

«АРХИТЕКТУРНОЕ ПРОГРАММИРОВАНИЕ» КАК СОЦИАЛЬНОЕ ИССЛЕДОВАНИЕ И ПРЕДПРОЕКТНЫЙ МЕНЕДЖМЕНТ

УДК: 72.067.2
ББК: 85.113(2)7

Кияненко Константин Васильевич

доктор архитектуры, профессор,
Вологодский государственный университет,
Вологда, Россия, e-mail: design@mh.vstu.edu.ru

Аннотация

Одно из ключевых для проектирования и общеупотребительное за рубежом понятие «архитектурное программирование» настолько не известно в России, что анализу его теоретических проблем и положений по необходимости должно предшествовать хотя бы краткое изложение последних. На основе классических текстов по данной теме в статье рассматриваются происхождение и эволюция концепции при переходе от традиционного общества к постиндустриальному, сравниваются разные представления об уровнях и моделях программирования, оценивается роль архитектора как «программиста», прослеживаются возможные связи программирования с проектированием, исследуется структура и содержание программной процедуры, формы представления ее итогов заказчику и другим членам «программного комитета/команды». Делаются выводы об актуальности теоретической разработки программирования как дисциплины и практического использования ее методов в профессиональной отечественной практике.

Ключевые слова

архитектурное программирование, социально-архитектурные исследования, проектное задание

Введение

Данный текст не имеет никакого отношения к компьютерной проблематике. Понятием «архитектурное программирование» за рубежом очерчивают особый вид профессиональной предпроектной научно-исследовательской и управленческой практики. Его суть – уточнение ключевых проблем, на решение которых будет направлено проектирование, и разработка того, что у нас называют «техническим заданием», а в англоязычных странах чаще всего – «архитектурной программой» (англ. architectural program) или «проектной инструкцией» (англ. brief). Собственно, сама необходимость делать для отечественной научной и профессиональной аудитории такого рода пояснение подтверждает актуальность первой из поставленных автором целей – научно-просветительской. За пределами России создана значительная академическая традиция изучения и формирования архитектурных программ и еще более впечатляющая практика воплощения теоретических изысканий. Даже предварительное знакомство с этим материалом будет способствовать распространению в архитектуре России более зрелых социально-фундированных профессиональных подходов. Это тем более актуально, что действующие федеральные государственные образовательные стандарты по направлению «Архитектура» уже пять лет как требуют от выпускников вузов знать «разработку заданий на проектирование», обретения соответствующих умений и компетенций. Но доступной студентам и педагогам русскоязычной литературы, раскрывающей и исследующей мировой опыт на эту тему, насколько нам известно, не существует. Вторая цель, которую пытается достичь автор, – содержательный анализ ключевых англоязычных работ по архитектурному программированию с точки зрения различий в трактовке этого феномена. Такие различия имеются, и их рассмотрение способно значительно углубить понимание темы.

Автор намеревался придерживаться следующей логической цепочки действий: определение исследуемого феномена, обозначение границ и указание источников – анализ

истории развития феномена и выявление закономерностей его эволюции – принципиальное содержание феномена как исследовательской процедуры, ее ключевые действующие лица – классификация встречающихся подходов и материализации результатов. В одном пришлось отклониться от описанной последовательности: трудно определить постоянно развивающийся предмет без отсылки к истории развития, поэтому две первые позиции поменялись местами. Итак, двигаясь к поставленным целям, автор рассматривает следующие вопросы: краткую историю становления архитектурного программирования (АП) как научной темы и раздела профессиональной практики; бытующие в теории архитектуры толкования АП; типологию разновидностей АП с учетом многообразия сфер и эволюции практики; уровни и модели разработки архитектурных программ; точки зрения на то, кто может и должен быть «архитектурным программистом», обоснованность претензий архитектора играть эту роль; представления о месте АП в структуре профессиональной работы, связи с проектированием; структуру и содержание процесса АП; формы представления архитектурных программ.

Метод исследования – анализ теоретических и документальных источников. Их первый слой – ключевые и общепризнанные за рубежом издания пионеров и экспертов по данной теме – Вильяма Пенья (W. Peña), Эдит Черри (E. Cherry), Доны Дюрк (D. Duerk), Роберта Хершбергера (R. Hershberger), Вольфганга Прайзера (W. Preiser)¹. Второй слой рассматриваемых источников – известные тексты по теории архитектуры и архитектурной деятельности, касающиеся темы АП, в том числе монографии и учебники Джона Зайсела (J. Zeisel), Джона Ланга (J. Lang) и Брайана Лоусона (B. Lawson). В некоторых аспектах оказались полезными отечественные публикации, по данной теме немногочисленные, – работы В.Л. Глазычева и А.Г. Раппапорта. Третий слой – профессиональные и образовательные регламенты и стандарты, в которых раскрывается роль АП в практике и обучении архитектуре. Данные источники рассматривались с позиций понятийно-терминологического, структурно-функционального и исторического анализа. Наконец, автор знакомился с опытом разработки и представления архитектурных программ в вузах и архитектурной практике и моделировал выводы графически.

Краткая история архитектурного программирования

Как пишет Э. Черри, архитектурное программирование началось вместе с архитектурным проектированием и археологов, раскапывающих старые постройки, более всего интересуют скрытые в них программы общественного устройства прошлых поколений [1]. Доиндустриальное (традиционное) общество выработало механизмы встраивания проектных программ в устойчивые социально-морфологические структуры – типы или, по А.Г. Раппапорту, «прототипы» зданий². Именно прототипы «задают основу и обеспечивают осуществление связи социального функционирования объектов, требований, предъявляемых к ним со стороны потребителей и социальных заказчиков, производства и самого проектирования» [2]. На этом этапе плавность и непрерывность процессов, устойчивость общественного устройства, стабильность социальных целей, потребностей, регламентов и культурных норм, а вместе с этим – прототипов зданий и лежащих в их основе программ, делали связь «прототип – социальная программа» почти нерушимым целым, воспроизведение одного гарантировало репродукцию второго.

Нужда в архитектурном программировании как специализированной деятельности возникает с приходом индустриальной революции, в условиях бурного роста масштабов, разнообразия социально-культурного устройства жизни, типов и сложности проектируемых объектов [3]. Правда, не сразу разработка программ осознается как ответственность архитектора. По В.Л. Глазычеву, до 1920-х годов доминирует модель профессиональной деятельности, в которой архитектор связан с индивидуальным клиентом, если и не умеющим детально сформулировать программу в целом (что «нужно»), то хотя бы способным

