

**FACILITIES PROGRAMMING AND A CASE STUDY OF KWANTLEN
COLLEGE RICHMOND CAMPUS : IMPLICATIONS FOR COMMUNITY
PLANNING**

By

Krista Aileen Cook

B.A. , University of British Columbia, 1988

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ARTS**

in

THE FACULTY OF GRADUATE STUDIES

School of Community and Regional Planning

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

December 1991

© Krista Aileen Cook, 1991

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature

FACULTY OF ARTS

Department of COMMUNITY & REGIONAL PLANNING
School

The University of British Columbia
Vancouver, Canada

Date DECEMBER, 1991

ABSTRACT

In the past thirty years, Facilities Programming has grown and evolved as a discipline, separate from both architecture and planning. Facilities Programming is a profession of such a small and specialized nature that it is not well known or understood by many people, and is not well documented in the architectural or planning literature. This thesis provides a detailed definition of programming, as well as a comprehensive study of the programming of the Kwantlen College Richmond Campus, in Richmond, British Columbia. From a review of selected literature on community planning as well as the author's knowledge of its practice, a comparison is made between community planning and facilities programming which shows that there are many similarities. This study of programming also suggests several aspects that can be applied beneficially to the practice of community planning. These are participation, interaction, staged continuous decision making, and good analytic tools.

TABLE OF CONTENTS

ABSTRACT	ii
LIST OF TABLES	iv
LIST OF FIGURES	v
ACKNOWLEDGEMENTS	vi
CHAPTER 1 : INTRODUCTION	1
1.1 Introduction	1
1.2 Assumptions	3
1.3 Practical Significance of Thesis	3
1.4 Methodology, Sources of Data, Analysis Methods	3
1.5 Organization of Thesis	4
CHAPTER 2 : PROGRAMMING DEFINED	5
2.1 Definition of Programming	5
2.2 Objective of Programming	6
2.3 Process of Programming	8
2.4 Program Participants	12
2.5 Programmer	13
2.6 Programming Information	15
2.7 Product	17
CHAPTER 3 : CHRONOLOGICAL DESCRIPTION OF THE PROGRAMMING OF KWANTLEN COLLEGE RICHMOND CAMPUS	19
CHAPTER 4 : DETAILED PROCESS OF THE PROGRAMMING OF KWANTLEN COLLEGE RICHMOND CAMPUS	36
4.1 General Structure of the Process	36
4.2 Detailed Process	39
4.3 Kwantlen Comments	53
CHAPTER 5 : IMPLICATIONS FOR COMMUNITY PLANNING	55
5.1 Detailed Discussion of Programming and Community Planning Characteristics	58
5.2 Consideration of Ideas that can be applied to Community Planning	69
REFERENCES	74
APPENDICES	78
Appendix A : Table of Contents of Kwantlen College Richmond Campus Facilities Program	79
Appendix B : Component Description of Adult Basic Education Program	81

LIST OF TABLES

Table I: The Richmond Campus Education Plan	21
Table II: The Task Outline	24
Table III: Ministry of Advanced Education, Training and Technology Space Standards	46
Table IV: Summary of Programming and Community Planning Characteristics	57

LIST OF FIGURES

Figure 1: Conceptual Diagram of Programming Process	10
Figure 2: Map of Temporary Richmond Campus	20
Figure 3: Map of Permanent Richmond Campus Site	23
Figure 4: Programming Process Flow Diagram	37
Figure 5: Detailed Process Flow Diagram	40
Figure 6: Excerpt from Space-Cost Model for the Richmond Campus	49
Figure 7: Excerpt from Utilization Model for the Richmond Campus	51
Figure 8: Conceptual Diagram of Programming Process	61
Figure 9: Conceptual Diagram of the Community Planning Process	61

ACKNOWLEDGEMENTS

**Thank you
Professors Brahm Wiesman and Peter Boothroyd
for your time, insight, and commitment.**

**Thank you
Jim, Susie, Jiuna, Paul, Sue, Tom, Mom, Dad
for believing in me.**

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the past thirty years, Facilities Programming has grown and evolved as a separate profession from both architecture and urban planning. Facilities Programming is a profession of such a small and specialized nature that it is not well known or understood by many people. “Facilities Programming has been defined as the process by which criteria are developed for the design of a space, building, facility, ...” (Canadian Handbook of Practice for Architects 1976, 2). The primary purpose of programming is to lay out the building requirements and functions clearly for the designer. It is the means through which information about the needs of the building users are determined and expressed for the instruction of the designer. In simple terms, programming information tells the designer how many spaces there should be in the building, what type of spaces there should be, how large they need to be, what is in each space, and what space needs to be next to what space. The Facilities Programming document describes for the designer how the building should work. Programming may also include qualitative aspects desired in a building. For example, a program may describe more than a functional arrangement of spaces, it may also describe qualitative concerns such as image, ambience, and character.

Planning has been broadly defined as “a process by which an individual, group, organization or community decides what it wants (goals) and how it is going to achieve these goals (strategies) (Boothroyd 1989, 1). The Webster’s Dictionary defines planning as “a formulated scheme setting out stages of procedure” (The New Lexicon Webster’s Dictionary of the English Language 1988). A plan outlines how an individual, group, or organization intends to respond to externally or internally driven change. Thus, facilities programming can be defined as a planning function, since an organization goes through a facilities programming process to arrive at a plan that outlines clearly how the organization will function and work in a new environment, as well as a physical plan that explicitly communicates the desired physical arrangements of space in a new or renovated facility.

Recently, Kwantlen College completed two facilities programming projects in the Lower Mainland; the Surrey Campus and the Richmond Campus. The planning of the Surrey Campus was a negative experience for the majority of the faculty and staff at the college. The planning process used for the Surrey Campus was described as unorganized and uncontrolled. Faculty did not understand the process, and they were unclear about their role in it. Faculty and management were mostly excluded from development of information for the Surrey Campus, but were instead presented with a programmed solution by the consultant. This process left them feeling unheard and not in control. Planning for the Surrey Campus was not set in the realm of budget constraints and other real life considerations. Because of this, cuts to space were required throughout the design process. Faculty were left disillusioned and somewhat bitter (Peter-Cherneff 1991). After this experience of planning the Surrey Campus, the Richmond Campus was seen as a new chance by the management of Kwantlen College. There was a desire to involve faculty to a greater and more meaningful extent. Management wanted the faculty to describe their own needs, have some control, and yet plan within realistic constraints.

The purpose of this thesis is to analyze how and to what extent facilities programming may be applicable and significant to community planning. What can we learn from facilities programming that is useful for community planning? To achieve this, programming and community planning are compared using categories derived from community planning and facility planning literature.

Firstly, Facilities Programming will be defined by answering the following questions. What do the terms programming and program mean? What is the scope of programming, how does it work, and what are its products?

Secondly, the case of Kwantlen College Richmond Campus will be used to describe and explain facilities programming from two perspectives, organizational and analytical. From an organizational perspective, how did the programmers of the Richmond Campus interact with the many different actors involved? Who was involved and consulted and why? What level of participation was achieved and how was it achieved? What was the decision making process? How were different interests and issues resolved? From an analytical perspective, how did the programmers gather and manipulate information? What was the nature of information development and methodology? How was information used to make decisions?

Thirdly, I will analyze the extent to which programming is applicable to community planning. What is the significance of programming to the community planning field? How is the organizational interaction and the information processing employed in the

programming professional relevant to community planning? What can community planners learn from programmers?

1.2 Assumptions

Programming is accepted as a discrete, professional field and it is a planning function. I am assuming that there is some similarity between community planning and facilities programming, and Chapter 5 shows these similarities.

1.3 Practical Significance of Thesis

The purpose of this thesis is to uncover, consolidate, and document descriptive information about the work of the programming profession. The thesis will afford readers insight into the programming profession, and will inform planners interested in entering the programming profession. In describing facilities programming through the use of a case study it will be better understood in practice. Programming is significant for community planning, as the programming process and the community planning process have many similarities. Some of the concepts of programming will be applicable to the community planning field. I feel that there can be some transfer of training between facilities programmers and community planners.

1.4 Methodology, Sources of Data, Analysis Methods

The first part of the thesis consists of a detailed definition of facilities programming. There is not much literature on programming, and indeed, no comprehensive explanations of the programming process. There are no case studies. Sources of data for the case study of the facilities programming of the Richmond Campus were key people who were involved in the project. Approximately 12 Kwantlen College faculty were interviewed as representative respondents. Brigitte Peter-Cherneff, chairperson and key contact for the Richmond College project, acted as my key informant at Kwantlen College. Interviews with respondents provided information on the organizational perspective of programming. How did the programmer and participants interact? How and on what basis were decisions made? To obtain a richer data base, I also interviewed the programmer for the Richmond Campus. Interviewing this consultant provided me with information about the analytical side of programming. Documentation of the project was helpful, and I accessed the files of Brigitte Peter-Cherneff and the programmer to piece together parts of the process that may have been forgotten.

1.5 Organization of Thesis

This thesis is organized into five chapters. Chapter 1 contains introductory information relating to the purpose of this thesis, its context, and how it is organized. Chapter 2 provides context by reviewing important literature regarding facilities programming. This chapter outlines the definition of facilities programming as found in the literature. Chapters 3 and 4 describe the facilities programming for Kwantlen College Richmond Campus. These chapters comprise the main body of information provided by Kwantlen College informants and the programming consultant. Chapter 3 describes the chronology of the programming case study, while Chapter 4 concentrates on the analytical perspective. Chapter 5 reflects on the finding that community planning can learn something from programming in terms of participation, interaction, staged continuous decision making, and good analytic tools.

CHAPTER 2

PROGRAMMING DEFINED

The purpose of this chapter is to describe and present a comprehensive guide to programming. Major ideas in the programming literature were reviewed, analyzed, and organized and are put forward in this chapter. The main source for this chapter was The Architect's Guide to Facility Programming by Mickey A. Palmer (1981). Secondary sources included writings by Harold Horowitz (1969), F.E. Preiser (1978), B.H. Evans and C.H. Wheeler (1969), Henry Sanoff (1977), and William Peña (1977). Secondary sources used for the chapter were difficult to find and were not very informative. This chapter presents a composite of the programming literature. In order to fully define the nature of programming, it will be examined within a conceptual framework. This conceptual framework comprises seven sections : definition of programming, objectives of programming, process of programming, program participants, programmer, programming information, and products of programming. At a general level, the definition of programming will be outlined first to orient the reader. The objectives of programming, or the purposes, will be reviewed. The programming process section will present how programming is done. Next, program participants, that is the people who are involved in the programming process, will be identified. The programmer, in terms of his/her skills and educational background, will be described. Programming information will explain the types of information contained in a program. Finally, the product, the program document, will be addressed.

2.1 Definition of Programming

The purpose of this section is to examine the definition of programming. What is programming? A comprehensive definition of programming will be outlined in this section.

Programming is broadly defined in the literature as information processing. "It is defined as a process of identifying and defining the design needs of a facility and communicating the requirements to the designer" (Palmer 1981). Agnostini defines

programming as a “...coherent, meaningful compilation of the facts needed to create facilities which will most effectively support the client’s operations and organizational goals...” (Agnostini 1986). Palmer agrees that “...it all comes down to information...”. “A program is information the designer can use. It is an organized collection of specific information about the client’s requirements which the architect needs in order to design a particular facility” (Palmer 1981). Since the program communicates the needs to the designer, programming also involves communication. “A program is communication. It transmits and interprets the needs of client to the designer” (Palmer 1981)

The most simple yet enlightening explanation comes from White, who argues that “...programming is a matter of planning before we act and really not different in intent from making a shopping list before buying groceries, planning an itinerary before a trip or rehearsing a play before opening night...” (White 1972). “Simply stated, programming is getting ready for design” (White 1972).

The key word in defining programming is information. Programming involves the collecting and processing of specific information about human, physical, and external factors which will influence design. This information is needed by an architect in order to design a facility. There are many problems that come along with the collection and processing of information. For the programmer, these present grave problems, including the organization of reams of information, the collection of relevant information, and the documentation of conclusions.

2.2 Objective of Programming

The purpose of this section is to explain the objective of programming. Why do we program? Objectives in programming include: to supply information to the architect and client, to express the client’s requirement, to state the design problem, to serve as a log or a record of the programming process, to serve as a communication link, and to encourage greater participation. The objectives might differ from project to project, and probably from programmer to programmer. Each of these objectives will be described in turn.

Palmer states that the “...primary purpose of programming is to supply the information needed by the architect for design...” (Palmer 1981). However, the objective of programming is more than obtaining and organizing information : it also includes developing and producing the information as well. As well as gathering the program information, the programmer must synthesize, analyse, and document that information in a communicable form to the client and the designer. The program must serve the information needs of both client and designer. It should contain as much data as the designer requires for creative judgement, without overwhelming the lay client with technical details. The

program must speak the designers' language while explaining terms to the client. The program must provide information and criteria for the designer. It acts as a information resource for the designer. In other words, the designer uses the information in the program to design a building. The designer can delve deeper into the design problem with a program, as the information needed for design is outlined in the program for him/her. Designers use a program as a guide to the design criteria which must be fulfilled, as a source of data for preparing design solutions, and as a reference for making design decisions. Used as a guide to the design criteria which must be fulfilled, the designer will check design solutions against criteria outlined in the program, such as dimensions and quality of environment concerns. The program will also be used as a reference for making design decisions, because the program is a definitive statement of client requirements that the design must fulfill.

A program should also express client requirements. The program should translate requirements that are implicit in facility operating plans and programs into explicit requirements for a particular facility. The program should state objectives of each group in terms of goals they wish to achieve, the issues they want to resolve, and the problems to be corrected. The programmer should make sure all needs and desires of the client are elucidated, that alternatives are considered and that the design implications of different goals are understood. The client's mandate, along with spatial requirements, should be expressed in the program.

Programming is figuring out and stating a design problem. A program must fully describe the problem by stating the underlying needs of the building and the clients. This involves investigation and analysis of project design needs. Put another way, programming is uncovering and stating the problem that the architect will solve through design.

A program also serves as a log, a record, and a progress report of the programming process. A program may be distributed in several iterations, or drafts. These drafts will convey current information on the progress of the project at its various stages of development. Program drafts will show the designers, as well as the client how the project is coming along. "It serves as a log, a memory and a set of conditions..." (Sanoff 1977).

Palmer also notes that programming serves as a communication link between client and designer. "Programming helps achieve effective communication in information development and in decision making" (Palmer 1981). It enables both client and designer to understand and agree on the design goals, project needs and criteria for design. Programming communicates client needs to the architect. The programmer's role is to communicate, either in verbal or written form, the program information to the designer.

Sanoff argues that programming can encourage greater client participation as well as user feedback (Sanoff 1981). It stimulates clients to make decisions. It provides a method for decision making and a rationale for future decisions. Programming can encourage user participation by getting building residents directly involved in the programming exercise. Participation means the involvement of people with complex behaviours and communication, thus the programming process can become complex. A number of people with different and frequently conflicting objectives and values are likely to exist and it is the programmer's job to resolve conflicts between the users. This, of course, can create a highly political atmosphere for the programmer. However, there is general agreement in the participation literature that more participation results in more feelings of satisfaction, competence and creativity in the users and in the programmer. As users become involved, more precise ideas about facility use and function become apparent. The result of user participation is that programs more precisely and reliably reflect the real needs of facility use.

2.3 Process of Programming

The purpose of this section is to try to explain the process used for programming. How is programming done?

As seen by reviewing the literature, there is no single standard procedure for programming. There seem to be many individual methodologies. The best method is that which works for each individual professional firm. Individual methods are developed and refined to match individual's work habits and philosophies. All the methods follow a similar general process which was put forward by Palmer. This process explanation will be outlined in this section. All the methods have several common characteristics ; they are systemized, iterative, and progressive. Each of these common characteristics explained in detail following.

The programming process is iterative. Programming is a process of repeated cycles of collection, organization, review, and evaluation. Large amounts of complex data accompany sophisticated programming. This data is accumulated in cycles of collection. Once enough factual information is acquired a written statement is reviewed, evaluated, and responded to by the client. In other words, once the programmer feels he/she has gathered enough data, a document will be prepared. This document will be read by the client. The client will respond to this document, and may direct the programmer to review or revise the document in certain areas. This enables the programmer to expand and refine the program information in a logical, useful manner. It gives the programmer a clear direction of how to proceed to the next stage of work.

Programming is also a progressive process. This means that programming proceeds from the general to the specific. The programmer builds the program towards a definitive, specific statement. The programmer does not know at the start of the project what information will be important or included in the program document. It is only through iterations with the client that the information needs become more focused and narrowed down. In other words, programming is progressive because the programmer starts with general information and during the course of the project the programmer works to strictly define the information needs.

The programming process involves systemization. That is, programming is a system of procedures developed by individuals through experience, trial and error. Systemization enables programmers to process information rapidly, reliably and economically. Systemization is required, as sheer volume of programming information may become unwieldy. Each programmer has a system that he/she developed to deal with the amount of programming information that accumulates. For example, micro computers may be used. Information taken in note form with the client group may be transcribed quickly into a micro computer, and this information will be used as a data base. In a data base, information can quickly be sorted, organized and used in a document.

Palmer outlines a general view of the programming process, which will be reviewed here. All programmers follow this process, however, it would most probably be adapted by them (Palmer 1981). Palmer states that the programming process is information processing (Palmer 1981). There are five information processing functions: Collection, Analysis, Organization, Communication, and Evaluation. Figure 1 illustrates the programming process stages and how they cycle. This diagram was composed to graphically illustrate the steps. Each function is interdependent. No step occurs in isolation from the other steps. Each of these information processing functions will be explained in detail.

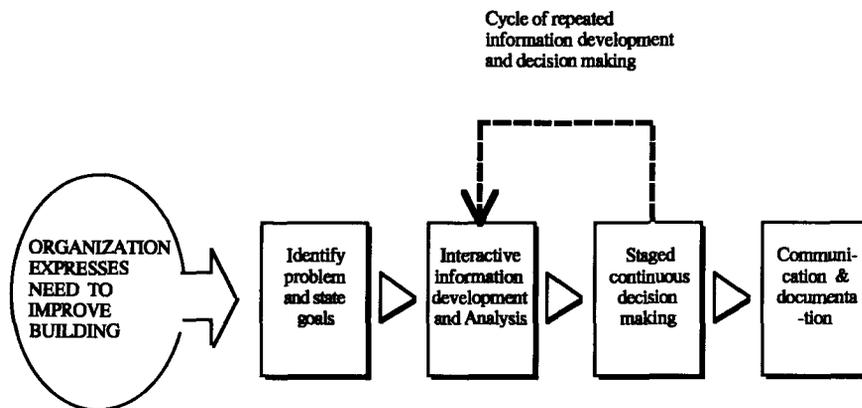


Figure 1: Conceptual Diagram of Programming Process, from Process Four Design & Aitken Wreglesworth Architects, Interactive Design : A Better Way to Involve Users (Vancouver, BC: Process Four Design & Aitken Wreglesworth Architects July, 1991) 5.

The first stage in the programming process is the data collection stage. The purpose of this stage is to gather information needed to produce a program. Collection also includes setting goals and priorities, which may be done by the client before the programmer's involvement. As the programmer moves along in the programming process, data collection becomes focussed and missing information is filled in. Data collection includes generating and eliciting data from client and other resources. This stage may involve getting information from clients, photos, sampling, observational studies, and questionnaires. Clients may be interviewed to provide important qualitative details about spaces and space needs. Photographs of the existing spaces may be used to provide details about furnishings, equipment, and sizes of spaces. The programmer may also decide to gather information through use of sampling, observational studies, and questionnaires. Observational studies occur when the programmer collects information by watching the client group in their existing space. The programmer may distribute questionnaires to the whole client organization to gather information, or the programmer may simply sample a small representative group from the client organization.

