and AN 2 print arrangements)
   3705 Communications Controller (270 x emulation mode)
   3270 Video display system
   2741 Hard copy terminals
   TWX Compatible terminals (30 and 120 CPS)

   Software
   OS/VS2 Operating system with time sharing option (Release 1.7)
   CICS Communication Control system for application processing
   COBOL/vs
   PL1 Optimizer
   FORTRAN G1 (extended)
VS BASIC
VS ASSEMBLER
MATH/BASIC
STAT/BASIC
BUSINESS BASIC
IMS/VS (batch only)
HASP
SPSS
SYNCSORT

The only modification to the above system currently under consideration is the conversion from 24 3330's to 16 3350's. The vendor must indicate core and storage requirements for his system.

b. Program modules must be written in ANS COBOL.

c. Documentation of the system should include the following:

* User procedure manuals

* System manuals
  Management Summary
  System Technical Summary
  System Operating Description
  System Flow Chart

* Program Manuals
  Job Flow charts
  Input/output interfaces
  File layouts
  File descriptions

* Program descriptions
  Functions
  Structure

* Operations procedures manuals
  JCL
  Re-run instructions
  Error messages
  Data entry instructions
  User request options
  Processing schedule
  User request options
d. The system access method should be specified by the vendor (VSAM, ISAM, HIDAM, etc.).

e. Any licensing rights of the software which would restrict its internal use by the City must be specified.

f. Technical assistance should be available as follows:

* Telephone assistance-available 24 hours a day 7 days a week.

* On site response within 24 hours.

g. An acceptance test of the system must be available as soon as possible after the vendor is selected. This will include execution of all modules against City data created by the vendor and run on the City of Boston's data processing equipment. For this part of the selection process as well as during implementation the respondee will indicate the frequency and duration of computer time desired, storage needs and any keypunching requirements.
A. Proposal Evaluation Criteria

The personnel software package proposals submitted to the City of Boston will be evaluated utilizing a point evaluation system based upon the following elements. The elements listed below are ranked in the order of their importance.

1. The capability of the personnel software package to meet the technical performance and flexibility requirements outlined in this RFP.

2. The level of installation and operating assistance to be provided by the vendor to ensure successful understanding, acceptance and implementation of the complete system. This includes the qualifications of the personnel assigned to work with the City.

3. The ability of the personnel software package to interface successfully with the city payroll system.

4. The projected timetable for installation and the successful operation of the system.

5. References.

6. The total dollar cost in order to render the system fully operational.

7. The nature and cost of product maintenance.

8. The nature and extent of product warranty.