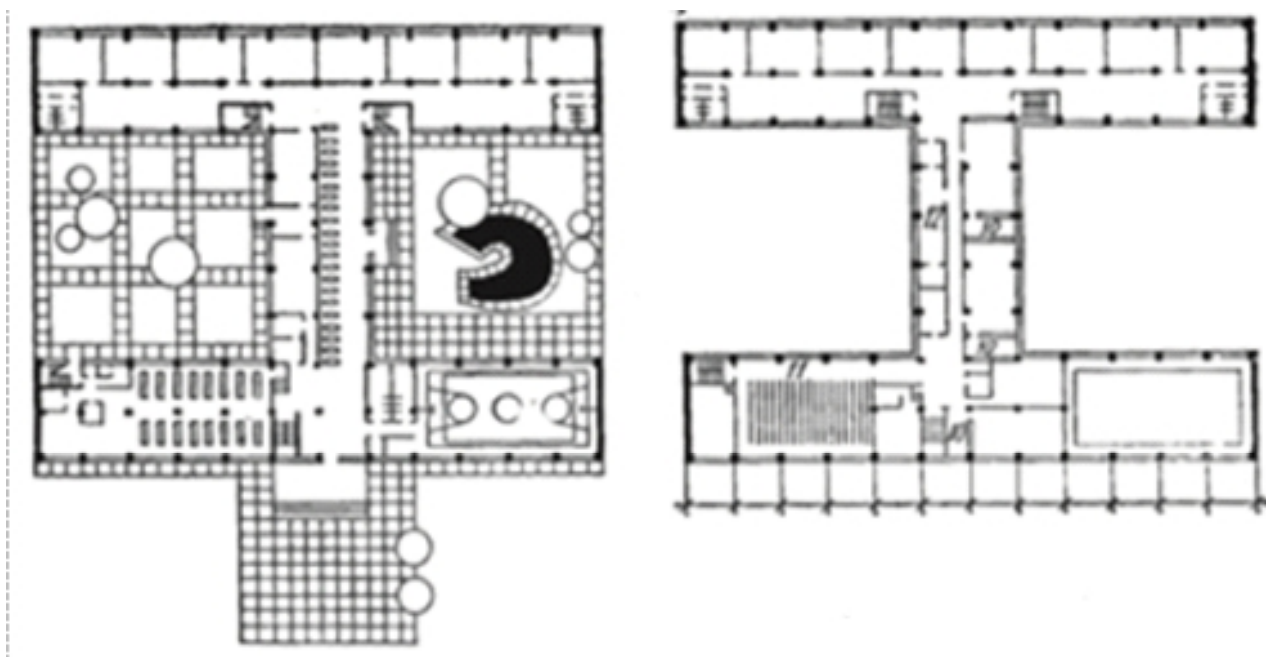


Рис. 1. Школа индустриального общества: архитектурный тип и его скрытая программа – классно-урочная система, ученик как пассивный реципиент знаний, тотальный контроль со стороны педагога, унификация учебных планов и программ ...

уточнить что «не нужно» [4]; здесь ядро программы выражает заказчик. Появление корпоративного клиента и проектного подрядчика (муниципального, государственного, частного) меняет ситуацию в корне. Архитектор крупной проектной фирмы начинает разрабатывать концепции и программы объектов «впрок», опираясь на маркетинговый анализ рынка и «заготавливая на все случаи жизни конгресс-холлы, жилые комплексы, кварталы», а далее – «подыскивая клиента, который слопаёт это и примет как свое» [4].

Ранняя модернистская этика породила тип архитектора, уверенного в своей способности формировать проектные программы без обращения к клиенту – от собственного «первого лица» и с опорой на идеологическую догму; «архитектор нагло полагал, что, он-то и есть носитель правильного человеческого восприятия» [4]. И хотя модернизм и жизнестроительные интенции архитекторов неразделимы, быстро и почти неизбежно наступающее разочарование общества в зданиях, программы которых инспирированы самими архитекторами, в конце концов, вынуждает последних обращаться к другим алгоритмам деятельности.

Появление корпоративного клиента, индустриализация строительства, массификация общества и проектных задач приводят к тому, что с конца 1940-х годов в мировой архитектурной практике (десятилетие-полтора спустя и в отечественной) начинают распространяться научные подходы к разработке архитектурных программ, прежде всего – для объектов так называемого «массового индустриального строительства». В это время архитектура обращается к физиологии, санитарии и гигиене, педагогике, антропометрии и эргономике, общей и средовой психологии, демографии, социологии, статистике. Отчасти на этой основе формируются новые индустриальные прототипы зданий (школ, больниц, вокзалов, жилищ), где архитектура и социальные программы вновь спаяны воедино (рис. 1)³.

Специфика индустриального «проектно-строительного производства» порождает нужду в унифицированных архитектурных объектах, а значит, – и в универсальных программах, которые кодируются с помощью строительных норм и правил, других регламентов. Американский исследователь Л. Попов (Lubomir Popov) из университета Боулинг Грин (Bowling Green State University) говорит о трех исторических формах архитектурного программирования. Самым ранним он называет «программирование,

основанное на прототипах» и использующее их «скрытые программы», воплощающее доиндустриальный культурный контекст, ремесленные подходы и беспомощное, если нужно внести изменения в функциональную структуру деятельности [5]. Индустриальному обществу со второй половины XX века принадлежит «программирование, основанное на архитектурно-строительных типах». Его социальная составляющая также встроена в тип, но менее жёстко, чем доиндустриальная прототипическая. Новые формы жизни и программы могут быть рекомбинированы из старых как из набора исходных элементов. Наконец, третья форма – «программирование, основанное на исследованиях», отражает постиндустриальный культурный контекст, строится на потребностях и структурах деятельности клиента, незаменимо для больших, сложных и новых объектов. Именно об этой форме речь в основном и пойдет далее.

В пятидесятые-шестидесятые годы XX века разворачиваются процессы, которые делают и унифицированную индустриально ориентированную, и упрощенную индивидуализированную стратегии информационного обеспечения проектирования недостаточными.

- Это первые импульсы постиндустриальной трансформации общества и культуры, ее демассификации, под действием которой ещё недавно устойчивые индустриальные типы вместе со своими программами стали обнаруживать признаки устаревания, а новые формы жизни требовали программного воплощения в новых материально-пространственных структурах.

- Это появление компьютеров, породившее надежду на передачу им проектных функций и потребовавшее дешифровки, рационализации проектного процесса и методологии, в том числе – на этапе предпроектного исследования и моделирования.

- Это эрозия старого профессионализма с его патерналистической, диктаторской установкой на право архитектора самостоятельно формулировать и решать архитектурные задачи – в пользу так называемых «моделей соучастия», когда архитектурная программа начинает складываться из реального множества установок членов «проектной команды».

- Это «средовой подход», заставивший усомниться в возможности универсальных программ для их тиражирования на рынке ли, в социальной ли подсистеме, раскрывший необходимость программирования всякий раз по месту и по обстоятельствам.

В США в конце 1950-х – начале 1960-х годов появились архитекторы, которые стали предоставлять услуги по разработке предпроектных программ-заданий [6], а также крупные проектные фирмы, начавшие развивать архитектурное программирование как часть проектного бизнеса⁴. В 1959 году В. Пенья и Б. Каудилл (Bill Caudill) опубликовали одну из первых статей об архитектурном программировании – «Архитектурный анализ – прелюдия к проекту». В 1960-е годы программирование становится широко признанной и распространенной дисциплиной в высших архитектурных школах США. В 1966 году Американский институт архитекторов (AIA) опубликовал свой первый буклет об архитектурном программировании в помощь профессионалам «Новые техники архитектурной практики». В 1973 году Национальный совет комиссий по архитектурному лицензированию США (NCARB) использовал книгу В. Пенья как основу для лицензионного экзамена по разделу «предпроектная стадия». Распространение методов архитектурного программирования сопровождалось развитием самого метода. К 1980-м годам обычная архитектурная программа уже не сводится к перечню и площади помещений, в ней чаще стали формулироваться критерии оценки эффективности проектного решения, требования приватности, территориального контроля, организации потоков людей, информации и услуг [7, p.52].