The second stage in programming is called analysis of data. The purpose of this stage is to separate raw information into relevant and manageable categories. There is typically a great mass of data obtained in the previous data collection stage. In the analysis stage, this large mass of raw information is sorted, compared and interpreted in order to determine the validity of that data. The programmer attempts to illuminate the significance of a mass of data by organizing the total into manageable portions. The programmer may organize the data by developing some preliminary conclusions about the type of data needed. The type

of data needed in the program will be compared against the type of data collected. The programmer also may test the relevance and validity of individual pieces of information to the larger programming project. This may result in determining the need for additional data. In the case of numerical data, the programmer may use statistical analysis to manage and manipulate the data. By using analytical and statistical procedures, the data can be reduced to manageable portions of a useful quality.

The third stage in programming is that of organization. The purpose of organization is to order the information into parts that make sense in terms of all the information. Organization is forming a whole from interdependent parts, in order to visualize it as a whole. As noted before, the stages seldom occur in a specific order. Organization can occur at the outset, when data objectives must be ordered and identified. Organization can help in generating and analyzing information in a consistent and comprehensive manner throughout the data collection and analysis stages. Organization must occur in order to analyse, and analysis of data is necessary for effective organization. After data collection, data are also sorted. Organization can occur at the end of programming, when conclusions are put into meaningful order.

The communication stage ties the programming process together, by making the programmer's work available and meaningful to program audiences, and by linking the various participants. Oral communication, as well as communication through written documentation, is critical. Communication is important between the client and the programmer, as the programmer must understand the client's needs and the client must understand what data the programmer needs. Communication fosters information exchange, and enhances effective decision making. Communication is critical in both effective development and presentation of information. It is also important in interaction between programmer and participants. Some communication aids that may be used by programmer are:

- Brainstorming- a free flowing worksession where ideas are generated, not discussed;
- Synectics- a technique using analogies, metaphors, similes to develop ideas;
- Buzz / Rap sessions- technique to freely talk in groups about information;
- Gaming- any number of creative games can be used to enhance communication; and
- Group planning- where client groups strategize together about the future.

The programmer may use the following communication techniques in documentation and presentation:

- Narrative- illustration of the program through use of narrative, or story telling technique;
- Graphics- audio visual means to illustrate and communicate the program information;
- Audio / Visual aids- using three dimensional models as visual cues;
- Oral presentations- program can be described in speech to client; and
- Workbooks- throughout the process, the developing program can be described in a written form designed for response.

Palmer's final stage in programming is evaluation (Palmer 1981). The evaluation step consists of the client reviewing the conclusions reached by the programmer, and pointing out gaps or revisions. Conclusions about facility needs should be evaluated by the client, the programmer, and the designer. However, this is not the final step in the programming process. Once the gaps are highlighted by the client and by the programmer, the whole cycle may be repeated from any step in the process. A succession of progressively refined program conclusions may be the end product, as the cycle may be repeated several times. Design begins as soon as the programmer has a final draft of the program. There may be some overlap between the programming and the design stages, but this depends on the particular architect and programmer and how they work together.

2.4 Program Participants

The purpose of this section is to explain who is involved in the programming process. Who uses a program? What are some of the interest groups involved in a programming exercise, and why are they involved?

The client or sponsor of the project is the main figure involved in the programming process. The client has a responsibility to provide program information that can be developed in a number of ways. The client must work with the programmer to provide direction, information, and decisions. Heath argues that the client organization has a role in the setting up of an information and decision network, and to ensure that the network is diffused down into the organization so that every decision can be taken as far as possible by the people most affected by that decision (Heath 1974). The final program is used by a client in deciding the feasibility of a project, determining if facility needs have been adequately addressed, and for making project and budget authorization decisions. Together, client and designer use the program to monitor and evaluate the design as it develops and to reconcile space and budget conflicts. The client can use the program's stated requirements as a check against the designer's solution. If the designer has not kept within budget and space constraints as stated in the program, the client can point to the program as the authority.

Users, as part of the client group, are also involved in the programming process. Users are defined as the people who will use the end product, the facility. User participation refers to attempts to incorporate the actual or prospective users of a facility into the planning and design of that environment (Becker 1977). The degree of user participation depends on both client wishes and the programmer's process. Users may provide information, as well as be part of the final audience for the program document. Users are involved because some people feel they are more competent to provide design information, since they are involved in the use of the building on an intimate day to day basis.

Various other interest groups can be involved in a programming project. Their involvement is based on the particular type of project. These groups include architects, other professionals, city officials, regulatory bodies, the general public, contractors. These groups are generally involved as information sources for the program information. However, depending on the particular project, they may be an audience for the final program document.

The problem the programmer faces in dealing with the different groups is how to ensure compromise occurs. It is the programmer's job to ensure that all groups are heard and have a say, and to facilitate compromise between groups. This is most difficult to accomplish.

2.5 Programmer

The purpose of this section is to examine who a programmer is, and the skills required to be a programmer. Who produces the program? The producer of a program is called a programmer. The programmer might be the client, the architect, or a specialized programming consultant.

The client, that is the person or group of people building a facility, may do their own programming. The production of the program is the clients' responsibility, whether the client does it or hires someone else to do it. In the case of a simple, straightforward building the client may do the simple programming.

An architect may produce the facility's program. This may occur as part of the building design stage, or as part of a predesign exercise by the architect.

A professional programmer may produce the program. "An independent programmer can perform the work on behalf of either the facility owner, or the architect" (Palmer 1981). The programmer is usually hired by the facility owner, but may be brought in by an architect on an unusually complicated facility. Programming is a specific service. Palmer states that there are non-architectural programming specialists that concentrate on specific

building types, such as hospitals or educational institutions (Palmer 1981). The architectural design team may have an in-house programmer, or retain an outside programming consultant. Programming needs have been met over the past several years by a small group of technicians and planners working as consultants under the label of “architectural or facilities programmers”. The educational background of this group is mixed, and includes architects, psychologists, sociologists, and planners.

What are the skills required to be a good programmer? Programming, because of its complex nature, is often the work of a multi - disciplinary team, because no one individual can possess all the skills needed for a complex problem. Required disciplines include architecture, management, psycho-sociology, computer technology, and engineering. A team must be skilled in architecture and planning to develop architectural ideas and planning concepts for the building. It is important that a team has an able administrator to deal with the production and documentation of the myriad of information. Involvement with participants demands a team have skills in psycho-sociology to work with groups of people. A team must also be knowledgeable in computers for producing spreadsheets and other documents. Complex projects may required engineering expertise for structural, electrical, and mechanical building systems. As pointed out by Evan & Wheeler (1969), it seems apparent that good programming often stems from a programmer’s prior experience in programming and even in other disciplines. Good programming results from four criteria, including the programmer’s capability and competence to use standard procedures, systematic procedure, data banks, and feedback information. These four criteria are the most important qualifications to be a good programmer. Programming skills may be possessed by people trained in other disciplines besides architecture, including planning, engineering, and computer technology.

Horowitz believes that there are professions better suited to programming than architects (Horowitz 1967). He argues that architects acting in a programming capacity are operating outside of their own scientific community. Horowitz believes that the behavioural sciences and architecture can gain from collaboration. Behavioural sciences can even offer an improvement of programming procedures, especially in techniques of collecting and organizing information. Therefore, programming is not exclusive to the architect’s domain. It can be performed by planners, engineers, space management consultants, and interior designers.

2.6 Programming Information

The purpose of this section is threefold. In the first subsection, the types of information in a program will be outlined and some examples will be provided. In the second subsection, the scope of information needed for programming will be presented. The last subsection outlines the sources of information to be consulted after the programmer has determined types and scope of information necessary. What type of information does a programmer collect? What level of detail is collected? The main problem in programming is the large amount of data that can be collected, and then must be sorted by relevancy. Another problem is the organization of these large amounts of information.

This subsection will explain and give examples of the types of information found in a program. It is hard to categorize the type of information needed in a program, because information is varied from program to program. Also a program must satisfy the designer's and client's information needs, which may be two different things. The client needs budget and cost information. The designer needs site, functions, space, and proximities information.

Palmer states that most programs address three basic types of information ; human, physical, and external (Palmer 1981). Programming must be comprehensive, that is, address as many factors as possible from all three categories. Human factors are behavioural aspects; for example, the program may call for a counselling office to be designed as a "warm and sympathetic environment". Community and public requirements may be addressed within this also. Physical factors include space types, functions, space requirements, space relationships, special equipment and systems, and site requirements. Each space would be described in terms of the functions performed in the space, the actual size of the space in square feet, as well as detailed information about the space's relationship to other spaces in the building. Detailed information about electrical, water, and heating and ventilation servicing would also be included. General information about the site may also be provided. External factors include codes, standards, and regulations.

It is important to indicate that there is no one comprehensive list of the information types to be included in a program. That is because all programs are different, depending on the program requested by the client and the building type. The program information provided will differ according to the type of program desired by the client, and the level of detail appropriate. The level and types of information will vary depending on the programmer's approved budget: a more comprehensive job can be done with a larger

budget. The client may not want a lengthy, detailed program; he may wish to quickly get into design.

The scope of information the programmer collects refers to the level of detail to be incorporated into a programming document. “Whatever data are necessary and relevant to the efficient, effective, design of a facility that accommodates the needs of the client are the concerns of the programmer” (Palmer 1981). This means the the scope of design information required is the scope of programming information. The scope should include specific factors that will have prominent influence on the facility. Identifying the factors that have the most significant impact on the facility may give the programmer direction for investigation. There is no specific process to determining the most significant factors, the programmer own judgement and experience must be used. The designer’s and client’s information needs and how they will use the program may determine the scope of the investigation of program information. Ideally, any program would gather all information which limits or influences a facility’s design. These statements all provide a wide latitude for interpreting the scope of a specific program. “However in reality the purview of a program is determined by the programmer’s inclination or routine, and the client’s interests, time, and \$” (Palmer 1981).

The most important and difficult part of programming is deciding what information is necessary and relevant to the design of a particular facility. Time and money constrain the scope of investigation. For any particular project, the programmer must select what is most relevant and important. The scope of investigation depends on the nature of a project, and the programmer may address one category, or focus on a single issue. The significance of this is that the design may be adversely affected if the program scope is unduly constrained.

How does a programmer go about collecting the information once knowing the scope and type of information required? “If the planned facility is a replacement or a renovation, the existing facility is one of the principal resources for the programmer” (Palmer 1981). Direct observation of existing facilities provide a data base for programming. Construction drawings, floor plans, previous programs, computerized inventories of existing space and equipment will all be useful. The client may also have mandatory standards, records, reports, and promotional literature, which will all be important information sources. The programmer may even take photographs of the existing building to use as information sources. Similar facilities may be observed, along with their drawings and diagrams, to provide useful information. The programmer’s own experience with the facility type may provide useful data.

Another primary source of information is the client. The client may provide data, but the programmer must work to elicit data from client. The client may direct the programmer

to gather information from people using the existing building. These people are called users. The programmer must employ techniques of gathering relevant information from the users. Users are important sources, as they have direct knowledge of a facility's contents, functions, and problems. Users have realistic information about how operations are performed, how interaction occurs, the equipment and furnishings required, and how organizational structures affect space and arrangement. The programmer may elicit information from the users by using a questionnaire, interview, observational study, or other techniques of direct data collection. A questionnaire is a list of questions compiled by the programmer about programming information to be filled out by the client organization. Questions may concern information about furnishings, and servicing. The programmer may ask these questions through an interview, or simply by watching the client organization in its existing building. The programmer may also use certain research materials for information sources. A library is a good source of information for the programmer.

Local authorities, such as planning, zoning and building code officials, utility companies, and government agencies may provide useful information. The programmer may retain consultants and other experts to address specific aspects of programming, as in the case of cost consultants.

2.7 Product

The purpose of this section is to outline the final product in the programming process. What are the final products of the programming process? What does the final program contain? What are its conclusions?

The final product of the programming process is a document called a program. Agnostini says that a program is a comprehensive report that presents, in text and tables, the detailed qualitative and quantitative requirements of the entire client organization. The end product of programming process is information. The program should always provide adequate, appropriate information that clearly communicates the specific needs of that project so that the designer can satisfy those needs. The actual program is the culmination of the information development process, and is the ultimate stage in programming.

Programs will differ greatly. The content of a program, the form of the information, the format of the communication will vary tremendously. The program will depend on the project, the client, and the programmer. There is a great variety in program types, as the application may be broad or narrow, simple or complex. The program will contain facts and conclusions distilled from raw data. The content of the program will depend on the

nature and complexity of the project and on the information the programmer has agreed to provide to client.

The final program should state conclusions. Conclusions may be expressed in terms of direction for the designer, space descriptions, design concepts, and spatial criteria. These all communicate ideas and requirements about the building's space to the architect. Conclusions define issues and present the concepts that must be addressed in design. The program must clearly present conclusions in conformance with the methodology and terminology of designers. That is, the program should speak the designer's language. The program should extract only essential conclusions and information.

In conclusion, programming concerns the collection, organization, and communication of information. Sources of information, types of information, and detailed programming processes will vary from programmer to programmer, and from project to project. Details and purposes vary, but the general pattern or direction of activity is consistent. The general process used in programming is similar from project to project; these process steps are Collection, Analysis, Organization, Communication, and Evaluation. However, the literature leaves some doubt as to detailed examples and descriptions of processes. The literature provides no real life examples or case studies of a programming project. It is not clear how programming is done by a professional programming consultant. It seems that programming is not a simple, mechanical process that can be explained easily, otherwise the literature would have presented a detailed process explanation. It seems the programmer must build up his knowledge and experience of the profession in order to develop his/her own personal programming process. However, there are some important conceptual ideas and tools that a programmer should know before adaption of the process begins. In order to further delve into the detail of the programming process, a case study of a programming project will be examined in the following two chapters.

CHAPTER 3

CHRONOLOGICAL DESCRIPTION OF THE PROGRAMMING OF KWANTLEN COLLEGE RICHMOND CAMPUS

As evidenced by Chapter 2, there is not much literature on programming, and indeed, no comprehensive explanations of the programming process. There were no case studies found in the literature search. It is important to review a case study to impart a better understanding of programming. Chapters 3 and 4 will study the project of the programming of Kwantlen College Richmond Campus. The purpose of this chapter, Chapter 3, is to present chronologically the events that took place in the the programming of the Campus. The programming took place from May 1989 to February 1990. Chapter 4 will explain the tools and the processes the programmers used in this project.

Kwantlen College's Richmond Campus is temporarily located in rented accommodation in Richmond at Elmbridge Way and Cedarbridge Way. (See Figure 2: Map of Temporary Richmond Campus). The space is somewhat limiting and was never intended to function as a permanent campus. The Richmond Campus will continue to function out of rented accommodation until September of 1992 when it will occupy the newly constructed permanent space.

The development of the Kwantlen College system was considered by many plans and other documents prepared by Kwantlen College staff, all of which proposed a permanent Richmond Campus. A Five Year Plan 1989 -1993 , prepared by Kwantlen College in 1989, called for the building of the permanent Richmond Campus. This Plan contained the Educational Plan for the new Richmond Campus outlining all the programs to be taught at the new Campus along with the forecasted student full time-equivalents to be taught. (See Table I: The Richmond Campus Education Plan)

This Educational Plan called for the new Campus to accommodate all the existing Richmond educational programs, increasing enrollment for some of these programs, and the relocation of some educational programs from other Campuses to the Richmond Campus, as well as implementing some new educational programs.

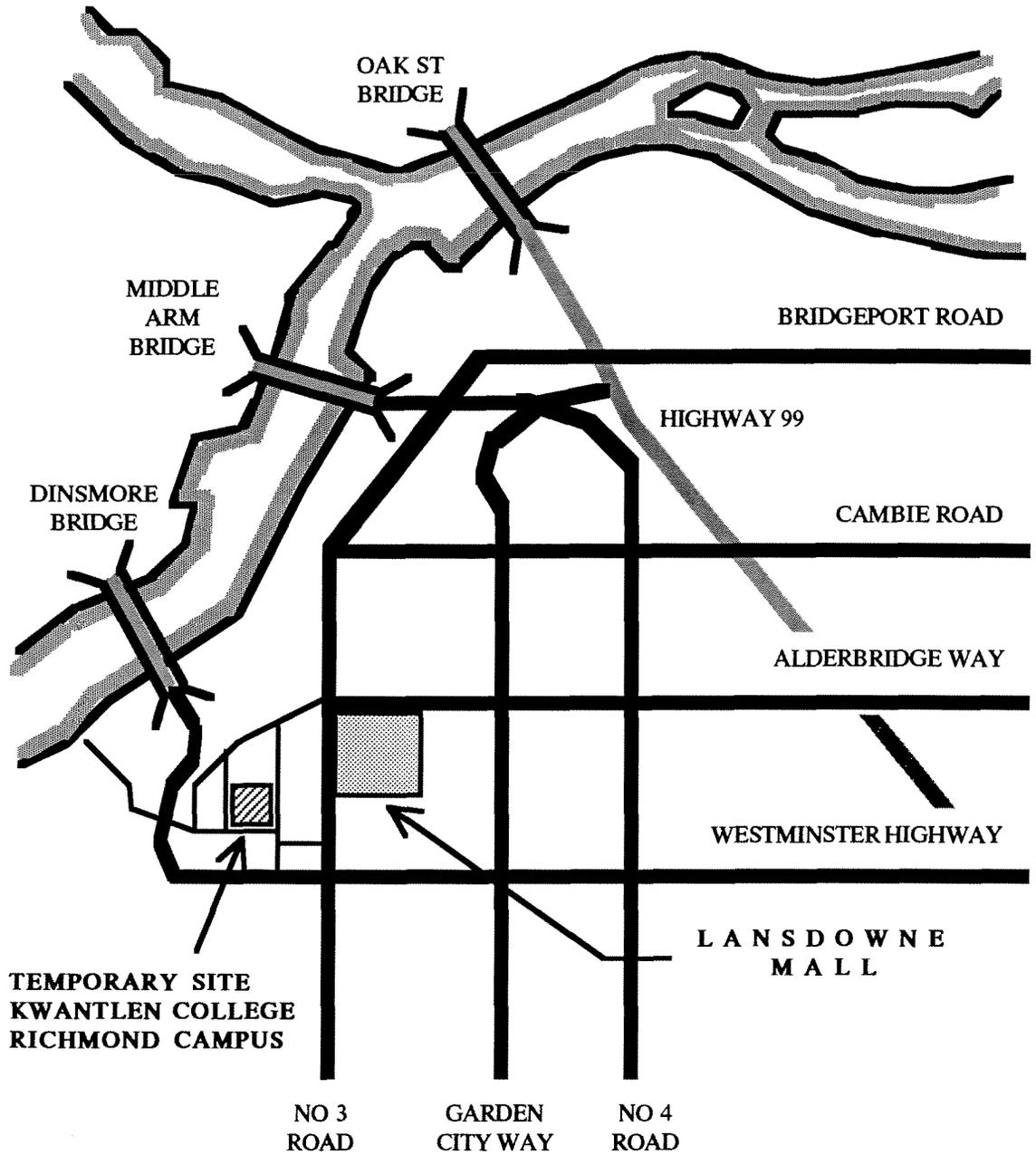


Figure 2: Map of Temporary Richmond Campus, from Kwantlen College Richmond Campus Facilities Program (Vancouver, BC: Process Four Design, February, 1990).