Тренд 1980-х – начала 1990-х годов, когда школы архитектуры охотнее принимали в штат философа-постмодерниста, чем социолога или психолога, перенос акцентов на архитектурную форму, вопросы ее визуального восприятия, семантики,

Стратегии и результаты архитектурного программирования в трактовке некоторых авторов

Результаты	Стратегии действий программиста		
	идентификация	оценка	принятие решений
информация о потребностях	<i>Bugni, 02</i>	<i>Preiser, 05</i>	<i>Duerk, 93</i>
задачи (предназначение зданий и их частей)	<i>Zeisel, 06</i>	<i>Hershberger, 99</i>	
проблемы (для принятия проектных решений)	<i>Lang, 87</i>	<i>Pena, 12</i>	<i>Cherry, 99</i>

деконструкционистские, постструктуралистские искания, привел к тому, что «архитектурное программирование» стало исчезать из учебных планов как дисциплина. В 2000-х годах постепенно и нелинейно, но интерес к социальной проблематике в целом и архитектурному программированию в частности возрождается и в профессии, и в образовании. Сегодня в США этап архитектурного программирования внедрён в стандартный договор на архитектурные услуги и в структуру национального архитектурного лицензионного экзамена [8]. По мнению Билла Хельмута, президента фирмы НОК, роль архитектурного программирования возрастает, в частности, по мере распространения технологий информационно-строительного моделирования (BIM – Building Information Modelling) [9]. Наконец, программирование для проектирования и строительства обрело статус нормативной процедуры в стандарте ISO 9699 [10, p.17].

Современная трактовка архитектурного программирования

Ни один автор, пишущий об архитектурном программировании, не обходится без определения этого феномена. Отдельные исследователи темы, не противореча друг другу, все же явно смещают акценты на разные аспекты содержания и стратегии действий «программистов». Так, Д. Зайсел называет программированием «формулировку того, для чего здание (или другой объект) предназначено» [11, p.51]. Э. Черри пишет: «Архитектурное программирование есть процесс исследований и принятия решений, который определяет проблему для проектного решения» [8, p.3]. Р. Хершбергер дает феномену такое определение: «Архитектурное программирование – первая стадия процесса архитектурного проектирования, на которой идентифицируются относящиеся к делу ценности клиента, пользователя, архитектора и общества; устанавливаются важные цели проектирования, вскрываются факты о проекте, а потребности в услугах становятся явными» [6, p.5]. Д. Дюрк подчеркивает менеджериальный аспект, по ее мнению, «архитектурное программирование есть процесс управления информацией с тем, чтобы на каждой стадии проектного процесса была доступна информация, требуемая в тот момент времени, и можно было бы принять наилучшее из возможных решений по формированию строительного проекта» [12, p. 8]. В. Прайзер делает акцент на аксиологическом содержании программы, он пишет: «Программирование есть процесс систематического сбора, документирования и обмена критериями оценки ожидаемых эксплуатационных свойств зданий» [10, p. 21].

Рассматривая эти и другие представления, можно заметить, что все они формируются в некой двухмерной матрице смыслов, задаваемой, во-первых, вариантами трактовки

основного содержания, результатов архитектурной программы как документа, а во-вторых, разными представлениями о роли и стратегиях действий «программиста» по отношению к упомянутому содержанию (табл. 1).

Ключевым контентом программы может быть признана социальная информация – о потребностях, ценностях, целях, ресурсах, регламентах... – применительно к проектируемому объекту. Эта информация бывает осмыслена социально-архитектурно как «задача» проектирования, «предназначение» или «функция» здания, а случается – и еще более глубоко: как проблема, требующая проектного решения и воплощения⁵. В одних случаях «информация», «задача» (предназначение) и «проблема» могут просто идентифицироваться, в других – оцениваться с точки зрения приоритетов и конфликтов интересов, а в третьих – служить основой принятия архитектурных и социальных решений. Очевидно, отдельные аспекты содержания архитектурных программ и стратегии действий, на которые они нацеливают, неравноценны. Наиболее целостной является стратегия «управления решением проблем», а ее элементарной составляющей – идентификация, сбор исходной социальной информации. Очевидно также, что в представленной матрице каждая следующая содержательная ступень (сверху вниз) и каждая следующая стратегия (слева направо) включают в себя предшествующие: принятие решений требует идентификации исходных данных и их оценки; проектная проблема осознается как внутренние противоречия между постановкой задач и собранной исходной информацией (как конфликт потребностей, потребностей и ресурсов, потребностей и регламентов ...).

Уровни и модели разработки архитектурных программ

Архитектурные программы могут отличаться уровнем проработки, детализации требований к будущим проектным решениям, уровнем сложности для объектов, разных по функциональной комплексности, а также процедурами разработки – по источникам исходной информации и характеру связи с проектным процессом.

На предварительных стадиях работы программа может представлять не более чем «декларацию общего видения» будущего архитектурного объекта, на продвинутых – «детальное задание», а на поздних – подробное и тщательно выверенное «специальное и операционное задание» [13, р. 39].

Д. Зайсел различает «количественные» и «качественные» программы. Первые «уточняют проектные цели и требования на языке необходимой общей площади, минимальных параметров помещений <...>, типов пространств <...>, максимальных пределов стоимости и минимального отношения площади оконных проемов и площади помещений» [11, р. 51]. Такие программы излишне регламентируют работу архитектора и оставляют за скобками один из центральных вопросов – критерии оценки эффективности проектного решения. «Качественные» программы, в первую очередь, уточняют, каким условиям должно отвечать проектное решение, каковы его ключевые эксплуатационные свойства, «нацелены на то, чтобы выявить нужды потребителя, а не на сужение поля выбора проектировщиков» [11, р. 51].

Четыре уровня архитектурных программ по сложности в зависимости от разрабатываемого объекта предлагает различать В. Пенья. Самые простые программы реализуются в сфере традиционных архитектурных услуг по проектированию, скажем, индивидуального жилого дома, когда задания клиента оказывается достаточно, чтобы архитектор мог сформулировать проектную проблему. Для разработки программ второго уровня сложности компетенции клиента как выразителя проектного задания оказывается недостаточно, требуется консультационная помощь подготовленного архитектора, как это происходит при проектировании образовательных, медицинских, офисных зданий. Еще более сложные программные услуги оказываются востребованы при проектировании многофункциональных и многостадийных зданий, использующих «системные строительные

Модель содержания программирования, основанная на учете ценностей (по: Hershberger, 1999)

Ценности	Элементы программы			
	цели	факты	потребности	идеи
человеческие	функциональные, социальные, психологические, физиологические			
средовые	участок, климат, контекст, ресурсы, отходы			
культурные	исторические, институциональные, политические, правовые			
технологические	материалы, инженерные системы, технологические процессы			
временные	рост, изменения, преемственность			
экономические	финансы, строительство, эксплуатация, содержание, энергозатраты			
эстетические	формы, пространства, цвет и смыслы (значения)			
безопасности	конструктивная, пожарная, химическая, личная и криминальная			
другие	...			

технологии», сложное компьютерное моделирование. Но самой высокой сложностью обладают программы в градопланировании с активным использованием компьютерного моделирования, с привлечением к разработке программ различных специалистов, включая социологов, экспертов по информационным коммуникациям; программирование становится здесь постоянным, циклическим процессом [9, p.7–8].

По Р. Хершбергеру, в проектной практике со временем сложились четыре подхода к программированию или четыре модели организации программирования. «Модель, основанная на проектировании», традиционна для архитектурной практики. Она подразумевает, что программирование и проектирование сопутствуют друг другу, цели и задачи проектирования уточняются на всем его протяжении, в ходе постоянного взаимодействия архитектора с клиентом. «Модель, основанная на знании», складывается в 1960-е годы как реакция на несоответствие большого количества архитектурных решений потребностям людей, т. е. в результате неудовлетворенности традиционной моделью. Особенно это касается функционально сложных объектов – больниц, тюрем, офисных комплексов... Здесь архитектор прибегает к помощи социальных наук, поставляющих достоверные знания о потребностях людей и сферах человеческой практики. «Модель, основанная на соглашении», также касается больших и сложных объектов, подразумевает активное соучаствующее формирование программы архитектором и всеми заинтересованными лицами (клиент, инвестор, местное сообщество...) как выявление всего спектра потребностей, нахождение компромиссов для их удовлетворения. Сравнивая данные три модели, Р. Хершбергер приходит к заключению, что каждая имеет свои сильные стороны и все они объединяются вокруг тех факторов или ценностей, которые формируют программу. На этом основании он выдвигает четвертую модель программирования – «модель, основанную на ценностях», объединяющую преимущества трех остальных и придающую единую информационную структуру программированию при разных алгоритмах его осуществления (табл. 2).