THE RICHMOND CAMPUS EDUCATION PLAN

	Number of Full Time Equivalent Students		
	Fall 1988	1991/92	Change
University Transfer			
Arts	223	223	-
Science	146	146	-
Subtotals	369	369	-
Business	231	252	21
Adult Basic and Special Education			
Adult Basic	48	67	19
Adult Special	21	22	1
Vocational Skills	12	12	-
Subtotal	81	101	20
Applied Design			
Design (new)	0	45	45
Fashion	75	75	-
Foundations (new)	0	24	24
Graphics (transfer from Newton)	0	75	75
Interior	44	44	-
Subtotal	119	263	144
Health and Human Services			
Child Care	16	16	-
Vocational			
Drafting	43	43	-
Office Administration	92	92	-
Part-time Vocational	40	40	-
Journalism	24	24	-
Subtotal	199	199	-
TOTAL	1015	1200	185

Table I : The Richmond Campus Education Plan, from Five Year Plan, 1989-1993 (Vancouver, BC: Kwantlen College, December 1988).

Educational planning for the new Campus took place during 1987-1988 within the Kwantlen College system. It was not until 1988 that action began to occur outside of the institution. Kwantlen College officials began to meet with the Ministry of Advanced Education to secure initial approval and planning funds for the new Campus. During early 1988, the Ministry asked Kwantlen College to illustrate a staged approach to developing the Richmond Campus. In the Autumn of 1988, after further consultation with the Ministry, the College began planning Richmond Campus as single project. Planning and programming money was obtained from the Ministry. The project budget was fixed at \$37.2 million in 1988 dollars, including money for construction, furnishings, fees, contingency, and parking.

A site for the new Campus was chosen at Lansdowne and Cook Roads. (See Figure 3 Map of Permanent Richmond Campus Site). This site is adjacent to Lansdowne Mall, a fully enclosed retail mall separated from the subject site by open parking lots. The site was acquired by the client through a process conducted independently of the programming process.

After securing a site for the new Richmond Campus, Kwantlen College sent out a request for proposals to programming consultants. Process Four Design, one of the programming consultants interested in the project, had an introductory meeting with Uli Haag, the Dean of Human Resources and Organization Support on February 17, 1989. During this meeting, Process Four Design made an effort to find out more about Kwantlen College and the Richmond Campus project. Subsequently, Process Four Design submitted a proposal for the project and was chosen for the interview short list. Shortly after Process Four Design was interviewed by Kwantlen College, the firm was chosen as programming consultants for the Kwantlen College Richmond Campus. On March 23, 1989 a letter from Kwantlen College was received by Process Four Design confirming their engagement for a fixed fee.

Process Four Design was hired by Kwantlen College because their philosophy had a good fit with that of the College. Both believe that users have valuable knowledge and that they can be used as valuable sources of information. Process Four Design were also chosen because of the tight time frame for the programming. Process Four Design was not working on any other major projects at that time and could work within the time frame set down by the College, which was to program a new Campus during April 01, 1989 to September 30, 1989, even though many faculty would not be available during these summer months.

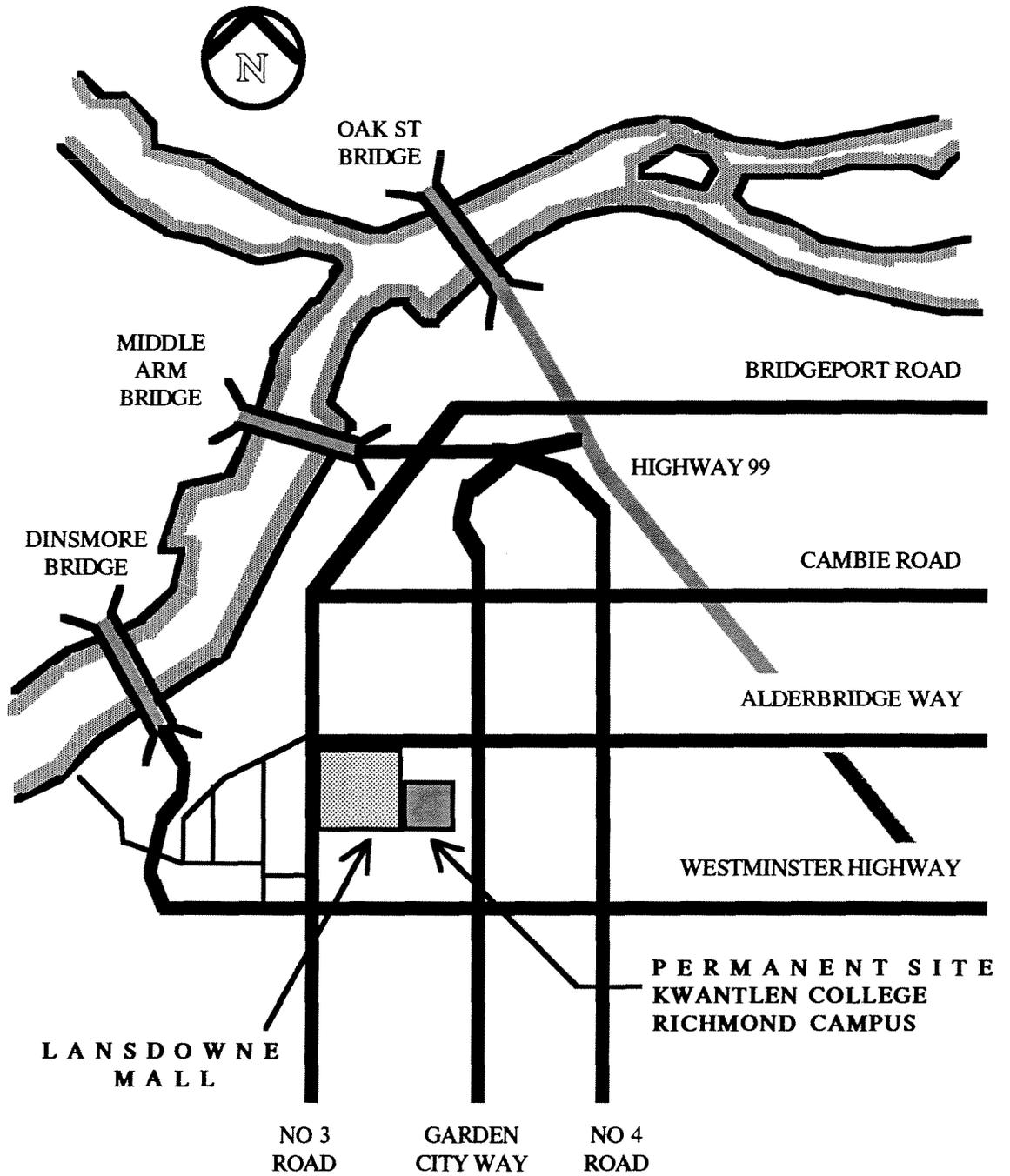


Figure 3: Map of Permanent Richmond Campus Site, from Kwantlen College Richmond Campus Facilities Program (Vancouver, BC: Process Four Design, February, 1990).

THE TASK OUTLINE

		Day Allocations	
		Principal	Senior Programmer
1.0	PROJECT INITIATION	April Weeks 1-2	
1.1	Establish contractual relationship	1.0	
1.2	Orientation and review	2.0	2.0
2.0	ROLE/PARAMETERS	April Week 2 -June Week 1	
2.1	Worksessions		
	Directing Committee	1.5	1.5
	User Committees	5.0	5.0
2.2	Working Paper #1-Role & Parameters	0.5	3.0
2.3	Review and Revise	0.5	1.0
2.4	Working Paper #2 - Issues	0.5	1.0
3.0	SCHEMATIC DESIGN PROGRAM	June Week 2-September Week 1	
3.1	Worksessions		
	Directing Committee	1.5	1.5
	User Committees	5.0	5.0
3.2	Standard Space	1.0	4.0
3.3	Enrollment & staffing	2.0	3.0
3.4	Planning Concepts	1.5	2.0
3.5	Space Lists	1.5	6.0
3.6	Space/Cost Model	2.0	3.0
3.7	Working Paper #3- Schematic Design	2.5	5.0
3.8	Review and Revise	0.5	2.5
3.9	Working Paper #4- Space List	2.5	5.0
4.0	SITE PLANNING	August Week 2-September Week 1	
4.1	Outline Planning Criteria	0.5	1.5
5.0	DETAILED PROGRAMMING	July Week 3-September Week 3	
5.1	Worksessions		
	Directing Committee	2.0	2.0
	User Committees	5.0	5.0
5.2	Working Paper #5- Space Data	2.0	5.0
5.3	Review and Revise	0.5	1.0
5.4	Working Paper #6-Directing Committee	0.5	1.0
6.0	FINAL DOCUMENTATION	September Weeks 1-2	
6.1	Prepare draft final document	1.0	3.5
6.2	Review and Revise	0.5	1.0
7.0	DESIGN LIAISON	September Weeks 3-4	
7.1	Block Planning Liaison	2.5	2.0
7.2	Schematic Concept Liaison	2.5	2.0

Table II: The Task Outline, from Sumi, Jim, Principal of Process Four Design, Letter to MacLaughlin, Adrienne, President of Kwantlen College, March 06 1989.

The first step for the programmers was to produce a Task Outline. The Task Outline, reproduced as Table II, lists all the tasks that needed to be completed, as well as a day allocation for each one. This chapter will explain the Richmond Campus programming in terms of this task outline. Each major group of tasks will be listed, and a detailed description of the group will follow.

April Weeks 1-2

1.0 PROJECT INITIATION

1.1 Establish Contractual Relationship

1.2 Orientation and Review

Kwantlen College Richmond Campus programming began in the first two weeks of April. The tasks performed during this time are found on the Task Outline under 1.0 Project Initiation. During these first two weeks, the consultant established a contractual relationship with the Kwantlen College organization by working with Uli Haag, the Dean of Human Resources and Organization Support. The programmer did this by working with the Mr. Haag to come up with a mutually acceptable budget and task outline. Both parties entered into a legally binding contract, the programmer promising to do the work as outlined, and the College promising to remunerate the programming firm. Orientation and review also took place over these two weeks. The programmers spent time looking at existing documentation including organization charts, Kwantlen College plans, and the Surrey Campus programming and architectural plans.

Kwantlen College did some initial work at the start of the programming process. They began to set up their control systems, roles, and mandates at the direction of the programmer. Uppermost in their minds was the programming of the recently constructed Surrey Campus. All group roles, membership, mandates, and operations were formally laid out by Kwantlen to avoid repetition of problems of the Surrey Campus programming. (MacLaughlin, Adrienne, Letter of 20 March 1989). Certain objectives were outlined to the programmer at the start of the project. The decision makers at Kwantlen wanted to achieve some objectives in the programming of the Richmond Campus. The College wanted to smooth over bad feelings remaining from the planning of the Surrey Campus, to have the process set within the existing decision making structure of the Kwantlen system, to have faculty involved in a useful way, and to not have faculty take control of the process.

To achieve this end, the Directing Committee, the decision making body of Kwantlen College, identified all the groups and individuals to be involved in the programming, and described their roles and mandates. As set out in a memo of April 6, 1989, the Directing Committee's mandate for the project was to:

1. To make all the required administrative decisions to ensure that the project proceeds expeditiously and remains within the financial resources allocated to the project.;
2. To receive reports from the Project Manager and the Chairman of the Users Committees;
3. To consider the recommendations of the functional planner (Process Four Design), architects (Aitken Wreglesworth Associates), and Project Manager (UMA Spantec);
4. To make recommendations to the Facilities Committee and the Board.

Decision and direction actions are based upon information generated by participating staff.

from MacLaughlin, Adrienne, President of Kwantlen College, Letter to Jim Sumi of Process Four Design, April 06 1989.

The Membership of the Directing Committee , as outlined in this memo, comprised the Vice President of Education for the Richmond Campus, the Dean of Educational Support, the Vice President of Administration and Finance, the Dean of Organizational Support, the Project Manager, and the President of Kwantlen College. The Project Manager was a hired consultant from UMA Spantec, a construction management firm. UMA Spantec was not involved during the programming part of the process.

The Directing Committee also established memberships in the User Committees, and their mandates. The role of users was to act as information sources about their own areas and how they work in the College. Their mandate for the Richmond Campus project, as described in a Kwantlen memo of April 06, 1989, was:

1. To be a liaison for their representative group to convey information to and from the group; and
2. To provide advice to the College regarding space requirements; space lay out; furnishings and equipment.

from MacLaughlin, Adrienne, President of Kwantlen College, Letter to Jim Sumi of Process Four Design, April 06 1989.

There were five User Committees in total, each representing an educational or administrative organizational grouping to be in the new campus. The five Committees included:

- Group 1 Applied Arts and Career Preparation;
- Group 2 Design Centre;
- Group 3 Academic and Applied Studies;
- Group 4 Educational Support; and
- Group 5 Administrative Services.

Each Committee contained five to seven people, in order that all the departments and programs to be included in the new Campus were represented. These representatives could consult with their colleagues who were not on the Committees.

A Chairperson's role was also created. The Chairperson's role, as documented in the memo of April 6, 1989, was :

1. To prepare agendas for the meetings.
2. To organize regular meetings to receive the advice of the members.
3. To produce and distribute minutes.
4. To provide regular reports to the Directing Committee.
5. To work closely with the Project Manager to ensure that all the required information is transmitted to the functional planner and architect.
6. To provide information requested by the Project Manager or Vice President Education.

from MacLaughlin, Adrienne, President of Kwantlen College, Letter to Jim Sumi of Process Four Design, April 06 1989.

The role of the programmer was to work with the User Committees to produce a program document for the Richmond Campus. The programmer was to work most closely with the Chairperson to schedule meetings, and distribute information and working papers. The programmer was to act as an information and communication link between the User Committees and the Directing Committee in order that all people were kept up to date and informed of decisions and directions. Thus, from the very outset of the project, all groups and individuals taking part in the programming process were clearly defined in terms of their membership and their roles.

April Week 2 -June Week 1

2.0 ROLE / PARAMETERS

2.1 Worksessions with Directing Committee and User Committees

2.2 Working Paper#1 Role and Parameters

2.3 Review and Revise

2.4 Working Paper #2 - Issues

From the second week in April until the first week in June, the consultant moved directly and quickly into the programming work. Two worksessions were conducted with each User Committee. The purpose of the worksessions was to develop Warts, Winners, Visions, and Opportunities lists with the User Committees, along with information on User Committees' anticipated roles within the College on the opening day of the new Richmond Campus. Developing Warts, Winners, Visions, and Opportunities lists with the User Committees was used as a warmup exercises for the users. Warts were things wrong, or things people did not like about the existing Richmond Campus. Winners were things right with the existing facility, that should be carried over into the new facility. Visions and Opportunities for new Richmond Campus gave the programmers ideas about spaces, images, and how the new Campus should work. For example, a Wart of the existing Campus was the poor heating and ventilation system. A Winner of the existing Campus

was the opening windows. Visions and Opportunities for the new Campus included ideas like having a student cafeteria, a faculty lounge, and adequate parking.

The User Committees also developed functional roles and parameters for each department or program in the new Richmond Campus. These functional roles and parameters included information about the anticipated functions, enrollment, and staffing on the opening of the new Richmond Campus. Role and parameter statements provided descriptions of services and activities each educational program will fulfill. The roles and parameters were described in current terms, as well as in terms of the changes that are anticipated in the future.

The information developed with the User Committees, (Warts, Winners, Visions, and Opportunities lists, and Role and Parameters about each department and program), was documented by the programmer and sent back to the User Committees in Working Paper #1. Working Paper #1 was distributed at the beginning of June, 1989. Revisions were noted by the users directly on this paper and were sent back to the programmer for incorporation into a Revised Working Paper #1.

However, some issues had arisen from the worksessions and Working Paper #1 comments. Working Paper #2 was compiled by the programmers to address all the issues that needed to be resolved before programming could continue. Working Paper #2 contained issues brought up by the User Committees as a result of Working Paper #1. Working Paper #2 was distributed June 1989 to the Directing Committee, the decision making body of Kwantlen College. The Directing Committee reviewed key role and parameter assumptions, and responded yes or no to each issue. The Directing Committee consolidated their responses and gave a revised edition of Working Paper #2 back to the consultant with decisions. Although the main issue to resolve at this stage was establishing the role of each department in the new building, time was also spent with the Directing Committee refining limits for staffing, and student counts; for example, one of the issues that came up was the number of Full Time Student Equivalent numbers that each area or department would teach. The User Committees felt that the ratio of the student numbers was incorrect between departments. They came up with a revised set of numbers that they felt would work correctly. This was forwarded to the Directing Committee in Working Paper #2 and approved.

- June Week 2-September Week 1
- 3.0 SCHEMATIC DESIGN PROGRAM
- 3.1 Worksessions with Directing Committee and User Committees
- 3.2 Standard Space
- 3.3 Enrollment and staffing
- 3.4 Planning concepts
- 3.5 Space Lists
- 3.6 Space / Cost Model
- 3.7 Working Paper #3 Schematic Design
- 3.8 Review and Revise
- 3.9 Working Paper #4- Space List

During the second week of June to the first week of September, Process Four Design conducted individual worksessions with users in order to produce a space list, neighbours list, internal relationships diagram, staffing, and detailed space information for each department or program. Also at this time, Process Four Design communicated decisions that the Directing Committee had made about the issues to the users. Part of the programmer's job was to be a communication link between the Directing Committee and the User Committees. The programmer communicated information between the two entities.

Worksessions began with the users giving the programmer a quick tour of the existing spaces in their area. The user would talk about what would be required in the new Campus and compare it to the existing spaces. From this, the programmer would begin to talk with the user about space required to fulfill the previously determined role of the department or program. A space list was the product of this process.

In these worksessions, a neighbours list was also generated by the user. Neighbours defined those other departments or programs in the Campus that the user needed to have close, and the reasons why. The programmer did this by simply asking the user.

As the space list had been generated by the user, the programmer took pre-cut paper circles ("bubbles"), and write a space name on each "bubble". The user would then be asked to organize their spaces by configuring the "bubbles". This was to determine the internal relationships of the spaces, that is, how the spaces should be physically organized. What space needs to be next to what space and why? This process was done for each user group, as well as for the overall relationship of user group to user group.

Staffing was also determined in this worksession. The programmer obtained information regarding how many staff were required to teach each program, or to support each department at the determined enrollment, in order to correctly allocate office space.

Process Four Design documented the information from the worksessions in Working Paper #3, which was distributed in the middle of July, 1989. This working paper included

information for each department or program on neighbours, internal relationships, previously determined role, staffing and enrollment. The purpose of the working paper was to document in writing the information developed in the worksessions. The users reviewed and revised the information directly by writing on the working paper, and then sent it back to Process Four Design.

Based on the information developed in each worksession, and the tours of existing furnishings and spaces, the programmers also produced an initial space list. This space list was checked against the spaces in the new Surrey Campus to ensure that no space was overlooked. On August 15, 1989, what the programmers called a “quick and dirty” space list was distributed to the users listing the names of spaces required. This was revised by the users and quickly returned to the programmers.

On August 29, 1989, Working Paper #4 was distributed. This working paper consisted of a revised space list with the number and names of spaces without square footage assigned. This space list was based on revised space lists returned by the users. The number of spaces required was based on user needs and was ascertained over the phone. However, this would need to be checked against various parameters and revised.