Универсальной основой программирования Хершбергер считает рассмотрение

целей, фактических данных, потребностей и созидательных идей в отношении ключевых «ценностей» – человеческих, средовых, культурных, технологических, временных, экономических, эстетических, ценностей безопасности и др. [6, p.731].

Архитектор как программист

Даже в традиционной модели архитектурного программирования роль архитектора как основного разработчика не очевидна и требует пояснения. По другим причинам, но комментарии необходимы к моделям программирования, акцентирующим большие объемы научных исследований или посреднических коммуникативных операций. А должен ли и в состоянии ли архитектор делать это?

Формально и по закону выдача задания на проектирование – архитектурной программы – обязанность заказчика проекта⁶. Но архитектор почти никогда, за редким исключением, не получает от клиента задания, с которым можно было бы приступить к проектированию. Детально исследовавший эту ситуацию Б. Лоусон утверждает: «Конечно, это заблуждение полагать, что клиент просто выдает проектировщику исчерпывающее задание, в котором проблема полностью определена, а ограничения ясно обозначены» [14, p. 183]. Причины понятны: заказчик не всегда осознает, что именно ему нужно, не представляет пространственных последствий своих желаний, не может изложить потребности технически грамотно. Заказчиком нередко выступает сложный коллектив, члены которого имеют противоречивые и взаимоисключающие потребности, сами по себе не складывающиеся в общую картину проектного задания. Такого рода проблемы возникают даже при проектировании индивидуального жилого дома для конкретной семьи, что уж говорить о крупных многофункциональных комплексах для акционерных заказчиков.

Некоторые профессиональные застройщики, заказчики на проектирование хорошо знакомых им объектов, приобретают необходимый опыт разработки концепций и программ-заданий и содержат специальный штат для их составления. Но они не обязательно являются конечными владельцами и пользователями зданий, а их программы могут тиражировать существенные ошибки. Если ни заказчик, ни архитектор не в состоянии разрабатывать проектные программы, это становится самостоятельным бизнесом, за который берутся специальные рыночные аналитики, проектные менеджеры, консалтинговые компании, поведенческие и средовые ученые, специально создаваемые ad hoc временные исследовательские группы. В результате архитектурные фирмы «ежегодно теряют миллионы долларов на том, что услуги по консультированию в области программ-заданий предоставляют другие» [3]. Но наиболее дальновидные проектные организации занимаются этим сами⁷.

Понятно, что если некий архитектор и принимается за программирование сложных объектов, то это не любой архитектор, а подготовленный, «специализированный на том, чтобы задавать нужные вопросы в нужное время, кто в состоянии отличать желания от потребностей и обладает умением классифицировать. Программисты должны быть объективны (до некоторой степени) и аналитичны, быть накоротке с абстракциями, в состоянии оценивать информацию и идентифицировать важные факторы, оставляя в стороне второстепенный материал. Проектировщики, – уточняет Пенья, – не всегда могут делать это. Обычно они субъективны, интуитивны и поверхностны в обращении с физическими концепциями» [9].

Одна из ключевых задач архитектора как программиста – составить по согласованию с заказчиком список всех тех физических и юридических лиц, кто должен стать источником предпроектной информации, включая конечных пользователей здания, инвесторов, представителей служб эксплуатации, экспертов, членов местных сообществ и т. п. Всех их называют «программным комитетом» или «программной командой». Архитектор



Рис. 2. "Последовательная" модель связи программирования и проектирования

выступает как организатор работы этого комитета, ответственный перед заказчиком. Архитектор-программист должен владеть методами сбора социальной информации: опросами (интервьюированием и анкетированием), наблюдениями и анализом документов. Есть и другие – социально-средовые методы⁸.

Учитывая сказанное, не приходится удивляться, что профессиональные организации, включая Международный союз архитекторов (UIA), Американский институт архитекторов (AIA), Королевский институт британских архитекторов (RIBA) единодушно вменяют в обязанность архитектора «понимание методов исследования и подготовки заданий на проектирование», умение «составлять задания на проектирование, формулируя потребности общества, пользователей и заказчиков, а также проводить исследования и определять, в зависимости от контекста, функциональные требования к различным типам искусственной среды» [15, с. 3;16, р. 8]⁹.

Архитектурное проектирование и программирование: логика взаимосвязи

Существуют вполне логичные, запечатленные в соответствующих схемах и моделях, представления о том, что программирование должно предшествовать проектированию, что, прежде, чем начать проектировать, архитектору лучше получить подпись клиента под заданием, а иначе – не избежать лишней и неоплачиваемой работы по переделке проекта из-за вновь возникающих желаний и капризов заказчика. На стороне этой логики и упомянутое выше отличие аналитической и рациональной процедуры программирования от синтетического и иррационального действия проектирования (рис. 2). «Если программирование – поиск проблем, то проектирование – решение проблем» [9], в общем, это по сути своей разные умственные процедуры.

Но доминирующая проектная практика показывает, что последовательная модель действий – это редкость. «Формально брифинг или программирование, – полагает группа исследователей из проектной фирмы Alexi Marmot Associates (Великобритания), – помещают после планирования и перед проектированием. В реальности же некоторая его часть реализуется еще на стадии планирования, а значительная доля детального программирования осуществляется на стадии проектирования; отдельные элементы специального программирования реализуют даже после начала строительства» [13, р. 40]¹⁰. С ними согласен исследователь методики проектирования Б. Лоусон, утверждающий, что «проектировщики не разносят деятельность анализа и синтеза по отдельным стадиям <...>

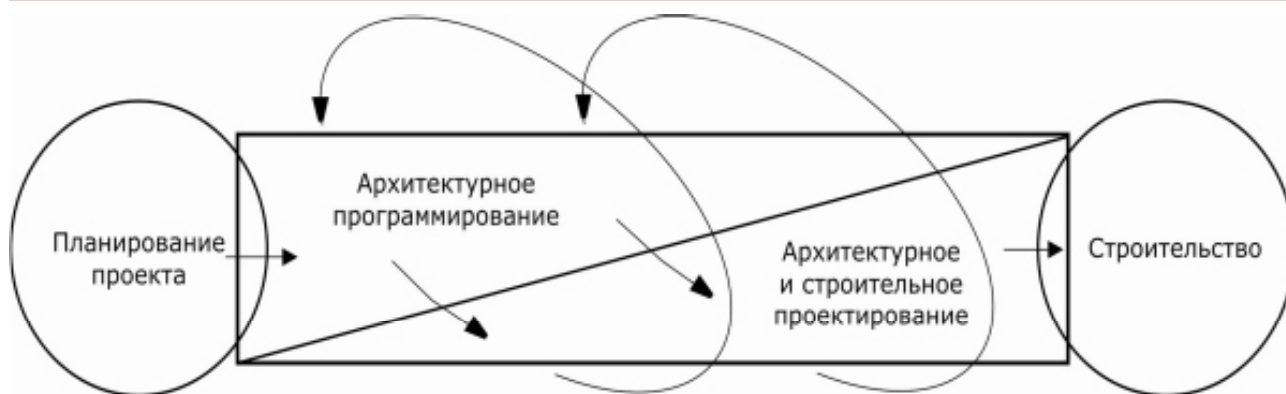


Рис. 3. "Параллельно-циклическая" модель связи программирования и проектирования

даже разработка задания не представляет собой изолированный этап, но есть деятельность, сопутствующая всему процессу проектирования» [14, р. 157]. Разделяет эту точку зрения и Р. Хершбергер: «Наиболее часто используемый сегодня метод программирования реализуется одновременно с проектным процессом» [6, р. 7]. В общем, логически «неправильная» параллельная модель взаимосвязи программирования и проектирования более реалистична. Если учесть, что от проектирования нередко приходится возвращаться к предшествующим этапам программирования, чтобы внести корректировки, то модель приобретает характер «параллельно-циклической» (рис. 3).

Структура и содержание процесса программирования

Что представляет собой архитектурное программирование как исследовательская и управленческая процедура, когда речь идёт о больших и сложных и/или проектируемых впервые объектах? Два американских автора дают подробные ответы на этот вопрос – Вильям Пенья и Эдит Черри [1,8,9]. Первый говорит о «пяти шагах», а вторая – о «шестиэтапном процессе» архитектурного программирования. Первый выдвинул и описал свою модель, опирающуюся на примерно двадцатилетний опыт архитектурных исследований, ещё в 1969 году, а вторая изложила свои построения сорок лет спустя. Сравнительный анализ показывает: две эти модели настолько совпадают, что вполне можно говорить о едином и проверенном шестидесятилетним опытом профессиональной практики алгоритме архитектурного программирования. Учитывая, что к пяти шагам В. Пенья сам добавляет шестой – «сбор предварительной информации», оба автора согласны и в отношении общего устройства программной процедуры. По мнению Пенья, «последовательность шагов может меняться, но сами шаги образуют упорядоченный каркас для классификации и документирования информации, которая приходит из многих источников» [9, р. 13]. Рассмотрим процедуру программирования (рис. 4).

I. Этап «сбора предварительной информации» (Пенья) / «исследования архитектурно-строительного типа» (Черри). Оба автора одинаково трактуют содержание этапа – как начальное знакомство с тем классом зданий, представитель которого станет объектом проектирования – школами, многоквартирными жилыми домами, торговыми центрами, больничными учреждениями, аэропортами и т.п. Подчеркивается особое значение этапа для тех, кому не приходилось ранее иметь дела с подобными рода объектами. Исследование охватывает и теоретические источники по теме, и проекты-аналоги (англ. precedents), и проектные нормы, документы, сметы, возможно также обращение за информацией к консультантам, экспертам. Предметом исследования становятся следующие стороны архитектурных решений:

- пространства и помещения, характерные для зданий данного типа и типичные



Рис. 4. Шестиэтапная модель процедуры архитектурного программирования Пеня – Черри

взаимосвязи пространств с данными функциями;

- пространственные критерии проектирования – удельная площадь на человека или другую единицу («машиноместо», «койкоместо», посадочное место и т.п.) и характерные для данных зданий отношения «чистой» площади к «общей»;
- характерная стоимость единицы площади;
- характерные требования к участку;
- региональные особенности, способные повлиять на применимость описанных выше типичных характеристик к конкретным местным условиям;
- требования, специфические для зданий данного типа – механические, электрические, безопасности и т.п.

Пеня подчеркивает, что в проектных аналогах важно рассматривать не сами готовые решения, а ту информацию, которая предшествовала их принятию.

II. Этап «выявления целей и задач». Исследователь устанавливает цели клиента-заказчика в отношении четырех аспектов будущего архитектурного решения: функции (виды деятельности, назначение здания, количество и категории людей – пользователей здания...); формы (желательное эстетическое и психологическое воздействие здания, имидж здания, отношение к окружению...); экономики (бюджет проекта, ожидаемый баланс первоначальных и эксплуатационных затрат, баланс качества/ стоимости, установки по отношению к сбережению ресурсов, «устойчивости»...) и учета фактора времени (тайминг проекта, ожидаемые перемены в требованиях к зданию на ближайшую, среднюю и далекую перспективу, факторы изменения и роста, которые повлияют на функции, форму,

экономичность...)¹¹.

III. Этап «сбора и анализа фактов / необходимой информации»¹². Факты Пенья предлагает структурировать по тем же четырем основаниям (функция – форма – экономика – время). Под фактами понимаются сведения о реальной организации жизни пользователей здания – кто и что, с кем, как часто и для чего делает. Выясняются сведения о необходимом оборудовании, о «пространственных критериях» (тех же удельных показателях площади на человека или другую единицу, ранее рассмотренных в отношении проектов-аналогов), о требованиях к освещенности, акустике, доступности для инвалидов, охране и безопасности. Анализируется участок в отношении правового статуса, зонирования, ограничений и регламентов на использование, существующего движения, обслуживающей инфраструктуры, наличия сетей, топографии, особенностей освоения и застройки, климата, растительности, животного мира и других природных качеств, видовых точек «из» и «на» участок. Рассматриваются факторы, влияющие на стоимость, включая требования энергоэффективности. Исследуется временная динамика процессов, для которых проектируется здание. Если в распоряжении клиента на момент начала проектирования есть какое-то здание, используемое для аналогичных целей, то исследуется и оно. Имеющиеся повторы в содержании второго и третьего этапов можно, по-видимому, трактовать как в одном случае – целевые установки владельца и заказчика в отношении четырёх параметров, а в другом – реальные характеристики процессов жизнедеятельности и пользователей.

IV. Этап «раскрытия и анализа концепций»/«идентификации стратегий». Исследователь выясняет предпочитаемые заказчиком пути, методы достижения выдвинутых ранее целей с учетом обнаруженных ресурсов, возможностей и ограничений. Чтобы эти пути не стали путями на ногах проектировщика, их стараются сформулировать в самом общем, принципиальном виде, языком организационно-структурных подходов и моделей, «провоцирующих слов» и «навязчивых концепций», обладающих максимальной степенью всеобщности. Черри и Пенья приводят следующие примеры возможных концептуальных установок заказчика, выявляемых, понятное дело, с помощью архитектора-исследователя: «централизация» vs «децентрализация», «интеграция» vs «дробление», «изменчивость» vs «многофункциональность», «гибкость», «потоки», «приоритеты» и «периодизации», «уровни допуска» (кому куда разрешено попадать).

V. Этап «определения потребностей»/«количественных требований». Рассматривается связь требуемого состава помещений определенной площади с имеющимся бюджетом и сроками проектирования/строительства. Сначала конкретизируются требования к каждому помещению проектируемого здания, на основе «пространственных критериев» подсчитываются и суммируются площади – чистая, коммуникаций и общая; затем подсчитываются коэффициенты планировочной эффективности и с использованием удельной стоимости аналогов определяется стоимость строительства. К стоимости строительства добавляют расходы на геологические изыскания, разработку и подготовку участка, проведение строительного тендера, непредвиденные расходы, стоимость оборудования и мебели, гонорар архитектору (всё это – на основе аналогов и укрупнённых расчётов) и т.п. Полученная цифра сравнивается с бюджетом проекта и, либо вносятся коррективы в расходы, либо меняется бюджет.