During this time, the programmer was also processing space information in the complete Campus space list in order to develop Space Standards. Space Standards needed to be developed for staff offices, staff workstations, classroom workstations, and lab workstations. The purpose of developing space standards for spaces that repeat a lot throughout the building is to eliminate redundant information in the program. The programmer did not want to copy office requirements for every single office in the new Campus. Space standards were developed by looking through the space list for patterns and using the patterns to develop the standards. For example, all faculty offices were shared and seemed to be about 160 Square Feet. A 160 Square Foot Shared Faculty Office became a standard.

During the month of September, more detailed work on the space list and associated costs was done. Process Four Design developed a space list with attached square footage. Allocations for spaces were based on tours of the existing Campus and more worksessions with the users. Cost information was gathered from Barnett Treharne Yates BC Limited, a cost consultant. Costs per square foot per space type were developed by the cost consultant. Cost information was inserted into the space list spreadsheet; that is, unit costs were linked to units of space in a dynamic spreadsheet model. This model was dynamic because if a change in space size occurred, the spreadsheet would quickly recalculate the total project cost.

Process Four Design also worked with the Dean of each Educational Department to develop the Utilization Model. The Utilization Model was a numerical computer model that showed how the new Campus would work in terms of number of courses and number of teaching rooms. In order to begin collecting information for the Utilization Model, the Consultant gathered information over the phone from instructors and Deans about courses. Each program was outlined in terms of courses, the number of students per course, the instructional hours per week in each course, the number of students in each course, and in what type of space. All information about educational programs was put into a computer spreadsheet to build a model to ensure the correct number and types of spaces had been planned. This information models how the new Richmond Campus would work in terms of all programs and space use. In September 1989, this utilization model was sent to Gerry Kilcup, the Dean of Applied Arts and Career Preparation for review. Gerry Kilcup, also a member of the Directing Committee, ensured the information collected was correct.

August Week 2-September Week 1
4.0 SITE PLANNING
4.1 Outline Planning Criteria

From the second week of August to the first week of September, Process Four Design collected information on the new Campus site. Municipal codes and bylaws as established for the site were examined for factors relevant at the programming stage. These factors included site coverage (the area of site that can be covered by building), setbacks (the distances from property lines that buildings may be located), and zoning (the designation of the site relative to Municipal zoning bylaws). Discussions with municipal staff also addressed the potential for flexibility in determining final site constraints; for example, the number of parking spaces to be provided was to be determined as a part of the design process.

This information was collected at a general level, to cue the design team in its research, as well as to identify factors that represented constraints at the programming stage. In this case, no such constraints were identified.

July Week 3-September Week 3
5.0 DETAILED PROGRAMMING
5.1 Worksessions with Directing Committee and Users Committee
5.2 Working Paper #5 Space Data
5.3 Review and revise
5.4 Working Paper #6-Directing Committee

From the third week in July until the third week in September, the programmer collected detailed programming information. Some of this had already been collected during worksessions and tours of existing spaces with the users, however, follow up and more detail was required in complicated areas. More worksessions were held with the users, for purposes of inventorying existing program spaces and furnishings. The existing temporary Richmond Campus was used as a starting point for technical and furnishings requirements. Each type of furnishing required was measured and documented. At this time, enough information had been collected to assign a square footage to each space with substantiating detailed programming information. The detailed programming information was documented in Working Paper #5 Space Data. Detailed programming information, or space data, included diagrams of furniture, equipment, layout of special spaces, and technical requirements of a space. It was special, detailed information about specific spaces, developed by the programmer in sketch form to arrive at the assigned square footage of a space. Users quickly responded to the Working Paper #5 and sent revisions.

Now that all detailed programming information had been collected, documented, and revised by the users, the space list could be completely revised. At this time, the space list was within the budget parameters set by the Ministry of Advanced Training and Technology. The programmers needed to inform the Directing Committee of Kwantlen College with an update of progress, as well as get them to endorse the information that the users and programmers had developed to date. All information was ready for final documentation pending the endorsement of the Directing Committee. Working Paper #6 was prepared for the Directing Committee with all information developed to date. The Directing Committee working paper included all educational programs and departments that would go in the new Campus. Each had a role description, enrollment, staffing, future changes, internal relationship, and neighbours. In addition, a complete space list, a space/cost model, and a utilization model were included. This Paper was forwarded to the Directing Committee in September 1989. A project meeting was held with the Directing Committee to discuss the space program. At this time, the Directing Committee approved the documentation. The go ahead was given to the programmers to produce a final program document.

September Weeks 1-2
6.0 FINAL DOCUMENTATION
6.1 Prepare draft final document
6.2 Review and Revise

At the direction of the Directing Committee, the programmers set to work to document and produce all the information for the final draft program during the first two weeks of September. A draft of the final program was issued to the users, the Directing Committee members, and the project architects at the end of September, 1989. The draft program document included the following sections: Document Use and Organization, Site Information, Project Parameters, Planning Concepts, Space Types, Standards Spaces, Program Summary, and Component Descriptions. See Appendix A : Table of Contents for the list of sections that made up the Final Program for the Richmond Campus. The Document Use and Organization section provided information about how to use the program document. The Site section provided general information about the site of the new Campus, and any constraints in site development. Project Parameters outlined the limits within which the design problem should be resolved. From the earliest stages of the programming process, parameters separate out the issues and requirements which are part of the problem from those which are not part of the problem; for example, a Project Parameter statement for the new Richmond Campus was "The provision of specialized facilities will be given first priority." The Space Types section described the number of different types of spaces in the new Campus, and described the characteristics of each. The Standards Spaces section attempted to define the repeating spaces in an effort to minimize redundancy later in the program document; for example, a faculty office was described in this section in order that it need not be described numerous times throughout the document. The Program Summary presented a consolidation of the space list. Component Descriptions provide all the detailed space information about each group. Components were the program, departmental, or administrative entities that were defined as appropriate building blocks for design. Each Component was described in terms of a Functional Role, Enrollment, Staffing, Neighbours, Internal Relationships, and Space Data. Appendix B provides an example of one Component with all its associated information.

September Weeks 3-4
7.0 DESIGN LIAISON
7.1 Block Planning Liaison
7.2 Schematic Concept Liaison

During the third and fourth week of September, Process Four Design began the liaison with the architect. For this, the programmers used an activity they called modelling. Modelling occurred with the User Committees of the new Richmond Campus in September. Color coded pieces of paper were cut out to scale for all the spaces in the new Campus. A building grid or outline was provided. In the modelling sessions, the User

Committees moved the pieces of paper about in the building outline. The users laid out the building using the paper pieces the way they wanted the Campus to work for them. The architects stood by and watched and listened. Modelling is Process Four Design's method of linking programming and architecture. The architects began to understand how the actual users of the building wanted the building to work, instead of making assumptions. The architect found this process very useful as it provided them with an initial estimate at a floor layout. It also provided them with lots of conceptual information which they otherwise would not have known until much later in their involvement. Modelling was to be the last step in Process Four Design's involvement in the project until problems arose in December of 1989.

In December, budget problems arose. The cost consultant advised that construction costs in Vancouver had risen significantly in a short period of time. Kwantlen College could no longer build the space list outlined in the program, as it would be over budget. Space needed to be cut to stay within the fixed budget.

On January 4th, 1990, an additional contract was negotiated between Kwantlen College and Process Four Design to cut 23% of the space outlined in their original program. The consulting firm was to review other college models to see if it was possible to cut space by 23% and still have a workable college. The firm was to report back to the Directing Committee by January 12, 1990 with the information.

Process Four Design embarked on this additional scope of work by touring other College Campuses in the Lower Mainland to find out where to possibly cut space in the Richmond Campus. Information was collected and documented about spaces in other Campuses. Process Four Design shortly determined it was possible for the Richmond Campus to work as a Campus with 23% less space than allocated in the Final Program document. Process Four Design made a list of recommendations about where the Directing Committee could cut 23% of space from the Richmond Campus space list. The Directing Committee examined the recommendations, and added some of their own recommendations. Decisions were then made by the Directing Committee to get space down to the targeted amount. This did not affect the Richmond Campus Educational Plan as outlined previously in this chapter.

Process Four Design proceeded to revise the space list, and by the end of January 1990 had a final space list, along with prioritized spaces to go back in, budget permitting. A working paper was distributed in January to the User Committees. This working paper attempted to inform the users of the situation to date, and showed them where cuts had been made. The purpose of this paper was to ensure that each educational program and

department could work with the newly revised space, and to ensure decisions were appropriate. For the most part, there were no further revisions.

Now that the revised space list was fixed, Process Four Design incorporated all the changes into the Final Program document. The revised Final Program was distributed February, 1990.

CHAPTER 4

DETAILED PROCESS OF THE PROGRAMMING OF KWANTLEN COLLEGE RICHMOND CAMPUS

In Chapter 3, the chronological events in the programming of the Richmond Campus were described. Now that this context has been laid, this chapter presents detailed explanations of processes used by the programmers in the Richmond Campus project. This Chapter consists of three sections, each addressing a more detailed level of information than the one before. First, the general process used in the project will be explained, with the help of a flow diagram. Secondly, specific details of the process used will be outlined, along with the description of important tools. A more detailed flow diagram will help to clarify the process. Lastly, we will review comments made by the Kwantlen staff and faculty involved in the process.

4.1 General Structure of the Process

First, a basic outline of the process used in the programming of the Richmond Campus will be reviewed. The purpose of this section is to review the general structure the programmers set for the flow of information. The programmers set up a cycle for information flows, each of the parts are repeated a number of times. These three parts included:

Input;
Control & Decision Making; and
Products.

The Process Flow Diagram (See Figure 4 : Programming Process Flow Diagram) following illustrates the basic cycling of the three parts. Information generation and decision-making are a series of these three parts, that is, “Input / Control & Decision Making / Products ” steps. This diagram also depicts the flow of information between the decision making body (Directing Committee of Kwantlen College), the User Committees, and the programmer. These steps are explained in more detail following the diagram.

PROGRAMMING PROCESS FLOW DIAGRAM

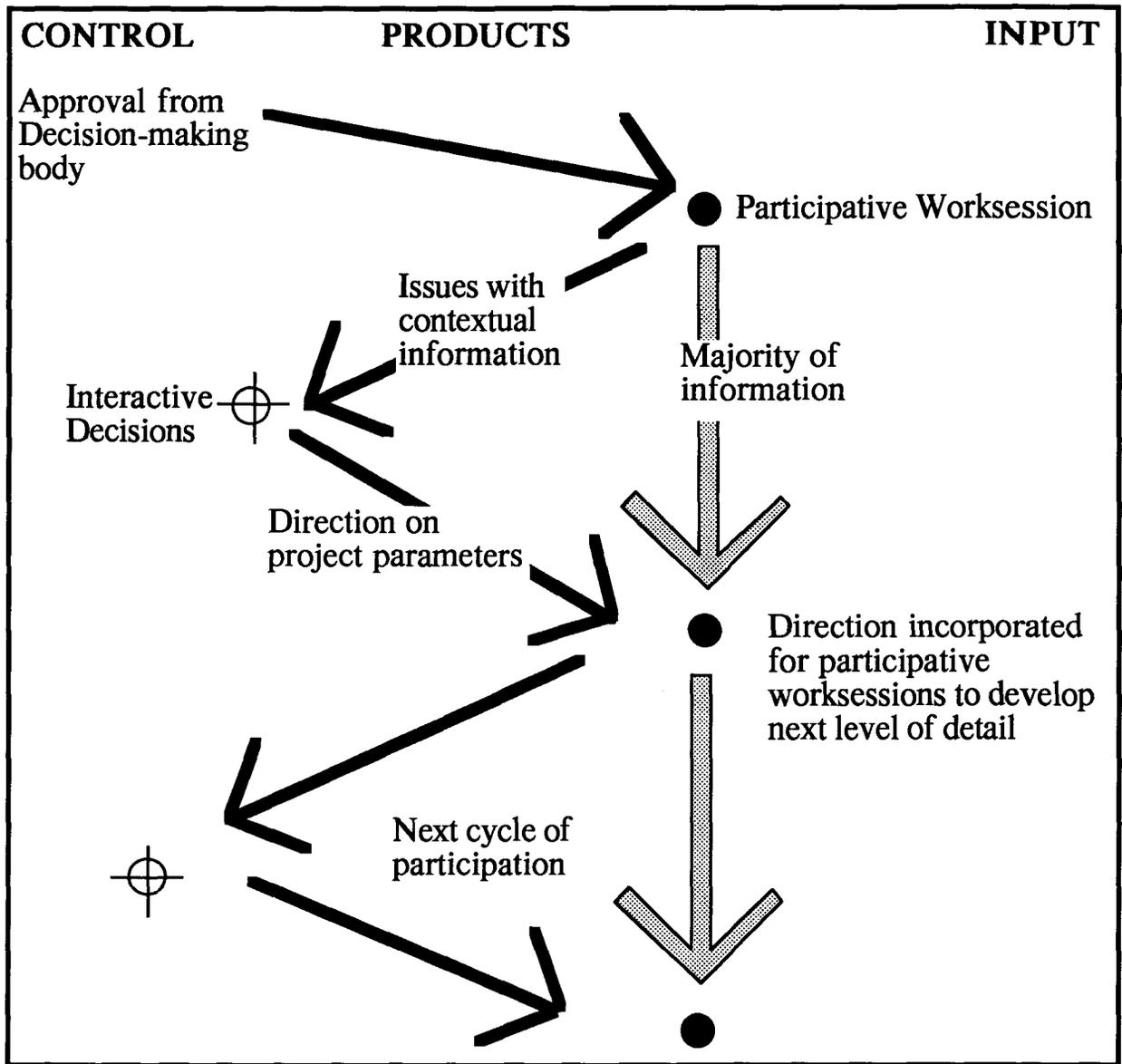


Figure 4: Programming Process Flow Diagram, from Process Four Design & Aitken Wreglesworth Architects, *Interactive Design: A Better Way to Involve Users* (Vancouver, BC: Process Four Design & Aitken Wreglesworth Architects July, 1991) 7.

Input steps, indicated on the far right side of the flow diagram, show the main body of information generation. Input represents the main source of programmatic information. Most information was developed with and by participating Kwantlen staff members. The mandate for participating staff, User Committees, was that they should have input into the programming process. The mandate and role of User Committees was strictly defined by

the Directing Committee. As outlined in Chapter 3, Five User Committees were established by the Directing Committee, including Applied Arts and Career Preparation, Design Centre, Academic and Applied Studies, Educational Support, and Administrative Services. Each of the Committees represented a major organizational teaching or administrative group within the Campus. Membership of these groups was also established by the Directing Committee.

Participative worksessions, held by the programmer with each User Committee, were the locus for Input. The purpose of meeting in a worksession format was to get people thinking together and working together on a single goal. The consultant recorded all information in worksessions and interviews. The material generated by users was always sent back to them in a working paper, with a structure and time frame for revisions and comment. The programmer used the working paper as documentation of the last meeting. Participants used the working paper as vehicle for further comment, writing ideas and getting additional information from other colleagues. Revisions to working papers were collected, and material from revised working papers formed the basis for document sections. Worksessions and interviews were used throughout the process to gather or generate a different type of information or a different level of detail.

Control actions, on the left hand side of the flow diagram, were the responsibility of the decision-making body established for the project. The Directing Committee of Kwantlen College was the decision-making body already in place, and it acted as the Control device. The Directing Committee maintained control of the process within a scope or mandate set out at the start of the project, which was largely financial. Chapter 3 provides more detail on the Directing Committee's role.

Information generated by User Committees generally fell within established parameters such as cost and space constraints, and was carried forward. However, sometimes issues or problems arose and the Directing Committee needed to make a decision in these cases. Issues were always presented to the Committee by the programmer in their information context, in order that informed decisions could be made. In other words, the programmer presented the Directing Committee with the issue and all information pertaining to it. It would not be effective for the decision-making bodies to concern themselves with the minutiae of detailed planning information, but they did have copies of all documentation in their files.

In some cases, issues went back and forth between the Directing Committee and the User Committees several times. This happened when the users felt particularly strongly about an issue. For example, during the stage of allocating office space, the User Committees desired single offices, that is, one person per office. The users presented their

case for this desired set of conditions to the programmer. However, the Directing Committee decided that faculty offices should be shared for financial considerations. The User Committees brought it up again because they felt so strongly, they wanted single offices badly. However, they were once again vetoed by the Directing Committee.

Products were the organized information resulting from any part of the process. Products acted as the link between the Directing Committee (the Control) and the User Committees (the Input). The Products were the programmer's responsibility, whether they were presented orally or in writing. Documents or products communicated information to the User Committees and the Directing Committee. Products flowed from the User Committees to the Directing Committee, as well from the Directing Committee to the User Committees. Movement from the User Committees to the Directing Committee indicated that the Directing Committee needed to make some kind of decision. Information already developed by the User Committees at the Input level was being transferred to the Directing Committee. This decision was sometimes a simple okay and go ahead, or it sometimes was a revision. Products also flowed from the Directing Committee to the User Committees. This occurred after the Directing Committee had made a decision, and this decision needed to be communicated back to the User Committees. The products in this case served as the basis for further direction to participating staff. User Committees were always working within real life constraints and parameters set down by the Directing Committee.

4.2 Detailed Process

Having explained in Section 1 the information flow set up for the programming process, the purpose of Section 2 is to provide detailed explanations of the programming process used for the Kwantlen College Richmond Campus.

The detailed process used in the programming of the Richmond Campus is illustrated in a detailed flow diagram, see Figure 5: Detailed Process Flow Diagram. This diagram is set up similarly to the more general Process Flow Diagram (Figure 4), using Control, Products and Input cycles. However, Figure 5 presents the process in greater detail.