VI. Этап «формулировки проблем»/«составления программы». Э. Черри описывает содержание этапа лаконично: «Вся информация документируется, программа одобряется заказчиком и интегрируется в проектный процесс» [9]. В. Пенья напоминает, что формулировка проблемы как главное содержание архитектурной программы должна структурироваться по тем же четырем основаниям (функция, форма...), быть выражена качественным языком и воплощать уникальность и сущность данного проекта» [9, р. 26]. Один из приводимых им примеров изложения

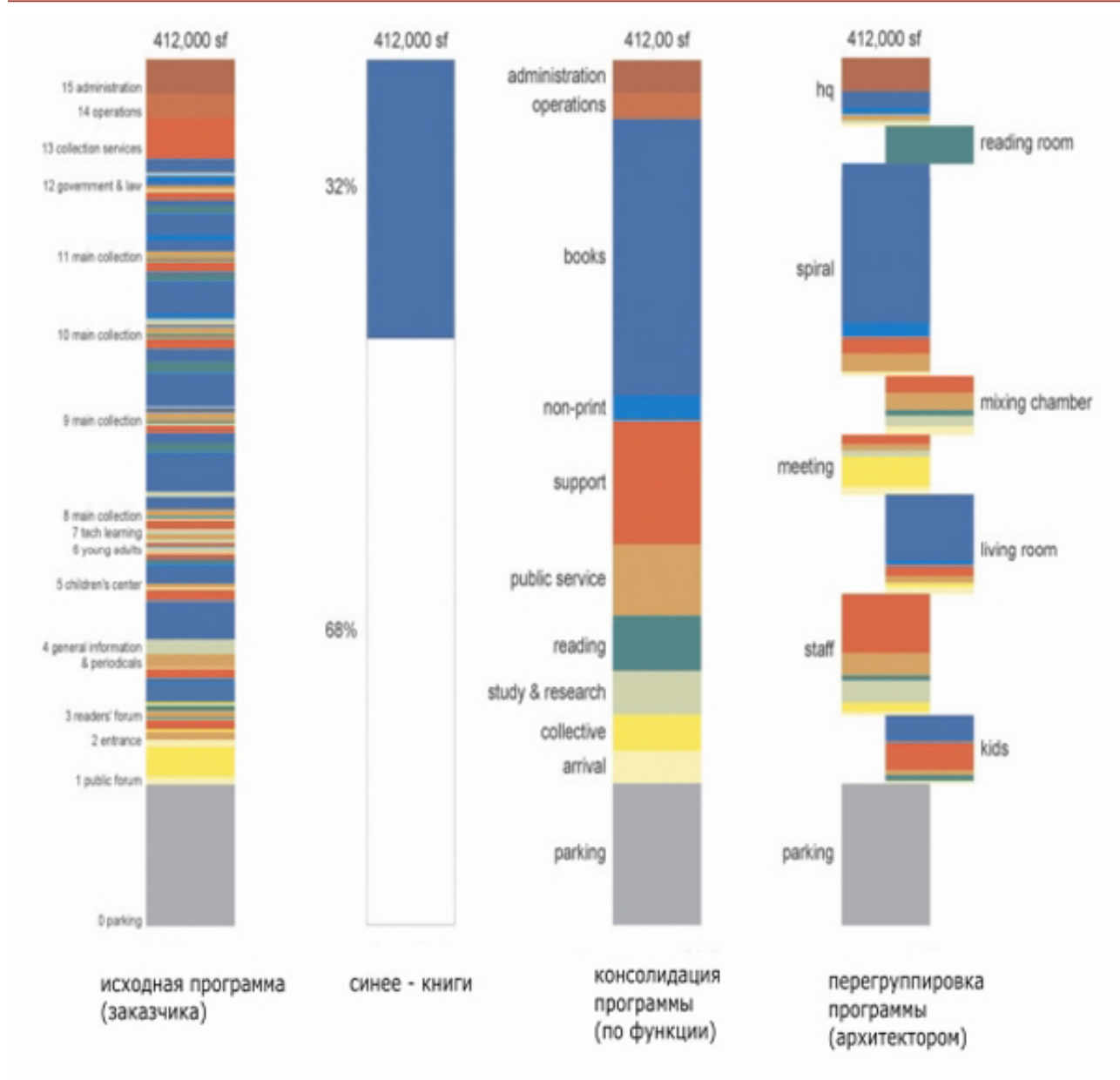


Рис. 5. Архитектурная программа Центральной библиотеки Сиэттла (США) в виде масштабированной столбчатой диаграммы. Арх. Р. Коолхаас и др. Источник: <https://pinterest.com/ayeshagaunt/architecture-drawings/>

функциональной проблемы проектирования театра звучит так: «Для некоторых людей антракты так же важны, как и сам спектакль. Поэтому здание должно удовлетворять стремлению индивидуального пребывания в составе больших групп» [9, p.27]. Другой пример иллюстрирует формулировку проблемы в аспекте «времени»: «Кампус должен расти. Должна сохраняться визуальная связанность (его частей. – К.К.) на всех этапах развития» (9, p. 27). Как видим, эти констатации не обязательно отличаются сложностью и несут характер «откровения», но должны содержать ключевые установки членов программного комитета и, в первую очередь, заказчика.

Архитектурное программирование: презентация результатов

Объем, содержание и формы регистрации и представления результатов архитектурного программирования варьируются в широких пределах: от нескольких и даже одной страницы «квартирографии» до сотен страниц документа, скрупулезно придерживающегося шестиэтапной структуры, описанной выше. Часто отличие первого от второго – это