DETAILED PROCESS FLOW DIAGRAM

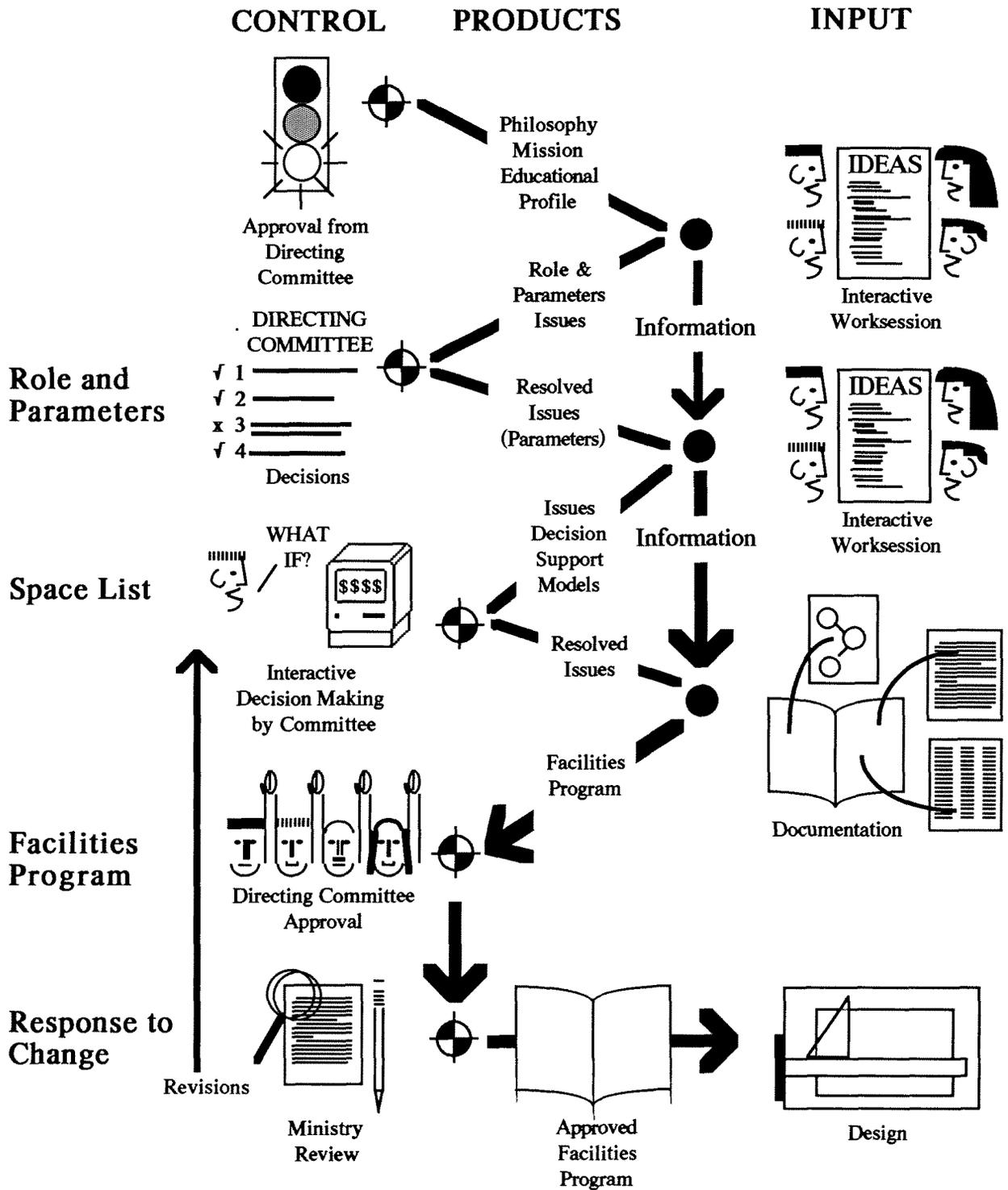


Figure 5: Detailed Process Flow Diagram, from Process Four Design & Aitken Wreglesworth Architects, Interactive Design : A Better Way to Involve Users (Vancouver, BC: Process Four Design & Aitken Wreglesworth Architects July, 1991) 9.

The programming portion of the process consisted of four major parts:

- I Role and Parameters (Macro decisions)
- II Space List (Micro decisions)
- III Facilities Program (Detailed information)
- IV Response to Change (Adjustments)

These four parts are seen on Figure 5 diagram on the left hand side. Each is also in fact the name of the final product of that part of the process. The diagram and the following text will help us to understand stages or cycles in arriving at these final products. Each part will be explained in terms of the Control, Products, Input cycles. The interactive process applied to the Kwantlen project problems ensured that staff provided the input and information, while the decision makers retain their decision making control.

I : Role and Parameters

The Control for this stage of the project was approval from the Directing Committee of Kwantlen for programming go ahead. This meant that the programmer was hired, protocols and structures were in place, and programming was initiated. The products at this point consisted of existing Kwantlen documentation, including education plans, educational profiles of the new Campus, mission and philosophy statements of the College, and descriptions of the various User Committees and their mandates. The programmer reviewed the existing Kwantlen products for context.

Kwantlen's mission and philosophy statement was and is an expression of the belief systems of their organization, or more simply, the reason for doing the things it is doing. The philosophy and mission acted as a general guide for facility planning and design decisions. The educational profile was the organization's first attempt to outline the programs and allocated FTEs to be taught at the new facility. It consisted of the educational programs to be taught at the new Campus along with the number of students allocated to each program. With appropriate resources, including dynamic numerical Decision Support Models (explained in Tools section), this initial educational profile was not treated as a concrete statement, but was viewed as a moving target. Kwantlen was not held to an obsolete educational profile as new information was learned, including the impacts of the budget.

Organizations should expect to juggle programs and FTEs during this programming process. In the case of Kwantlen College Richmond Campus, a complete educational program, a Waste Management program, was added after the Directing Committee had approved a final draft program.

Input sources and processes at this stage consisted of participative worksessions with User Committees. A key attitude on the part of the consultant was that the user had expert

information on functions and spaces, but was not accustomed to articulating this knowledge. The consultant's role was to help the user to discover and articulate the information, input, and opinions which they have. Development and generation of information occurred through worksessions with groups of people, called User Committees. These meetings were a very fast way of generating a lot of information in a short time, and included brainstorming sessions. Their objectives included achieving group consensus on issues that arise, as well as generation of information. The consultant's strength was experience and innovative facilitation skills that helped keep meetings in control, and make them exciting for users.

Users acted as representatives for their areas. At this stage, Process Four Design used User Committees. A User Committee contained a mixture of people from within a department, to achieve a cross fertilization of ideas, and to get input on departmental issues. The strength of meeting departmentally was that issues arising at this level were resolved quickly, and thus did not have to be referred to the decision-making body. In most other planning exercises, this key step is forgotten and staff in one area do not know what the others are doing, which leads to defensive, uncooperative attitudes.

The primary tool used by the consultant in information gathering and development was the worksessions with User Committees. In worksessions, the programmer used brainstorming techniques to focus on a specific subject and get all information about that subject before making any decisions. Worksessions were also held with individuals, where these individuals had relevant knowledge.

Flipcharts were used by the consultant during user worksessions. Rather than taking notes by hand on a note pad, the consultant kept a record of the information developed by jotting it on the flipchart. The flipchart was kept visible to all participants to ensure that what was written down by the consultant was a correct documentation of what was said.

Working papers were used to document information developed in the user worksessions. Flipchart information is quickly transcribed into a computer file. Material developed at the worksessions is then sorted and processed by the consultant, using a computer database if necessary. The sorted and processed information, along with any questions the consultant may have about the information, is organized into a working paper. This working paper had an introduction, an explanation of the process, and an explanation of what will happen next, together with the workshop transcript. The working papers enabled participants to ensure that the programmers understood and heard exactly what the users told them, and gave them a second chance to cover errors and omissions. Users also distributed working papers information among their colleagues to get broader

participation. Users and their colleagues responded openly and directly on the papers by writing their comments. Papers were sent back to the consultant when completed.

FAX and phone were also tools that were used extensively in the programming process. The FAX machine was used to instantly send and receive notes about information. The phone was used in the event that meetings could not be arranged and information had to be transferred quickly. The phone was also used where the programmer needed clarification on an issue.

There were two products from the Role and Parameters part of the process, including Role and Parameters Statements, and Role and Parameters Issues. Role and Parameters Statements was one of the first documented products to come from the interactive worksessions with the User Committees. These statements were documented in Working Paper #1 as outlined in Chapter 3. Role and Parameters provided a broad description of services and activities that each entity (educational program or administrative entity) would fulfil. These statements described the limits and overall framework within which planning would occur, setting out what each entity would and would not do. Role and Parameters were described in current terms, as well as in terms of changes anticipated in the future. In the case of Kwantlen College Richmond Campus, the users described each educational program or administrative entity in terms of its role within the College and Campus. Limits for role, staffing, and student numbers (student full time equivalent numbers) allocations were established.

Role and Parameters Issues arose from the participative worksessions at the department level. For example, Kwantlen users had responses to the initial educational profile regarding student full time equivalent numbers allocated to educational programs. Other issues came up, such as a certain educational program was seen as not viable, and the users felt it should be cut. These issues fed into a Control cycle. Issues were forwarded to the Directing Committee in the form of Working Paper #2 for decisions. Few issues, in fact, were carried forward to the Directing Committee, as most were resolved at the User Committee level. Only issues that needed to be addressed by the decision makers were forwarded to them ; it was not necessary for them to address all information generated. The programmer extracted the necessary and relevant information from the worksessions to compile a working paper for the Directing Committee. This working paper addressed the issues and provided the context in which the Directing Committee could make well informed decisions. This approach ensured that the time of the decision makers was not wasted, and that they were responsible for the final decisions in this process.

Resolved issues, products from the Directing Committee's decisions, were fed back to the Users at the next interactive worksession.

II : Space List

The Space List was the final product of this part. Input consisted of participative worksessions with Department Groups. Worksessions were held with each representative user and their department for detailed information on space requirements, organizational concepts, furnishings, technical requirements, and courses or programs. A framework and structure was already in place for this information to be developed, as the limits of each program were established previously in Role and Parameters. Users worked to describe the needs for teaching the education plan as defined.

The tools used in these worksessions were Decision Support Models. Process Four Design set up microcomputer models that simulated the complex situation of the new Campus. Instead of needing to work with lengthy manual calculations that make changes difficult to incorporate, these models allowed the client and consultant to manipulate all the variables interactively. Multiple iterations did not significantly affect time and results were derived instantly. Decision Support Models developed by the consultants for Kwantlen College Richmond Campus include a Space List, a Space-Cost Model, and a Utilization Model. These models created a legacy of the programming process, and were intended to be used in operational planning long after the facility is occupied to resolve new issues and problems which arose, especially in regards to space allocations.

The Space List listed all the space to be included in the new Campus, along with the number and sizes of the spaces. The Space List was arrived at by using Ministry of Advanced Education, Training and Technology space standards, Surrey Campus space standards, other College space standards, as well as user input on spaces needed to teach the educational profile and an inventory of the existing facility. At the direction of the Directing Committee, spaces at the Richmond Campus were based on Surrey Campus spaces, that is, Richmond Campus spaces were to be similar in size to Surrey Campus spaces. Space size was also based on Ministry space standards (See Table II : Ministry of Advanced Education, Training and Technology Space Standards). These standards provided Net Assignable Square Footage per Student for a space. Space size was also based on the existing Campus, as well as user expressed needs. The programmer worked with the user to arrive at the required size of a space. The Space List for Kwantlen College Richmond Campus is part of a larger Space-Cost Model.

The Space-Cost Model (See Figure 6: Excerpt from Space-Cost Model for the Richmond Campus) was used as a tool to help the client make decisions about space based on total project cost considerations. The client quickly saw how total budget changed with space changes. The Space-Cost Model is simply the Space List with the costs of space added. Cost consultants (Barnett Treharne Yates BC Limited) supplied the programming

consultants with unit costs for different types of spaces, and the programmer linked unit costs to units of space to arrive at a total cost for the project. As choices were made about space, changes in cost allocations were determined immediately. The client saw the financial impact of “what if” questions about spaces. This was critical to Kwantlen College as they were working within a fixed budget set by the Ministry.

**MINISTRY OF ADVANCED EDUCATION, TRAINING AND
TECHNOLOGY SPACE STANDARDS**

**Recommended Student Stations Unit Area Guidelines for Instructional Rooms in BC
Post-Secondary Institutions**

Room Type	Recommended Unit Area
Lecture Theatre	12-15 square feet
Seminar Room	15-18 square feet
Classroom	25 square feet
Demonstration Room	43 square feet
Micro Computer Area	40 square feet
Accounting Lab	49 square feet
Biology Lab	43 square feet
BTSD Lab	50 square feet
Chemistry Lab	53 square feet
Dental Assisting Lab	74 square feet
Drafting Lab	92 square feet
Language Lab	37 square feet
Nursing Lab	68 square feet
Physics Lab	48 square feet
Typing Lab	39 square feet
Autobody Shop	220 square feet
Automechanics Shop	220 square feet
Carpentry Shop	150 square feet
Intro Electricity Shop	121 square feet
Heavy Duty Mechanics Shop	353 square feet
Welding Shop	134 square feet

Table III: Ministry of Advanced Education, Training and Technology Space Standards, from Kwantlen College Richmond Campus Facilities Program (Vancouver, BC: Process Four Design, February, 1990).

	Quantity	Capacity	Unit Alloc'n	Alloc'n Sq Ft	Net Ass'ble Square Footage Cost
Lecture Theatre	1.0	125.0	13.0	1625.0	\$150,967
Support Space for A/V	1.0			80.0	\$7,432
ACADEMIC AND APPLIED STUDIES					
Humanities and Social Sciences Dept					
Classroom, 40 Seats	12.0	40.0	25.0	12000.0	\$925,314
Classroom, 25 Seats	4.0	25.0	25.0	2500.0	\$192,774
Lab, Seminar & Converts	1.0	10.0	18.0	180.0	\$17,559
Lab, Language Audio	1.0	20.0	40.0	800.0	\$78,039
Lab, Psychology	1.0	12.0	40.0	480.0	\$46,823
Project Room	2.0	1.0	120.0	240.0	\$23,412
Observation area	1.0	1.0	80.0	80.0	\$7,804
Prep Room / Tech Station	1.0	1.0	120.0	120.0	\$11,706
Lab, Anthropology	1.0	15.0	30.0	450.0	\$43,897
Lab, Computer & English	2.0	20.0	30.0	1200.0	\$128,206
Lab, ELT	4.0	20.0	18.0	1440.0	\$140,469
Office, Shared Faculty	22.0	2.0	80.0	3520.0	\$310,668
Office, Technician	4.0	2.0	60.0	480.0	\$42,364
Office, Educational Director	1.0				
Workstation, Clerical	1.0				
Science, Applied Science, and Technology Dept					
Classroom, 40 Seats	6.0	40.0	25.0	6000.0	\$462,657
Classroom, 25 Seats	1.0	25.0	25.0	625.0	\$48,193
Lab, Biology 1st year	1.0	40.0	50.0	2000.0	\$269,419
Tape Dispensing / Tech Stn	1.0	1.0	50.0	50.0	\$4,877
Storage	1.0	1.0	300.0	300.0	\$29,264
Ecology & Demo Area	1.0	1.0	600.0	600.0	\$80,826

FIGURE 9: EXCERPT FROM SPACE-COST MODEL FOR THE RICHMOND CAMPUS

	Quantity	Capacity	Unit Alloc'n	Alloc'n Sq Ft	Net Ass'ble Square Footage Cost
Lab, Biology 2nd year	1.0	22.0	50.0	1100.0	\$148,180
Prep Room / Tech Station	1.0	1.0	600.0	600.0	\$58,529
Storage	1.0	1.0	300.0	300.0	\$29,264
Lab, Microbiology	1.0	12.0	45.0	540.0	\$80,268
Prep Room	1.0	1.0	600.0	600.0	\$58,529
Lab, Chemistry 1st year	1.0	40.0	50.0	2000.0	\$297,290
Prep Room / Tech Station	1.0	1.0	600.0	600.0	\$58,529
Storage, Glassware	1.0	1.0	300.0	300.0	\$29,264
Storage, Chemicals	1.0	1.0	300.0	300.0	\$29,264
Instrument Room	1.0	1.0	600.0	600.0	\$58,529
Bunker	1.0	1.0	300.0	300.0	\$29,264
Balance Room	1.0	1.0	300.0	300.0	\$29,264
Lab, Chemistry 2nd year	1.0	28.0	50.0	1400.0	\$208,103
will share 1st year support					
Lab, Physics 1st year	1.0	40.0	40.0	1600.0	\$156,077
Prep Room / Tech Station	1.0	1.0	600.0	600.0	\$58,529
Storage	1.0	1.0	300.0	300.0	\$29,264
Lab, Physics 2nd year / App Sc	1.0	28.0	35.0	980.0	\$95,597
Project / Optics Room	1.0	1.0	300.0	300.0	\$29,264
Storage	1.0	1.0	300.0	300.0	\$29,264
Lab, Computer	1.0	40.0	35.0	1400.0	\$149,574
Classroom / Lab, Geography	1.0	40.0	45.0	1800.0	\$242,477
Classroom / Lab, Geology	1.0	40.0	45.0	1800.0	\$242,477
Prep Room	1.0	1.0	1.0	600.0	\$80,826
Office, Shared Faculty	10.0	2.0	80.0	1600.0	\$141,213
Office, Lab Supervisor	5.0	2.0	80.0	800.0	\$70,606
Office, Educational Director	1.0				
Workstation, Clerical	1.0				

FIGURE 9: EXCERPT FROM SPACE-COST MODEL FOR THE RICHMOND CAMPUS

Student Association	Quantity	Capacity	Unit Alloc'n	Alloc'n Sq Ft	Net Ass'ble Square Footage Cost
Office	1.0	8.0	50.0	400.0	\$35,303
Locker, Student	120.0	1.0	1.0	120.0	\$10,591
Lounge, Student	1.0	40.0	20.0	800.0	\$70,606
Job Placement					
Office, Placement Officer	1.0	1.0	120.0	120.0	\$10,591
TOTAL SQUARE FOOTAGE				172506.9	
BUILDING GROSS SQ FOOTAGE	1.43			246684.9	
LABS	54.0				
CLASSROOMS	41.0				
CONSTRUCTION COST, NASF					\$15,970,377
GROSS FACTOR	1.43				\$6,867,262
TOTAL BUILDING CONSTRUCTION COST					\$22,837,640
PARKING					
SURFACE (600 CARS)		600.0			\$600,000
UNDER BUILDING (300 CARS)		300.0			\$2,550,000
SITE DEVELOPMENT					
GENERAL					\$580,000
ROADWAY					\$165,000
SITE CLEARANCE & PREPARATION					\$318,000
OFFSITE SERVICES & CONNECTION CHARGES					\$80,000
TOTAL CONSTRUCTION COST					\$27,130,640
DEVELOPMENT COST CHARGES					
BUILDING PERMIT					\$720,000
FURNISHINGS & EQUIPMENT					\$163,000
PROFESSIONAL FEES					\$2,771,000
SUPPLEMENTARY COSTS					\$2,925,000
ESCALATION ALLOWANCE					\$554,000
					\$3,483,000

Figure 6: Excerpt from Space-Cost Model for the Richmond Campus, from Kwantlen College Richmond Campus Facilities Program (Vancouver, BC: Process Four Design, February, 1990).

The Utilization Model set up for the Richmond Campus was an analytical tool to help users and decision makers make decisions about spaces and see the implications of those decisions based on the level of utilization of a space. (See Figure 7: Excerpt from Utilization Model for the Richmond Campus). This was an objective way to show space use, and helped decision makers to make informed decisions about both spaces and programs interactively. The Utilization Model was a computer based numerical model that showed how the new Campus' programs would function with respect to space use. It mathematically linked the various functions of the new Campus, by describing all programs and courses that would be taught. Variables used in the model included day hours, evening hours, section size, room sizes, number of courses, weekly contact hours by course, and lecture or lab format. For each course, the faculty outlined the input figures. The input figures included the names of all courses, the number of sections taught per course, the weekly contact hours per course, the number of students in each section, and the name of the teaching space. Through the use of complex spreadsheet formulas, the number of spaces required to teach those courses is calculated, along with the percent utilization of each space. This model was used as a check against the number of classrooms and labs being planned in the program. The model helped the Directing Committee to make informed decision about spaces, for example combine spaces with low utilization.

HOURS / WEEK	35 DAY	12 NIGHT	# OF	HRS OF	HRS OF	SECT	# CRSE	RM	#	ROOM	
Maximum Utilization	0.80		SECT/	CL USE	CL USE	SIZE	SEATS	SIZE	LABS	UTIL'N	
Available Day Hours	28	WCH/	ASSOC	HRS OF	HRS OF	SECT <td># CRSE</td> <td>RM</td> <td>#</td> <td>ROOM</td>	# CRSE	RM	#	ROOM	
Available Evening Hours	9.6	CRS	HOURS	LAB USE	CL USE	SIZE	SEATS	SIZE	LABS	UTIL'N	
		SECT/	HOURS	LAB USE	CL USE	SIZE	SEATS	SIZE	LABS	UTIL'N	
ACADEMIC AND APPLIED STUDIES											
BUSINESS AND CAREER PREPARATION											
CLASSROOMS, 40 SEAT	1				148.0			40	0.0	6.0	70%
Night	1				66.0			40	0.0	7.0	79%
CLASSROOMS, 26 SEAT	1				78.0			26	0.0	3.0	74%
Night	1				28.5			26	0.0	3.0	79%
(pooled)											
ACCOUNTING											
CLASSROOM											
First Year courses	1	3	4	0	0.0	36.0	37	40			
Second Year courses	1	4	4	0	0.0	32.0	25	25			
First Year courses (night)	1	2	3	0	0.0	18.0	37	40			
Second Year courses (night)	1	1	3	0	0.0	12.0	25	25			
					27.0	Total Sections					
BUSINESS											
CLASSROOM											
First Year courses	1	3	4	0	1.0	12.0	37	111			
Second Year courses	1	2	3	0	3.0	18.0	25	150			
First Year courses (night)	1	1	3	0	4.0	12.0	37	148			
Second Year courses (night)	1	1	3	0	2.0	6.0	25	50			
					15.0	Total Sections					
COMPUTER BASED SYSTEMS											
CLASSROOM											
First Year courses	1	2	2	0	3.0	12.0	37	222			
Second Year courses	1	1	2	0	2.0	4.0	25	50			
First Year courses (night)	1	1	1.5	0	5.0	7.5	37	185			
Second Year courses (night)	1	1	1.5	0	2.0	3.0	25	50			
					15.0	Total Sections					
COMPUTER LAB											
First Year courses	0	2	2	0	4.0	16.0	37	296	40	1.0	57%
Second Year courses	0	2	2	0	1.0	4.0	25	50			46%
											11%

FIGURE 10: EXCERPT FROM UTILIZATION MODEL FOR THE RICHMOND CAMPUS

Figure 7: Excerpt from Utilization Model for the Richmond Campus, from Kwantlen College Richmond Campus Facilities Program (Vancouver, BC: Process Four Design, February, 1990).

Control was achieved by the Directing Committee using the Decision Support Models. Decisions were made expediently because the programmer extracted issues and relevant information. Decision makers were aware of the context, and made fully informed decisions. They also tested “what if” types of questions, as the models quickly regenerated and showed the cost impact of these “what if” questions. An interactive session, using a portable computer, was used to explore the impact of potential decisions. The Utilization Model showed the impact of combining spaces, or providing dedicated spaces. It pointed out spaces with low utilization, which might not have been acceptable to the Ministry. The Space-Cost Model helped decision makers to see the financial impact of adding or cutting a specific space within seconds. This was critical to Kwantlen College, since they were working within a given fixed budget.

Products included a Space List that fell within the financial parameters, as well as other programmatic information including organizational concepts, furnishings, and technical requirements. Issues resolved by the Directing Committee were communicated to users prior to the next worksession. It was important that users know the “why and what for” of issues resolution. This ensured that users were always working within defined parameters and being asked for input, not decisions. The programmer was responsible for communicating the issues to the Directing Committee, as well as communicating the decisions back to the User Committees.

III : Facilities Program

At this stage the collected information and decisions were organized and documented by the programmers. The assembly of the final facilities program occurred quickly as all information developed during the process was already contained on microcomputer files. Because of this, the programmer also had the ability to revise material quickly. The final product was a Facilities Program. Process Four Design bound the documents in a three ring binder to permit insertion of additional pages as changes occurred, because the program was not seen as a static document.

IV: Response to Change

At this point, the Ministry of Advanced Education, Training and Technology reviewed the Facilities Program material supplied to them regarding space utilization, space figures, and costs. After review, the Ministry tried to renegotiate the scope of the project with Kwantlen College. They had some questions about utilization of spaces, the number of spaces, and the size of spaces. Basically, the Ministry tried to cut the scope of the project so it would cost less money. During these negotiations with the Ministry, the Directing Committee was able to respond to their inquiries from a position of knowledge about the

facilities program. The Directing Committee could also immediately determine the impact of budget changes and facilities changes on programs, enrollments and staffing. Process Four Design could quickly understand and respond to changing directions set by the Ministry, as all information was set up on spreadsheets and could be quickly revised.

Revisions to the program material, initiated by the Ministry or other influences, occurred quickly because the complete Kwantlen College organization was familiar with the process and the information. Each user had participated in developing the information for their area and knew it in detail. All information is set up on spreadsheets, which enabled quick recalculation. In this case of the Kwantlen College Richmond Campus programming process, the final draft of facilities program had been completed when it was determined that construction costs had increased significantly and the project was suddenly over budget. The Directing Committee immediately met and gave direction to the programmers to cut spaces to get back within the project budget. In a timeframe of two weeks, with the interactive participation of the users, 23% of space was cut.

Finally, after Ministry approval, the final product was a Facilities Program. The Facilities Program was passed on to the architects, and the project proceeded into design.

4.3 Kwantlen Comments

The purpose of this section is to review how the faculty and staff of Kwantlen College felt about their involvement in the programming process. We cannot yet determine how they feel about the final product, the Campus building, as it is still under construction, but a main idea behind this process is that the participants feel a sense of involvement and importance. Also the participants give the best indication of how the process actually proceeded, as they were involved on a day to day basis.

Twelve Kwantlen staff members who were participants in the programming process of the new Richmond Campus were interviewed. They were asked a number of questions in order to flesh out the history of the project, as well as to find out how they felt about being involved in the programming process. As recounted by the participants, the pros and cons of the programming process of the Richmond Campus are outlined.

It seems that the most highly cited or most important facet of the process for the participants was the level and extent of faculty and staff involvement. All interviewed agreed that there was a tremendous number of people involved. The involvement was excellent, and there was a great deal of interest and excitement generated by the process. Participants enjoyed the fact that they had “an opportunity to verbalize their concerns and ideas” (Peter-Cherneff 1991). Participants were sure their input was used, and what they said had an impact. Everyone got a fair hearing.

The process "... challenged users to define their needs rationally, objectively, and clearly" (Baker 1991). Users gained perspective of others' needs, and they were brought closer to understanding each others needs. The process forced users to compromise and work out their own conflicts. Gary Baker, the Director of Continuing Education at Kwantlen College stated that the participants "... negotiated their own compromises, which resulted in partial ownership of process and total ownership of product" (Baker 1991).

In general the participants agreed that the programming process moved along well. It seemed clear and organized. The process had clarity of structure. It was well organized and thought through.

Participants found that the process had hidden benefits. The Kwantlen staff at the Richmond Campus are still working as a team today. The team benefits are in place far after the programming is over. The sense of achievement for all Kwantlen staff and faculty remains. And most importantly from the College's perspective, the staff and faculty at the Richmond Campus feel ownership of the product.

Despite the benefits associated with the process, there were also some shortcomings. It seems the main fault of the process was the time frame. Participants were often asked to provide information on short notice. There was little time provided to respond to working papers. The process was short and intense, and sometimes participants did not have a chance to consult with colleagues.

There was also the problem of raised expectations. A hazard of any consultation is that "... people think they are being asked for more than input." (Francis 1991). As soon as organizations involve staff in any process, the staff believe they have a democratic vote. Participants believe that they will get what they ask for, even though their mandate as information sources only was clearly outlined.

Finally, because of the newness of the process to Kwantlen College, there were some doubts whether the process would be successful or not. There was a certain ambiguity that was present in the process.

CHAPTER 5

IMPLICATIONS FOR COMMUNITY PLANNING

The purpose of this chapter is to examine the implications of programming for community planning. Chapter 2 presented a detailed explanation of programming within a conceptual framework. Chapters 3 & 4 presented the case study of the programming of the Richmond Campus. This case study was presented as an example of “leading edge” programming. That is, it was more participative and more interactive than traditional programming as well as being supported by staged continuous decision making and good analytic tools. Although these four ideas are reflected in some aspects of community planning thought and practice, the case study brings them into very sharp focus as important components of a successful planning process in the programming profession. The following analysis shows that these four ideas in “leading edge” programming could be applied beneficially to community planning.

In order to discuss these four ideas that community planning can gain from programming, it is important to discuss the similarities and differences between the two professions. We need some evidence that there are similarities between community planning and programming that would permit transfer of training to occur. It is not possible to compare programming, as described in this study, to all the different theories and processes of community planning as this is clearly beyond the scope of this thesis. The purpose is therefore limited to the implications programming provides for community planning as described in selected literature and as typically practised today in Canada. The main sources for definitions of community planning include Salomaa (1981), Hodge (1990), Ministry of Municipal Affairs (1985), Faludi (1973), Gauld (1986), and Chapin & Kaiser (1979).

The theory of community planning as practised today in Canada that is accepted for this analysis is based on the rational comprehensive model as defined by Hodge (1990). Hodge states “the rational comprehensive model stands at the foundation of our modern plan-making processes for Canadian communities” (Hodge 1990).

I will describe the rational comprehensive theory of community planning, and compare it to programming. However, one of the problems with comparing programming to the theory of community planning is that practice frequently differs from the theory described in the literature. I will therefore also discuss my understanding of the practice where I believe it departs from the theory.

This chapter is organized into two sections. In the first section, a table is used as a general method of comparison of the important characteristics of programming and community planning theory. Detailed discussion of the characteristics presented in the table is provided, as well as a discussion of community planning in practice. In the second section, implications of programming for community planning are presented.

Summary of Programming and Community Planning Characteristics

Category	Programming	Community Planning
Purpose	Outline and specify parameters and criteria to guide facility design	Outline and specify parameters and criteria to guide community development
Technique or Method	Identify problem & state goals, Interactive information development, Analysis (including solution synthesis), Staged continuous decision making (issue resolution), Communication and documentation	Identify problem and state goals, Survey community, Design alternative plans, Evaluate, Adopt best plan
Participation	Interactive Controllable	Reactionary Uncontrollable
Decision Makers	Appointed Project Committee	Elected Politicians
Scope	Micro	Micro & Macro
Implementation	Through design Subject to interpretation	Through development and regulations Subject to interpretation
Client	One organization	Municipal council Public Vague
Role of Professional	Sets up processes for information development and decision making	Participate in setting up processes for information development and decision making
Product	Program Document	Plan Document
Time Frame	Relatively short	Continuous Into future

Table IV: Summary of Programming and Community Planning Characteristics

5.1 Detailed Discussion of Programming and Community Planning Characteristics

Each of the characteristics will be discussed in turn as presented in the above figure. The community planning characteristics are as defined in planning theory, but the discussions will also cover some aspects of community planning in practice, where it appears appropriate to do so.

Purpose

The purpose of programming is to outline and specify the parameters and criteria to guide a facility's design. The programmer serves the client organization by gathering and outlining this type of information for the architect. Programming is defined as the collection and analysis of information which the designer needs to design the facility. It involves solving problems about competing needs and interests. The program must meet the goals outlined by the clients and the participants in the programming process.

Similar to programming, the purpose of community planning, as defined in theory, is to outline and specify parameters and criteria to guide a community's development. A secondary purpose "...is to serve people...", (Ministry of Municipal Affairs 1985) or in other words, planning aims to promote the public interest when coping with various urban problems.

Community planning is also defined as the collection and processing of information for urban development. "...Planning derives initially from the need to solve problems or the desire to achieve ambitions or aspirations..." (Hodge 1990). In theory, the purposes of community planning and programming are similar. Both try to objectify criteria and parameters to guide physical development and both set criteria for implementation.

In community planning practice, however, there are often competing and conflicting goals. Programming involves working and guiding one organization with clearly defined goals, while planning involves guiding a large community with no clear, common goals.

Technique or Method

The programming process is a problem solving activity, that involves a large number of variables. The stages or steps of the process are seen in Figure 8, and include identifying the problem and stating goals, interactive information development, analysis, solution synthesis, staged continuous decision making, issue resolution, and communication and documentation. The steps are interdependent and none occurs in isolation from the others. This process is characterized by participation, interaction, staged continuous decision making, and the use of analytic tools. Participation in the programming process means that the staff of the client organization become involved in information development at the very beginning of the process and are involved throughout. Worksessions are held with staff to

develop programmatic information. Participants are involved at every process stage, implicit in the feedback loop in Figure 8.

This process is also interactive, meaning that the participants and the programmer worked together during information development as well as information analysis. This means the programmer, the participants, and the decision makers all communicate throughout the process and work together to develop a solution. The programmer comes to the process with no preconceived ideas about what the solution should be. The programmer helps the organization to articulate its needs.

This process is also characterized by staged, continuous decision making. At each stage of the programming process the decision making body holds control by exerting its power to resolve conflicts and issues that arise during information development. After each information development stage the programmer extracts issues and conflicts and presents them to the decision making body. The majority of issues and conflicts are resolved through interaction at the participant level, since most solutions are developed within the existing parameters of the programming problem. The focus of the programmer is to help find mutually acceptable compromises within the scope established by the decision makers. Some issues must be resolved at a higher level, and these are presented to the decision making body with enough information to make informed decisions. Two of the advantages of this approach are less time wasted on non-issues and decision making becomes fully informed.

Since participants are working interactively with each other, they develop an understanding of the issues from others' perspectives. Rather than simply defending their own positions, they are able to interactively develop acceptable solutions without the need for a higher decision. Only those issues that can't be resolved face-to-face, or exceed the current scope of the programming problem must be presented to the decision making body for resolution. Otherwise, the decision making body only needs to be kept informed of the solutions being developed. Many issues and conflicts that would otherwise need to be resolved by the decision making body become non-issues. Since there are far fewer issues to resolve, the decision making body is able to spend more time understanding the context for the issues presented, and the programming team is able to provide more information about fewer issues within the same project time and budget.

Decision making is also continuous. The use of microcomputers for information processing allows both participant and decision making body levels of decision to be made continuously throughout a project. Participants and decision makers are advised from the beginning that when it is in the interests of the project to change a previous decision, they are free to do so. Where it is not possible to make a final decision at a particular point in

time, the client is therefore comfortable in making a “subject to future change” decision at that time. All decisions are documented and communicated back to the participants, to ensure that participants are continuously working within current parameters. Goals and parameters are thus clearly defined for the participants throughout the process, even when they change. This type of structured process ensures that the decision making body has legitimacy that is created through the structure of the process.

The last characteristic of the process is the use of analytic tools, called decision support models. Decision support models are computer based numerical models that are quite precise, regardless of the complexity of the project. The impact of decisions can be shown immediately, allowing the testing of what if questions by participants and decision makers. These models also facilitate a continuous decision making process, which allows for changes in decisions in the progress of the process. In the case of the Richmond Campus, a Utilization Model and a Space-Cost Model were used.

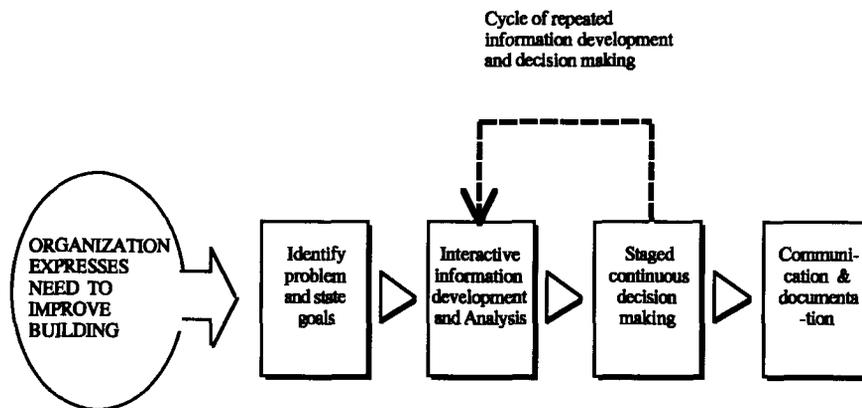


Figure 8: Conceptual Diagram of Programming Process, from Process Four Design & Aitken Wreglesworth Architects, Interactive Design : A Better Way to Involve Users (Vancouver, BC: Process Four Design & Aitken Wreglesworth Architects July, 1991) 5.

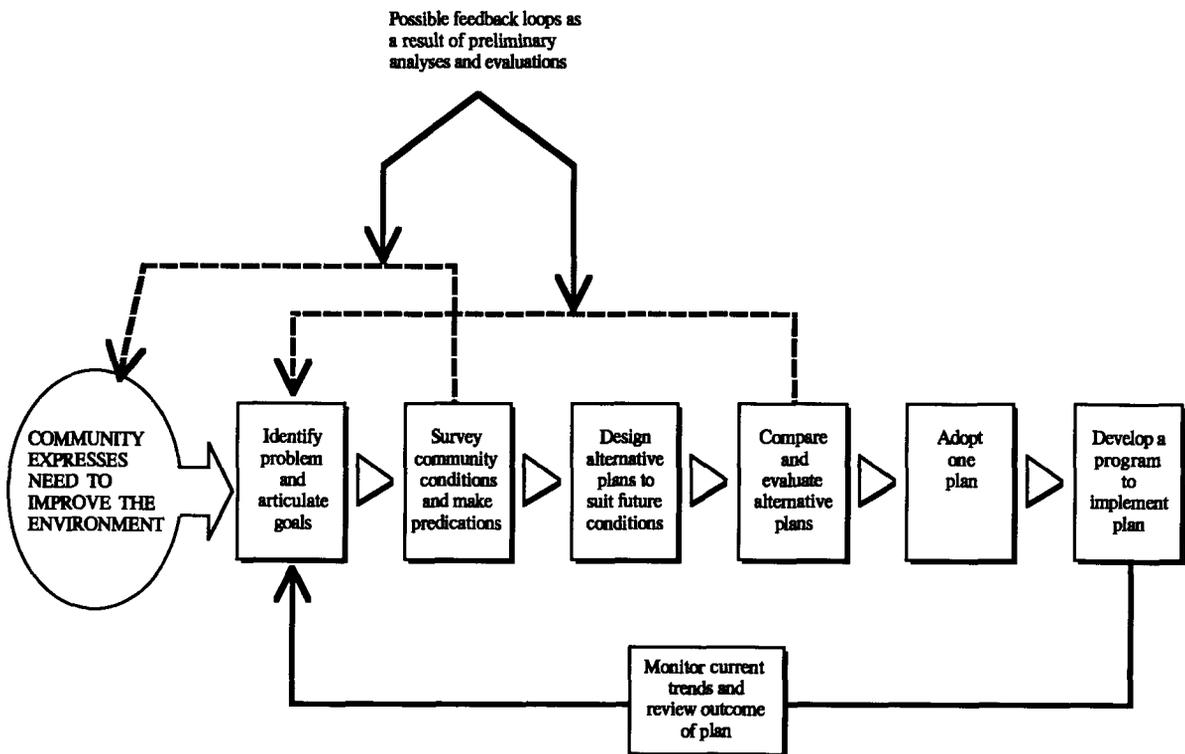


Figure 9: Conceptual Diagram of the Community Planning Process, from Hodge, Gerald, Planning Canadian Communities (Canada: Nelson 1990).

The community planning process is, in theory, very similar conceptually to the programming process. Community planning is also a problem solving activity involving a large number of variables. Figure 9, reproduced from Hodge's book, explains this process (Hodge 1990). Comparing this diagram to Figure 8 on programming, it is evident that the processes and stages are similar, even though the terminology is different. Both processes follow the sequence of identifying problems, stating goals, gathering information, analyzing alternatives, and adopting solutions. Similar to programming, the steps in the community planning process are interdependent. Hodge cites that many of the phases are linked to preceding phases by feedback loops (Hodge 1990). Participation in this theoretical model is also similar to participation in the programming model. The "...community traverses all those steps..." (Hodge 1990). Implicit in the diagram is that input from the community occurs in each phase. This theoretical process is characterized by the interdependence of steps, feedback loops, and the participation of the community throughout the process. This theory is strikingly similar to the programming process.