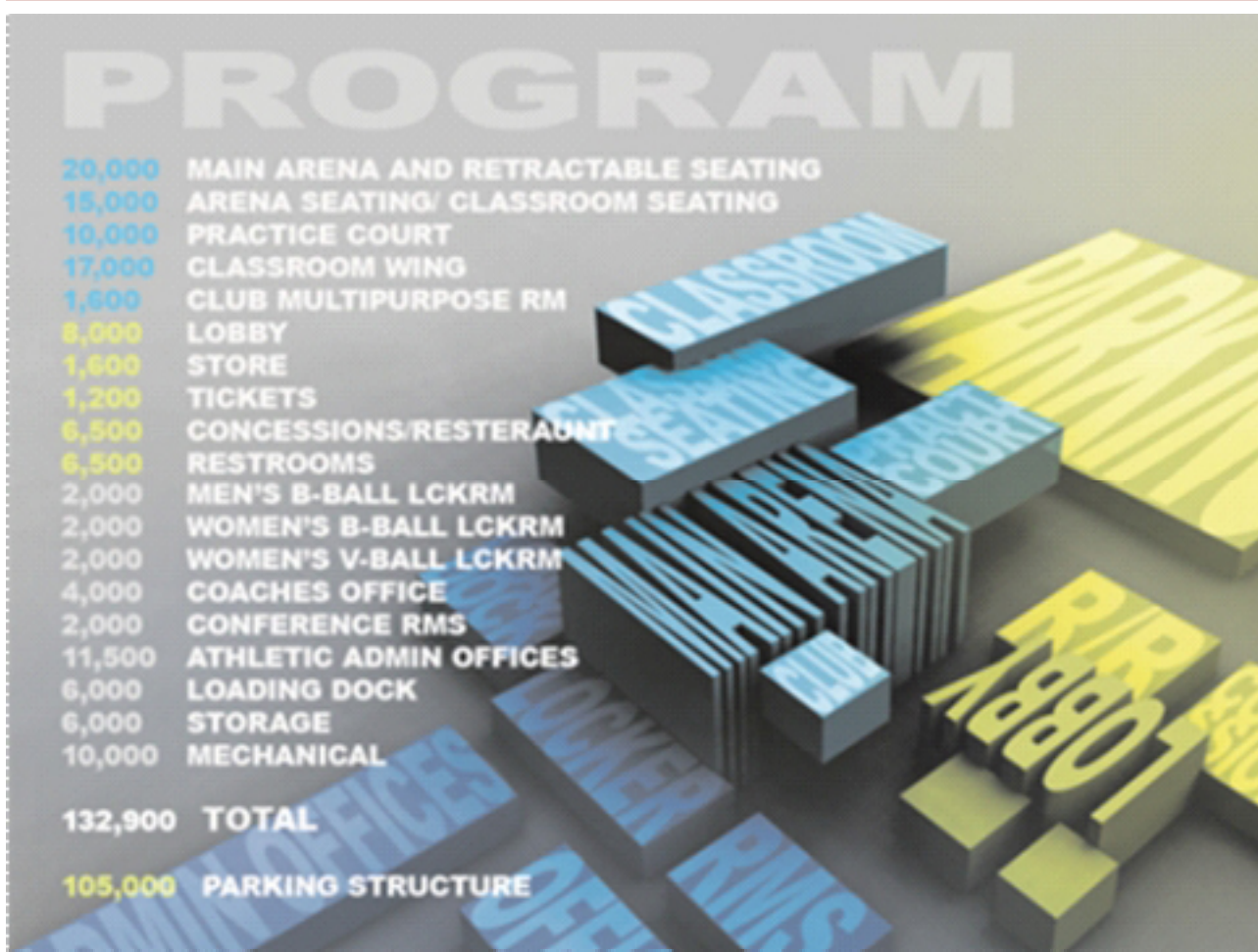


Рис. 6. Архитектурная программа в виде масштабированной 3D-структуры. Авт. Alex Hogrefe.
 Источник: https://vizualizing_architecture.com/3d-program-diagram/

расстояние между программой, представленной клиентом, и программой, разработанной архитектором или другим программистом на этой основе.

Умеренно детализированная программа содержит несколько десятков страниц текста, аналитических матриц и графической информации. Во введении описывают цели и задачи исследования, истоки возникновения проектной задачи (инициатор, заказчик, предыстория темы), состав программного комитета (команды), источники и методы получения предпроектной информации. Основная часть текста представляет результаты анализа проектов-аналогов и теории проектирования, опроса членов программного комитета (часто – по структуре матрицы Р. Хершбергера, см. рис. 3), изучения проектных регламентов и норм, обследования градостроительного контекста, участка и климата. В заключении приводятся систематизированные качественные и количественные требования к будущему объекту, его устройству, размещению, функционированию и эволюции. Излюбленная форма аналитической презентации информации в архитектурных программах – матрицы: состава, площади и взаимосвязи помещений, удельных показателей плотности, стоимости, баланса территорий, дополнительных требований к планировке и оборудованию...

Ключевое содержание программы – функциональное, не случайно широко используемое в отечественной традиции понятие «функциональной программы» есть близкий синоним «архитектурной». Полезное и привычное архитектору выражение этого содержания – графическое. Хорошая архитектурная программа изобилует графическими изображениями. Выводная графика – социально-функциональные композиционно-структурные модели. Анализ большого их числа позволяет автору

говорить о трех основных формах графических программных моделей (англ. architectural programming diagrams). Масштабированные столбчатые диаграммы показывают спектр и сравнительную развитость основных функций. Использованные Р. Коолхасом и ОМА при проектировании Центральной публичной библиотеки в Сиэтле, эти диаграммы стали классической иллюстрацией по теме программирования (рис. 5)¹³. Немасштабированные и масштабированные пузырьковые диаграммы (англ. bubble diagrams) показывают состав и связи зон и помещений. Это хорошо знакомые любому архитектору «функциональные схемы», отличающиеся только тем, пропорциональны ли размеры каждого «пузыря» площади соответствующей зоны или помещения. Трехмерные масштабированные диаграммы занимают промежуточное положение между условными схемами и объёмно-пространственными решениями. Сегодня они популярны, но не все знают, какая большая аналитическая работа предшествует их появлению на свет (рис.6)¹⁴.

Заключение

Эпоха стабильных «прототипов» и «архитектурно-строительных типов» вместе с интегрированными в них социально-архитектурными программами уходит в прошлое, а необходимость развивать «архитектурное программирование» для возникающих форм жизни как самостоятельную компетенцию еще не вполне осознана российской архитектурной профессией и архитектурным образованием. Более чем шестидесятилетний опыт развития данной дисциплины за рубежом требует осмысления, использования и развития в отечественной науке и практике.

«Архитектурное программирование» есть сложная интеллектуальная и творческая деятельность, называть которую «составлением технического задания» – значит примитивизировать и исказить ее смысл, вводить в заблуждение относительно целей, алгоритмов и результатов. Использование международно признанного понятия представляется гораздо более обоснованным.

Не существует единых представлений о содержании и необходимой детализации архитектурных программ. В зависимости от стадии разработки (предпроектная или проектная), от сложности объекта, от модели отношений между заказчиком и архитектором, речь может идти об очень разных документах и предшествующих им исследовательских процедурах.

Программированием занимаются и архитекторы, и другие специалисты. Взваливая на себя эти сложные и требующие специальной подготовки обязанности, архитектор повышает степень влияния на конечный архитектурный продукт свою роль как координатора и лидера в программной и проектной командах, собственную значимость как профессионала и общественной фигуры. Отказываясь от программирования, он во всем этом теряет. Избавиться от хлопот по архитектурному программированию не просто неразумно, это еще и нереально, поскольку оно тесно интегрировано в проектный процесс на всем его протяжении.

Структура и содержание архитектурного программирования как исследовательской и управленческой процедуры подробно описывает шестиэтапная модель Пеньи-Черри. Ее и достоинством, и недостатком является рациональность и последовательность действий. Поскольку модель строится как предшествование программирования проектированию, то для параллельно-циклической схемы взаимосвязи потребуется ее корректировка самим исследователем.

Многолетней практикой выработан и активно развивается язык архитектурного программирования – вербальный и графический, помогающий регистрировать, анализировать и интерпретировать результаты исследований в удобной для архитектора форме и представлять их для одобрения заказчику.

Примечания

¹Помимо рассматриваемых в тексте и цитируемых публикаций по теме АП есть и другие, – упомянутых выше и неназванных авторов, – среди них: Kumlin, R. *Architectural Programming: creative techniques for design professionals/ Robert R. Kumlin.* – New York: McGraw-Hill, 1995; Cherry, E. *Programming for Design: from theory to practice/ Edith Cherry.* – New York, John Wiley & Sons Inc., 1999; Preiser, W. *Professional Practice in Facility Programming/ W.F.E. Preiser (ed.).* – New York, N.Y.: Van Nostrand Reinhold, 1993; Sanoff, H. *Methods of Architectural Programming/ Henry Sanoff.* – Stroudsburg, Pa: Dowden, Hutchinson & Ross, 1977; Sanoff, H., *Integrating Programming, Evaluation and Participation in Design: A Theory Approach.* – Brooffield, VT: Avebury, 1992; White, E.T. *Project Programming, A Growing Architectural Service/ E.T. White.* – Tucson, Arizona: Architectural Media Ltd., 1991.

²Смысл, вкладываемый в понятие «прототип» его автором, можно понять из следующего фрагмента: «Для того чтобы обеспечить процесс проектирования, необходимо иметь взаимосвязанную совокупность, систему различных средств и представлений объекта, фиксирующих его строение, конструкции, функции. Необходимо иметь методы и способы расчета отдельных частей сооружений, знания о технологии строительства и многое другое. Мы называем совокупность знаний и средств такого рода прототипом объекта только в том случае, если она позволяет проектировщику с его способностями и навыками вести проектирование» [2].

³На сохраняющихся островках индивидуального проектирования – другая жизнь, здесь архитектор привычно работает, следуя упрощенным кратким заданиям клиента с перечнями помещений и их размерами (отсюда англ. brief – краткая проектная инструкция).