However, in practice, community planning rarely follows this theory. Participation is generally not similar to the programming process. The public is seldom involved from the beginning of the planning process; they are generally only involved in evaluating the alternative solutions at the end of the process. The planners rarely consult the public during the information gathering stage, and do not consult the public at all during the analysis stage. Solutions are presented to the public and the public can present their input by choosing from among the solutions. This input may or may not affect decisions. The assumption is that the individual citizen does not know what is best, but simply has a point of view. There is no staged continuous decision making in planning. The decision making occurs at the end of the process when solutions are presented to a political body that chooses from among them. This means that in practice, the decision makers have less control, because they do not know what is going on throughout the process. They are simply presented with solutions at the end.

In practice, the goals of community planning are often not clearly defined, and almost always change due to political climate and whims. There is more conflict over goals, because of the number of competing interest groups in the community. The process is seldom clearly structured.

Planners rarely use numerical computer models to inform the public or the decision making body. Planners may use them for their own information or statistical analysis, but do not make them understandable or useable for their clients. Typically, microcomputers are used primarily as word processors and calculators, to speed up the logistical aspects of the planner's work. Microcomputers have evolved tremendously over the past three to five

years. It is this evolution in speed and simplicity of use that allows the development of new tools and new processes. There are a myriad of calculations and “logic paths” (i.e. if A, then B or C, if D, then B or E, etc.) that are currently handled as separate steps or by separate task groups in the planning process that need to be recalculated or re-examined each time decisions are made or changed. The integration of these distinct task groups or problem solving streams into a single computer model is not yet seen in community planning. Once these tools are developed, it is an easy and natural step to use them interactively with the public or decision making bodies.

Participation

Staff participation within the client organization is a very important facet of the programming process. Participation can be controlled by having the participants selected by the Project Committee or the programmer and by absolutely defining the roles of the participants.

The role of the participants is to develop information for the program document, but this type of participation is much more than the simple input of opinions and ideas from the participants. Information is developed with and by the participants. Participants wish to be involved in the process because they have a stake in the facility as they will reside in it. The participants, or users as they are called in programming terminology, have to work out compromises and issues between themselves. They are presented with the problem of defining space needs and asked to find a way to make things work within constraints established by the decision makers. Because of this, a sense of achievement and team spirit is created. As the Continuing Education Director of Kwantlen College said, the process results in total ownership of the product by the users (Baker 1991).

The texture of participation in programming is what I would term fine grained. This means that the participation is highly representative of the overall organization attitudes and opinions. In fact, in the case of Kwantlen College Richmond Campus, the user committees actually were derived from the new Campus educational plan. There was a user committee for each educational division and administrative department.

In community planning, participation seems to work differently, however, it is still an important part of the process. “...Public participation is an important element in the planning process. Opportunity must be given to the public to express themselves on planning issues...” (Ministry of Municipal Affairs 1985). Unlike programming, participation in community planning cannot be controlled at any stage of the process. Planners cannot limit or prescribe involvement in any way; planners have no control over the structure or make up of the participants. It is more difficult to define the role of the participants; participants in the community planning process do not have clear cut roles,

mandates, or goals as in the programming process. It is usual that participants in community planning will have competing goals or purposes, as they may be united by geographic location only. Unlike the programming process, the participants in the community planning process have limited resources to become involved, including time and money. Participants may also not be involved throughout the process. This presents the planner with additional problems.

The way the public are involved is also much different. In community planning, the public is usually involved in a reactive way, then are usually presented with options or alternative solutions. The public is rarely included in the process from the very beginning, as in programming. The public is put in the position of identifying what is wrong with the solutions, but are not invited to assist in determining them. This is not an interactive process, in programming. It is a reactive process that, from the public's perspective, leaps from product to product.

The texture of participation in programming is what I would term coarse grained. This means that the participation is not representative of overall public attitudes and opinions. In fact, the content of the plan may reflect what is acceptable to those in the community that wield influence. It is difficult to obtain agreement in a community about the public interest. "There is almost never total agreement on the priorities to be assigned to different problems and there will certainly be disagreement about their solutions" (Ministry of Municipal Affairs 1985).

Decision Makers

In programming the decision makers are a select group of people, usually the decision makers for the organization. In the case study of Kwantlen College the Directing Committee was in fact the President's Advisory Committee, the administrative decision making body. The appointed project committee made all decisions during the process. Their decisions were based on a clearly established goal defined at the beginning of the project. This goal may be financial, for example: the project must be designed and constructed within a budget of \$22 million. Thus the criteria for decision making in programming is tangible and understandable.

In the community planning process decision making is quite different. Decision makers are elected representatives. "Decisions are reached through the democratic process: elected politicians will reflect what they perceive to be the wishes of the people" (Ministry of Municipal Affairs 1985). The decision making process for community planning is open and democratic. However, the decision making process takes place within the politics of the community. "The form that community planning takes in a community is inevitably a reflection of the political climate..." (Ministry of Municipal Affairs 1985). Issues can be

very complex and decisions may not be as clear cut as they sometimes are in programming. Community planning decisions may affect more people than those of programming decisions. Planning decisions affect people where they live, and people have strong emotional ties to their homes and communities. "Any planning decision is likely to affect a segment of the population..." (Ministry of Municipal Affairs 1985).

Lastly, planners and decision makers have no clearly set goals to measure decisions against. Goals and priorities are constantly changing. This ensures that decision making will always be difficult and the public may not find it easy to rationalize.

Scope

Programming occurs at a micro scale, and is usually based on a single facility. However, to describe the scope of community planning is not as simple a task as it is characterized by a macro scale. Community planning usually presents a holistic view, and a broad based comprehensive perspective. Community planning, by its nature, is long range and forward looking (Chapin & Kaiser 1979). It is continuous and is future focussed. However, community planning can also be micro and have a specific focus. The scope of community planning is thus not clearly defined, it depends on the particular project. For example, community planning may involve the planning for a single park, or may involve the planning of a regional park system.

Implementation

Implementation in programming occurs through the design of the facility and is subject to the interpretation of the designer. Implementation usually occurs quickly after completion of the program document., as the client organization wishes to get the building built as soon as possible because of time and money constraints.

Implementation in community planning occurs through the development of the community and is subject to the interpretation of the developer. Implementation does not occur as quickly as in programming. This is because most implementation occurs through reactive planning tools, such as zoning and subdivision controls which constrain or limit development decisions (Hodge 1990). The planners cannot count on investments or other action by the community because a plan simply guides the development decisions and is not a proactive statement like a program document. Thus, the implementation of a community plan depends on the amount of development taking place in a community. Neither planners nor council have much control over the amount or timing of these investment decisions.

Client

Programming is undertaken within a public or private organization on its behalf. The client is the organization that usually has established or implicit mandates and philosophies that drive its operations, and has the widely recognized goal to build a new facility. The

client is a single organization that has clear goals and mandates to fulfill in the programming exercise.

In community planning, the nature of the client is somewhat vague. The planner's client is the political body, as well as the public at large. The planner's professional responsibility is to the political decision making body, but the planner's social responsibility is to the public. Sometimes this mixed allegiance becomes confusing. Also, the planner's clients have no clear directions or goals as in programming. Usually, they hold a myriad of competing goals.

Role of Professional

The role of the programmer is to set up processes for information generation and decision making. The programmer serves the client organization, and is financially reimbursed by this organization. There is an established professional relationship between the programmer and the organization. The role of the programmer is that of a facilitator. The view held by programmers, as in the Kwantlen College case study, is that the client organization knows most about the facility they require. The programmer's role is to facilitate the process of generating that information with the organization. The programmer holds no preconceived notions about the outcome of the project. The programmer simply helps the organization to articulate the best solution.

In theory, the role of the planner is similar to that of the programmer. The role of the planner is to set up processes for information generation and decision making. The planner serves the public and the governmental body, but is employed by the governmental body. Professional planners are people who assemble and analyse relevant information to help decision makers make choices. The "...planner's role is to obtain the relevant information and present it in a balanced manner..." (Ministry of Municipal Affairs 1985). Similar to the programmer, the planner's theoretical role is also that of a facilitator. The "...planning of a community is a political process in which the professional planner's role is to advise the participants and facilitate the process..." (Ministry of Municipal Affairs 1985).

In practice, the planner's role seems to be different. The community planner's role is frequently that of an expert. The planner gathers information, researches, and makes recommendations to the political decision making body. This assumes that the planner knows what is best for the community, and has information and knowledge that others do not possess. This is a very different role than that of a programmer, who has no agenda or preconceived notion of a solution.

Product

The final product of the programming process is the program document. However, the program is viewed as a changing target. The document is not a definitive statement; it will

change as the design proceeds. The document does not sit on a shelf unused. It is used extensively by the designers. It is important that the product be understandable and easy for both the client and the designer to use. The program is usually put into a temporary binder to permit the insertion or replacement of pages after publication, as change will undoubtedly occur.

The product of the community planning process is the community plan. In theory, community plans are also seen as a changing target. "...Plans are viewed as documents which evolve under the pressure of circumstances, not a definitive plan which will never be changed..." (Ministry of Municipal Affairs 1985). In practice, however, planning documents are often seen as dogmatic and fixed. The logistics of changing a community plan are onerous, and therefore changing a plan seldom occurs. The product in planning is usually not definitive because there is no clear client to carry out the plan. The product could be an Official Community Plan, a zoning bylaw, or some design guidelines. Unlike the program document, the planning document is often difficult to understand. It is often cryptic and expressed in terms that require an understanding beyond that of the document itself. The public also rarely looks at planning documentation from an overall perspective. They typically have a more myopic view, using or criticizing only that part which affects them or their project directly.

Time Frame

The time frame of the programming process is relatively short compared to community planning. Time frames vary from one week upwards. Time frames in programming are critical, because the client organization is interested in moving quickly as time is money. Programming occurs within the control of one organization, thus the scope and number of conflicts may be minimal compared with those of community planning. In programming, all participants typically have a shared corporate or organizational focus, and since a routine programming objective is to improve the conditions and/or environment through facility change, the process tends to act as a catalyst for conflict resolution. Participants have a number of positive motives for conflict resolution, and this also drives the shorter time frame.

Community planning has a relatively long time frame compared to programming, and time frames seem to be less important. The time element in community planning is continuous and into the future. Planners are not working within one organization; they are working with a myriad of interest groups and organizations within a community. The number and scope of conflicts may be quite large, encompassing the entire community. Additionally, not all interest groups share the same objectives of change as the status quo is frequently preferred. In fact, few planning processes are conducted in a context where

there is an expressed, let alone urgent, need for change from within the community. It is usually the planners who are seeking to implement change. The unknown consequences of the planning process can be kept at bay by nurturing, rather than resolving, conflicts.

In conclusion, programming and community planning theory are similar. In fact, of the ten categories of characteristics examined in this chapter and summarized in Table IV, four (Purpose, Technique or Method, Role of Professional, and Product) are virtually interchangeable and two more (Scope and Implementation) are very similar. One characteristic (Participation), could be modelled after programming, while the remaining three categories (Decision Makers, Client and Time Frame) are clearly different. These three areas of difference between programming and community planning do not significantly affect the opportunities to apply the ideas derived from this case study to community planning. The fact that there are differences in the two processes does not preclude the potential to benefit from incorporating the four ideas into the planning process. These four ideas to improve the community planning process are to make it more participative and interactive than traditional community planning as well as supported by staged continuous decision making and good analytic tools.

The first two ideas, more participative and interactive, are transferable to four categories (Technique or Method, Participation, Implementation, and Role of Professional). Staged continuous decision making is relevant to three categories (Purpose, Decision Makers, and Client). The fourth idea, good analytic tools, is not only relevant in reducing the logistical load of the last three categories (Scope, Product, and Time Frame), but is the catalyst to facilitating the application of all four ideas.

Why haven't these ideas been widely applied to community planning in practice? There may be many obstacles, unique to planning, that prevent their application. It is also critical to note that this thesis compares the leading edge of one profession against general practice, or the average, in another.

Community planning largely occurs in the public realm, within organizational environments that encourage application of known (and consequently predictable and safe) processes. These factors alone are sufficient to account for the fact that ideas at the leading edge of programming have not widely appeared in community planning. Once planners have increased access to more powerful and transportable microcomputers and software that have only been available for the past few years, things might change. Once they are given the mandate to "do it differently in order to improve", things might change.

5.2 Consideration of Ideas that can be applied to Community Planning

The findings of the case study on participation, interaction, staged continuous decision making, and the use of analytic tools, are explored here for their application in the community planning process. How might these four ideas work if applied to community planning? Why might the ideas not work? Each idea and its applicability to community planning will be discussed separately.

Firstly, it can be argued that participation, as used in the case study, may be applied in community planning, so that public involvement in community planning could more resemble Kwantlen College staff involvement in programming. At the beginning of the planning process, a role could be clearly articulated for the public, as was the case for the Kwantlen staff and faculty members in the Richmond Campus project. The public could be informed that their role is to provide information, and not to make decisions. The public could be involved from the very start of a project and be clearly informed of the interest groups involved and the constraints of the project, in particular the financial constraints. The public could then provide input within a setting of clear parameters and real constraints. This process would indicate that the public was credited with knowing what is best for their community.

Following this model, a potential community planning process may consist of a community group being told how much money is allocated for a community project, and the community group deciding among themselves how to achieve the project within that budget. Conflicts and issues would be decided by the decision making body throughout the process. The public might find it easier and more fun to provide input and help when they know their role, why they are involved, and what the real constraints of the project are.

Secondly, findings on interaction in the Richmond Campus programming project could be applied to community planning. The information flow of the Richmond Campus project could be used in community planning, that is the ultimate decisions would rest with the the political body, and the input (within real life constraints) would be the responsibility of the involved community groups

It is critical that the role of the public be clearly outlined and understood, in order that their involvement become legitimated. It is also important that people's input not disappear, but be recognized even if it is not used. All input should be preserved in some kind of documentation. Decisions made by the political body should be fed back to the public somehow, along with the reasons. This will make the public realize and feel that their input is valued and is being heard. It would also begin to mitigate problems of continuity.

The planner's role in this process would be that of a facilitator with no preconceived ideas about a solution. A dialogue between the professional planner and the public would occur, based on "what if", "how much", etc. The planner would facilitate information development with the public. Conflicts and issues would be referred to the decision making body.

Specifically, working papers could be used as a tool to accomplish some of the participation and interaction ideas. Instead of presenting the community with a finished product, the constituents could take part in the development of ideas and products. They could be kept up to date on projects through working papers. This would solve many problems of poor feedback mechanisms to the public. It would also show to the public how their input was being used, and that their input was valued. A working paper would include all input, whether it was used or not.

Why might the findings on participation and interaction not work in community planning? There is the problem of continuity of participants and continuity of process. The public's involvement in planning occurs at their whim. It is not a controlled type of participation. The planner has no control over the membership of the participants as the programmer does. The same participants may not show up at consecutive meetings. The public has limited time, and they participate in planning when they are interested in a specific issue. This is a real problem in the legitimacy and continuity of the process. It means that participants may not be clear on their role or the development of information to date, or how issues were resolved. Things may become reopened and may need to be dealt with over again.

Thirdly, the finding on staged continuous decision making, as used in the Richmond Campus programming project, could be used in the community planning process. Information development would take place through and by the participants, but all conflicts and issues would be resolved by a decision making body. The planner would present the decision making body with issues, and the debates around them. The decision making body would then make a decision based on that information. It is possible that the decision making body could be an appointed body made up of community members, or it could be the elected politicians. The benefit of having the body made up of community members would be that those members would not bring changing priorities or goals that many politicians would bring. And the process of community planning could become legitimized through constant input and control cycles. The public would then understand decisions made throughout the process and the solution at the end of the process would be theirs. That is, the public would "own" the product. The final solution would not come as a fait accompli or a surprise to them, because they would have been involved from the

beginning of the process developing the solution. This would result in ownership of the product.

Why might this not work? There may be problems with the concept of community members being appointed to a decision making body. It may not be possible in British Columbia municipal government to give decision making power to an appointed committee of community members. This type of committee may only be able to have advisory powers. Recommendations forwarded to the elected politicians may or may not be seriously considered by the politicians. It depends on how the politicians would view the committee and its recommendations. Sometimes the politicians may support the recommendations, and sometimes they may question them. Another possible problem with appointing community members to a committee is that of continuity. These members would need to be dedicated individuals, and must be available for regular meetings. This problem may be solved through limiting the length of an individual's term. An individual may serve only one year, and then a newly appointed member may serve. Staged continuous decision making may also not work if the elected politicians were the decision making body. It is not that the time involved is much greater, but because the politicians bring other agendas and priorities to decision making. Community members may or may not have these same agendas. If community members do not have the same agendas, decisions made by the politicians will never be understood and will undoubtedly raise major conflicts.

Lastly, analytic tools could be used in the community planning process. The most significant tool that community planning could adopt from programming is greater use of micro computers. Micro computers are critical in working with changing parameters and variables inherent in both the community planning and the programming processes. Micro computers allow rapid processing and recalculation of data at a speed that was unthinkable a few years ago. Decision support models can be set up on micro computers that simulate real life planning situations. Logic and relationships between variables could be set up to simulate any situation. These models could minimize the logistical barriers to dealing with the ambiguities in planning situations. Models could show what impacts a decisions will have and this would provide realistic knowledge on which to base decisions.

Planners could use interactive decision support models set up on micro computers. These could be used for most kinds of projects to outline the constraints and parameters. The impact of decisions could quickly be calculated and planners could involve constituent groups in their use. Planners and constituents could work together in understanding the impacts of decisions about the community, whether it be development, new parks, or other improvements.

As in the Space Cost Model for the Richmond Campus, which linked space to costs, the planners for a community could produce a spreadsheet to show the different options for redevelopment. For example, the residents may want a property to become a park, and the decision support model could show if the cost of upkeep or of development is within the budget provided. What if questions could be answered through the use of such a model. The model could be used to raise questions, and to get the input of various groups. This would be a more interactive, less solution oriented process. The use of decision support models would allow constituents to work within defined parameters. This would allow the public involvement in a project where they knew their input was being used.

Community planning may have difficulties with the use of analytic tools, such as computer models because some information in community planning is qualitative and such data does not lend itself to numerical calculation on a spreadsheet.

In conclusion, it is not a simple task to state definitively that community planning can use participation, interaction, staged continuous decision making, and analytic tools. Community planning is a confusing and complex profession. The best we can do in this thesis is to outline ideas that the community planning profession should think about. As previously stated, planning occurs largely in the public realm, generally within organizational environments that encourage application of known and safe processes. Once planners have been given the mandate to do it differently in order to improve, we may begin to see the implementation of some of these ideas.

The principal finding of this thesis that community planners can benefit from is that the input of the public should be considered important and valued and their involvement can potentially be structured to benefit rather than hinder the planning process. The attitude of the community planner is critical. If the planner is the expert, why would public participation be a necessary and important part of the community planning process? The public will become involved and active when they believe that the planner is a good listener and is meeting with the public to get important insights and information from them. The planner should believe that the community's residents are the experts in how their community is developing and how they want their community to develop. In fact, I believe that the public may offer the best solutions or ideas; ones that planners may not have thought of.

Further research is needed on the constraints in using the finding outlined in this thesis. The question that needs to be answered is why community planning has not implemented in practice what it has outlined in theory. Further research also needs to be done on community planning processes in practice. Specifically, case studies of community

planning should be done to the level of detail that the case study of programming in this thesis achieved. When the organizational atmosphere of planning changes and when additional research begins to impact on the community planning profession, I believe we will begin to see the implementation of the four leading edge aspects of programming achieved in this case study.

REFERENCES

- Baker, Gary, Director of Continuing Education, Kwantlen College. 1991. Personal Interview.
- Becker, F.D. 1977. User Participation, Personalization, and Environmental Meaning : Three Field Studies. Program in Urban and Regional Studies. Cornell University.
- Bergman, Monique, Director of Bookstore, Kwantlen College. 1991. Personal Interview.
- Bevard, Joseph H. 1985. Capital Facilities Planning : A Tactical Approach. Washington, D.C: Planners Press.
- Bobrow, Phillip. 1974. "Experimental Changes to the Architectural Process." Industrialization Forum: 9 -19.
- Boothroyd, Peter, Associate Professor. 1989. Notes and Exercises for Community Planning Course held summer 1989 at Gitksan-Wet'suwet'en Tribal Council Boardroom. Vancouver, BC: School of Community and Regional Planning, University of British Columbia.
- Boothroyd, Peter, Associate Professor. 1989. Developing Community Planning Skills : Applications of a Seven-Step Model. Vancouver, BC: School of Community and Regional Planning, University of British Columbia.
- Boothroyd, Peter and Anderson, Owen. 1984. "The Difference Between Corporate and Social Planning, and the Implications for Indian Affairs." Saskatchewan Forum: 1-12.
- Brill , M. and Masterson, C.H. 1972. Some Thoughts on the Direction of User Requirements Research. Washington, D.C: The American Institute of Architects.
- Canadian Handbook of Practice for Architects. 1976. Ottawa, Ontario: The Royal Architectural Institute of Canada, 2: 2.
- Caulfield, DA and Handa, VK. 1974. "The User's Role in Programming and Project Management." Industrialization Forum: 57 -58.
- Chapin, F.S. & Kaiser, E.J. 1979. Urban Land Use Planning. Third Edition. Champaign, Illinois: Univeristy of Illinois Press.
- Davis, Gerald and Szigeti, Francoise. 1979. Functional and Technical Programming: When the Owner / Sponsor is a Large or Complex Organization. Louain-la-Neuve: Proceedings of the Fourth International Architectural Psychology Conference, July 10-14, 1979.

- Designing for Human Behaviour : Architecture and the Behavioural Sciences. 1974. Lang, Jon, Editor. Stroudsburg, Pennsylvania: Downden, Hutchinson, & Ross, Incorporated.
- Dluhosch, Eric. 1974. "Flexibility/Variability and Programming." Industrialization Forum: 39-46.
- Duggan, Barbara, Faculty of Interior Design, Kwantlen College. 1991. Personal Interview.
- Evans , B.H. and Wheeler, C.H. 1969. "Emerging Techniques 2 : Architectural Programming." The American Institute of Architects.
- Facility Planning Technology. 1987. Conway, M. & L. Liston, Editors. Atlanta, Georgia: Conway Data, Incorporated.
- Facility Program, Kwantlen College Surrey Campus. July, 1987. Vancouver, BC: Cornerstone Planning Group.
- Faludi, Andreas. 1973. Planning Theory. Permagon Press.
- Five Year Plan, 1989-1993. December, 1988. Vancouver, BC: Kwantlen College.
- Francis, Derek, Dean of Educational Support, Kwantlen College. 1991. Personal Interview.
- Gauld, Don. 1986. Public Participation and the Preparation of Official Community Plans in British Columbia. Master's Thesis. Vancouver, BC: School of Community and Regional Planning, University of British Columbia.
- Gutman, Robert. 1969. "The Sociological implications of programming practices." Building Research 6 (April - June, 1969): 26.
- Hales, H.L. 1984. Computer-Aided Facilities Planning. New York: Marcel Dekker, Incorporated.
- Heath, Tom. 1974. "Programming and User Involvement : an Issue Based Approach." Industrialization Forum: 3 -8.
- Hodge, Gerald. 1990. Planning Canadian Communities. Canada: Nelson.
- Horowitz, Harold. May 1967. "The program's the thing." AIA Journal 47: 94-100.
- Klosterman, Richard. 1976. Toward a Normative Theory of Planning. PhD Thesis, Cornell University.
- Kwantlen College Site Analysis. July, 1987. Vancouver, BC: Cornerstone Planning Group.
- Kwantlen College Regional Development Strategy. February, 1987. Vancouver, BC: Cornerstone Planning Group.

- Kwantlen College Richmond Campus Facilities Program. February, 1990. Vancouver, BC: Process Four Design.
- Lombard, Francois. 1974. "An Organized Process for Programming - Application to the Centre Beaubourg." Industrialization Forum: 31 -38.
- McDonald, Cathy, Librarian, Kwantlen College. 1991. Personal Interview.
- MacLaughlin, Adrienne, President of Kwantlen College. Letter to Jim Sumi of Process Four Design, April 06 1989.
- MacLaughlin, Adrienne, President of Kwantlen College. Letter to Jim Sumi of Process Four Design, March 20 1989.
- Marketing Plan. January, 1988. Vancouver, BC: Kwantlen College.
- Martel, A and Ignazi, G. 1974. "An Experiment with Adaptable Housing at Montereau." Industrialization Forum: 59 -64.
- Ministry of Municipal Affairs. 1985. An Introduction to Community Planning. Ontario: Queen's Printer.
- Nanson, Derek, Faculty of Adult Special Education, Kwantlen College. 1991. Personal Interview.
- The New Lexicon Webster's Dictionary of the English Language. 1988. 1988 Edition. New York: Lexicon Publications.
- Palmer, Mickey, A.. 1981. The Architect's Guide to Facility Programming. Washington, D.C: The American Institute of Architects.
- Peña, William M., with William Caudill and John W. Focke. 1977. Problem Seeking - An Architectural Programming Primer. Boston: Cahnern Books International.
- Peter-Cherneff, Brigitte, Librarian, Kwantlen College. 1991. Personal Interview.
- Preiser, F.E. 1978. Facility Programming : methods and applications. Stroudsburg, Pennsylvania: Dowden, Hutchinson and Ross, Incorporated.
- Process Four Design & Aitken Wreglesworth Architects. July, 1991. Interactive Design : A Better Way to Involve Users. Vancouver, BC: Process Four Design & Aitken Wreglesworth Architects.
- Quick Facts. January, 1989. Vancouver, BC: Kwantlen College.
- Rathie, Bob, Faculty of Sciences, Kwantlen College. 1991. Personal Interview.
- Salomaa, Diana Rita. 1981. Planning Strategies for Canadian Urban Planners: A Case Study of Regina. Master's Thesis. Vancouver, BC: School of Community and Regional Planning, University of British Columbia.
- Sanoff, Henry. 1977. Methods of Architectural Programming. Stroudsburg, Pennsylvania: Dowden, Hutchinson, and Ross Incorporated.

- Slattery, John, Director of Design, Kwantlen College. 1991. Personal Interview.
- Struder, Raymond G, and David Stea. October 1966. "Architectural Programming and human behaviour." Journal of Social Issues 22: 127-136.
- Sumi, Jim, Principal of Process Four Design. 1991. Personal Interviews.
- Sumi, Jim, Principal of Process Four Design. Letter to MacLaughlin, Adrienne, President of Kwantlen College, March 06 1989.
- Three Year Plan, 1991-1994. 1990. Vancouver, BC: Kwantlen College, October.
- Tompkins, James A. & White, John A. 1984. Facilities Planning. Washington, DC: John Wiley & Sons.
- Warsaw, L. 1974. "Programming for Participation." Industrialization Forum: 45-56
- White III, Edward, T. 1972. Introduction to Architectural Programming. Tuscon, Arizona: Architectural Media.
- Wiesman, B. 1977. Facility Planning and Program Rationalization : Post Secondary Education in the Lower Mainland. British Columbia: Ministry of Education.

APPENDICES

APPENDIX A : TABLE OF CONTENTS OF KWANTLEN COLLEGE RICHMOND FACILITIES PROGRAM

EXECUTIVE SUMMARY	i
TABLE OF CONTENTS	iv
DOCUMENT USE & ORGANIZATION	
PURPOSE	1
DOCUMENT ORGANIZATION	1
GLOSSARY OF TERMS	4
ACKNOWLEDGEMENTS	5
CONTEXT	
THE SITE	6
PROJECT PARAMETERS	13
PLANNING CONCEPTS	15
SPACE TYPES	22
STANDARD SPACE	28
PROGRAM SUMMARY	
COMPONENT GROUPS	35
SUMMARY OF SPACES	
1.0 APPLIED ARTS AND CAREER PREPARATION	37
2.0 DESIGN CENTRE	39
3.0 ACADEMIC AND APPLIED STUDIES	41
4.0 EDUCATIONAL SUPPORT	43
5.0 ADMINISTRATIVE SERVICES	45
COMPONENT DESCRIPTIONS	
1.0 APPLIED ARTS AND CAREER PREPARATION	
1.1 Adult Basic Education Program	1-1
1.2 Adult Special Education Program	1-6
1.3 Business Systems Technician Program	1-10
1.4 Continuing Education	1-12
1.5 Drafting Program	1-15
1.6 Early Childhood Education Program	1-20
1.7 Office Administration Program	1-26
1.8 Public Safety Dispatcher Program	1-30
2.0 DESIGN CENTRE	
2.1 Fashion Design and Marketing Program	2-1
2.2 Foundations in Applied Design Program	2-8
2.3 Graphics and Visual Design Program	2-13
2.4 Interior Design Program	2-21
2.5 Mass Communication Program	2-26
3.0 ACADEMIC AND APPLIED STUDIES	
3.1 Business and Career Specialties Department	3-1
3.2 Humanities and Social Sciences Department	3-7
3.3 Science, Applied Science, and Technology Department	3-13

APPENDIX A continued

4.0 EDUCATIONAL SUPPORT

4.1 Main Entry	4-1
4.2 Admissions and Registration	4-4
4.3 Counselling	4-8
4.4 Financial Aid	4-11
4.5 Job Placement	4-14
4.6 Library	4-15
4.7 Instructional Resources	4-17
4.8 Technical Services	4-30
4.9 Student Association	4-36

5.0 ADMINISTRATIVE SERVICES

5.1 Administration	5-1
5.1.1 Campus Administration	5-2
5.1.2 Educational Administration	5-6
5.1.3 General Administration	5-10
5.2 Bookstore	5-18

Appendix A : Table of Contents of Kwantlen College Richmond Facilities Program, from Kwantlen College Richmond Campus Facilities Program (Vancouver, BC: Process Four Design, February, 1990).

APPENDIX B : COMPONENT DESCRIPTION FOR ADULT BASIC EDUCATION PROGRAM

FUNCTIONAL DESCRIPTION

Adult Basic Education (ABE) will provide students with high school level skills in a variety of subject areas. This program will act primarily as a feeder into other Kwantlen programs, but will provide upgrading to students for personal reasons or for entry into other institutions. Course work is presently a mixture of group instruction and individual work. Students complete course modules at their own pace.

Adult Basic Education currently operates on a continuous intake basis with students able to enter the program throughout the academic year. Full time students attend during the day, between the hours of 8AM and 4PM. Part time students attend either mornings, afternoons, or evenings. Evening classes are offered four days/week, from 6PM to 9PM. Classes operate 10 months of the year from September to June.

ENROLLMENT

56 FTE

STAFFING

3 full time day instructors.
2 half time evening instructors.

FUTURE

Future enrollment in Adult Basic Education will grow significantly. While the method of instruction will not change dramatically, the focus will change to increased small group instruction.

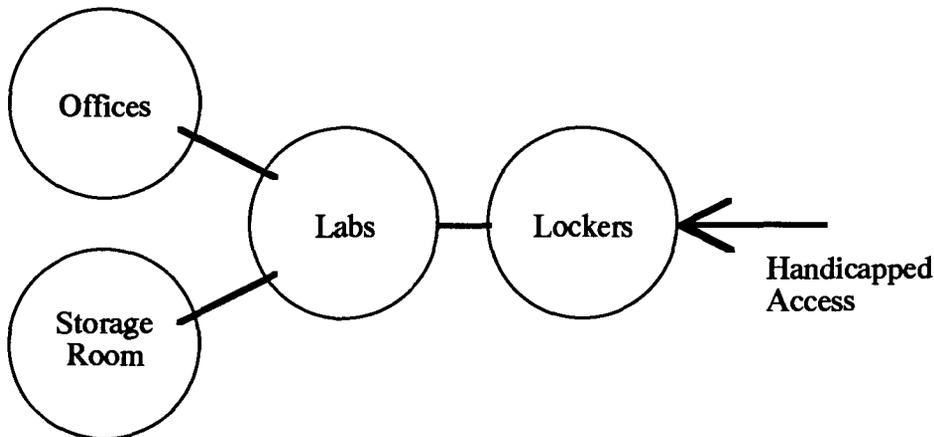
NEIGHBOURS

Establish the outside areas that will relate to the component. These related areas are ranked in order of importance.

Neighbours	Reason for Closeness
1. Admissions and Registration	Counsellors send students to ABE, and ABE send students to counselling.
2. Student & Staff Lounges	For ease of student and staff access.
3. Food Services	
4. ECE	Easy access to daycare facilities for use by students.
5. Library	Access to resources (A/V equipment).
6. Bookstore	Access to supplies.
7. Campus Administration	Secretarial support.
8. Office Administration	ABE acts as student feeder
9. Academic Programs	ABE acts as student feeder
10 Drafting Program	ABE acts as student feeder
11 Business courses	ABE acts as student feeder

INTERNAL RELATIONSHIPS

This diagram represents the functions within a component and how they relate to each other. The diagram makes the first step towards relating zones of space.



SPACE DATA

The spaces required to accommodate the needs of this component are described following. Bold lettering relates a space or group of spaces to the "Internal Relationships" diagram previously outlined. If more than one of a particular space is required, the number is indicated in parentheses immediately following the space name. The allocated area in Net Assignable Square Feet -NASF (Net Assignable Square Metres-NASM) is also listed. A brief description of the space, as well as a conceptual diagram, may also be provided.

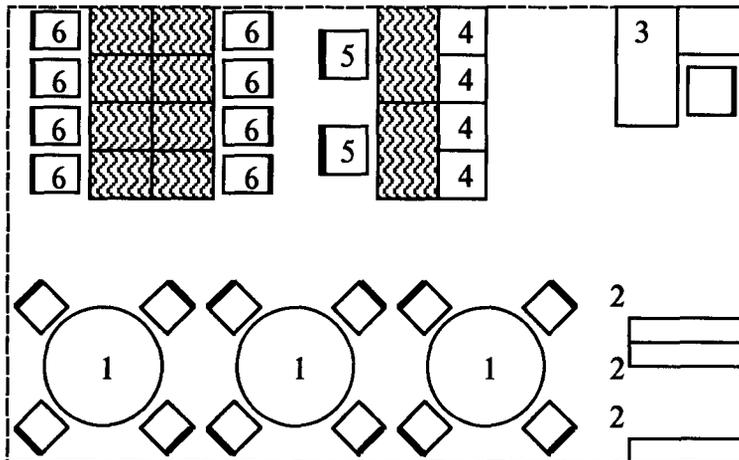
These diagrams should **not** be interpreted as the required disposition of space, furnishings and equipment. They are simply included to assist the reader in achieving a general understanding of the desired space.

Labs

Lab, ABE Level 1

600 NASF (55.7 NASM)

A maximum of 22 students will move about the room to access reference materials, sit at a table, or occupy a study carrel or computer station.

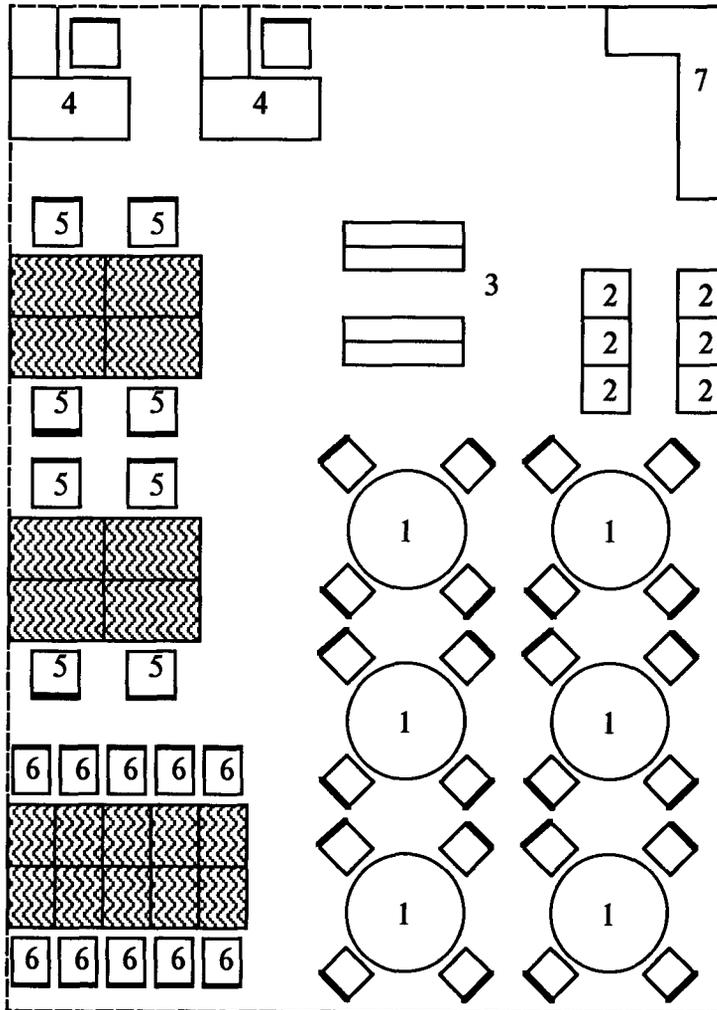


1. Round Table w/Chairs
2. Bookcase
3. Desk w/Runoff & Chair
4. Filing Cabinet
5. Computer Worktation w/Chair
6. Study Carrel w/Chair

Lab, ABE Level IIA

1330 NASF (123.6 NASM)

A maximum of 43 students will move about the room to access reference materials, sit at a table, or occupy a study carrel or computer station.



- 1. Round Table w/Chairs
- 2. Filing Cabinet
- 3. Bookcase
- 4. Desk w/Runoff & Chair
- 5. Computer Workstation w/Chair
- 6. Study Carrel w/Chair
- 7. Science Corner w/Sink

Classroom, 26 Seat

Shared use of spaces allocated to 1.3 Business Systems Technician.

Lockers

Design solution will attempt to place lockers in circulation space.

Office, Shared Faculty

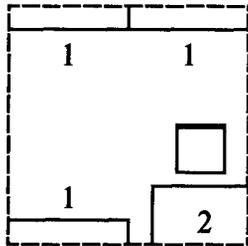
2 @ 160 NASF = 320 NASF (29.7 NASM)

See generic Office, Shared Faculty.

Storage Room

100 NASF (9.3 NASM)

Includes one workstation for a student assistant.



- 1. Bookcase
- 2. Workstation Desk w/Chair

Appendix B : Component Description for Adult Basic Education Program, from Kwantlen College Richmond Campus Facilities Program (Vancouver, BC: Process Four Design, February, 1990).