⁴Среди самых известных – фирма Caudill, Rowlett & Scott Architects (CRS) из Хьюстона (США), основанная в 1946 г. Работавший в ней Вильям Пенья именно здесь создал методику и приобрел опыт архитектурного программирования, нашедшие отражение в его знаменитой книге (Рейна, 1969 ...2012). Фирма стала известной благодаря новой технике архитектурного программирования, опробованной при проектировании школы в Blackwell, Оклахома. Названный «техникой сквоттеров», метод заключался в организации на территории реконструируемой школы временного офиса для контактов архитекторов с педагогами, учителями, учащимися и родителями при разработке и реализации задания. Этот метод впоследствии многократно использовался при проектировании других школ, больниц, жилищ CRS и другими компаниями. Со временем при фирме был создан специальный отдел по «управленческому консалтингу», занимавшийся архитектурными программами. В 1994 году ее архитектурная группа перешла в другую знаменитую корпорацию – НОК (Hellmuth, Obata + Kassabaum). В 2015 г. НОК признана крупнейшей архитектурно-инжиниринговой компанией США, она сохраняет лидерство в «устойчивом проектировании», «высокопроизводительном проектировании», «проектировании, основанном на фактах» (англ. evidence based design), архитектурном программировании.

⁵То, что «проектная задача» и «проектная проблема» - не синонимы, есть тема важная и заслуживающая отдельного разговора. Здесь же ограничимся констатацией: задача в архитектурном проектировании формулируется обычно языком «предназначения» или «функции» здания, например, «спроектировать общеобразовательную школу на столько-то учащихся на таком-то участке». А само проектирование начинается в тот момент, когда задача осмысливается как проблема, т. е. невозможность решить задачу из-за некоего внутреннего противоречия или конфликта (например, школьный стадион «не вписывается» в участок и будет размещен на плоской кровле здания). Именно поэтому большинство исследователей методологии архитектурного проектирования рассматривают «архитектурные решения», как «решения проблем». В частности, Р. Хершбергер полагает: «Именно природа проблемы, в том виде как ее выражает архитектурная программа, имеет наиболее фундаментальное влияние на проектное решение в архитектуре» [6, р. 3].

⁶Такова ситуация и в России, и в других странах. По Градостроительному кодексу РФ «Подготовка проектной документации осуществляется на основании задания застройщика или технического заказчика» (ст.48, п.11). В США, согласно стандартным договорам на проектные услуги Американского института архитекторов, «программирование есть ответственность владельца (будущего здания. – К.К.)» [1].

⁷Помимо упомянутых ранее гигантов CRS и НОК, в США есть небольшие фирмы, которые специализируются на архитектурном программировании. Одна из них – старейшая компания Архитектурные исследователи-консультанты (Architectural Research Consultants. – Альбукерке, Нью-Мексико) основана в 1976 году. Директором по исследовательской работе в ней является профессор архитектуры Вольфганг Прайзер – крупнейший в мире эксперт по архитектурному программированию и «оценке зданий после заселения».

⁸Подробнее о социально-средовых методах исследования см.: Кияненко К.В. Общество, среда, архитектура: социальные основы архитектурного формирования жилой среды: учеб. пособие/ К.В. Кияненко; Вологод. гос. ун-т. – Изд. 2-е, перераб. и доп. – Вологда: ВоГУ, 2015. – 284 с.

⁹Американскому институту архитекторов удалось выстроить стройную систему, в которой профессиональное требование «уметь программировать» внедрено в национальный стандарт архитектурного профессионализма, в состав лицензионного экзамена, далее – в положение об аккредитации высших школ архитектуры и их учебные планы. В России такая компетенция записана в Федеральном государственном образовательном стандарте бакалавров по направлению «Архитектура», но пока это требование не подкреплено стандартом профессионализма (на момент написания статьи он только разрабатывается) и требованиями к лицензированию архитекторов как физических лиц (пока не существующему).

¹⁰Известна, например, история проектирования третьего международного терминала аэропорта Пекина к XXIX Олимпийским Играм 2008 года (арх. Н. Фостер). Уже после возведения наземных конструкций здания в программу проектирования были внесены радикальные изменения: по требованию МОК пришлось внедрить в подземную часть этого гигантского сооружения два этажа парковки, которая изначально предусматривалась наземной. И изменения были реализованы.

¹¹Э. Черри добавляет к этому перечню организационные цели заказчика – повысить комфорт деятельности, улучшить общественный имидж компании, рекламировать свою экологическую устремленность, улучшить взаимодействие сотрудников и т. п.

¹²На наш взгляд, этапы второй и третий довольно смутно разграничены по содержанию. Не так-то легко разделить, скажем, информацию о функциях-целях (второй этап) и функциях-фактах (третий этап).

¹³Данный пример прекрасно визуализирует важный факт: программа, представленная библиотекой, и программа, сформулированная архитекторами, – не одно и то же. Между ними – проектная концепция. В случае с данной библиотекой – концепция «книжной спирали» – новой модели пространственной группировки книг – не по отдельным помещениям согласно традиционным разделам десятичного классификатора Дьюи, а в виде единой спиралевидной формы. Эта концепция реструктурировала все пространства библиотеки.

¹⁴Автор этой 3D-модели – Алекс Хогреф (Alex Hogrefe) разработал ее в рамках учебного магистерского проекта. Модель выполнена в программе Sketchup.

Библиография

1. Cherry, E. Architectural Programming/ E. Cherry, J. Petronis// Whole Building Design Guide. WDBG: A program of the National Institute of Building Science. – 09.02.2009. [Электронный ресурс]. – URL: http://www.wbdg.org/design/dd_archprogramming.php
2. Раппапорт, А.Г. Проектирование без прототипов / А.Г. Раппапорт // Разработка и внедрение автоматизированных систем в проектировании (теория и методология). –

М.: Стройиздат, 1975.

3. Bugni, V. A Look at Programming/ V. Bugni, R. Smith // Connections. – September, 2002.
4. Глазычев, В.Л. Метод проектирования/ В.Л. Глазычев // Сайт Вячеслава Леонидовича Глазычева. 2008 [Электронный ресурс]. – URL: http://www.glazychev.ru/courses/2008_lecture_medot_proectirovania.htm
5. Popov, L. Evolution or Revolution: The Role of Facility Programming/ L. Popov. Interactive paper presented at the 33rd Annual Conference of EDRA, Philadelphia, PA, May 22-26, 2002 (unpublished manuscript).
6. Hershberger, R.G. Architectural Programming & Predesign Manager/ R.G. Hershberger. – New York: McGraw-Hill, 1999. – 506 p.
7. Lang, J. Creating Architectural Theory. The Role of the Behavioral Sciences in Environmental Design/ J. Lang. – New York: Van Nostrand Reinhold Company, Inc., 1987. – 278 p.
8. Cherry, E. Programming for Design: from theory to practice / E. Cherry. – New York: John Wiley, 1999. – 327 p.
9. Peña, W. Problem Seeking: an architectural programming primer / W. M. Peña, S. A. Parshall. – 5th ed. – New York: John Wiley & Sons, 2012. – 274 p.
10. Preiser, W. A Conceptual Framework for Building Performance Evaluation / W. Preiser and U. Schramm// Assessing Building Performance/ W.F.E. Preiser, J.C. Visher (eds.). – Elsevier, Oxford, 2005. – p.15-26.
11. Zeisel, J. Inquiry by Design: environment/ behavior/ neuroscience in architecture, interiors, landscape, and planning/ J. Zeisel. – New York, London: W.W. Norton & Company, 2006. – 400 p.
12. Duerk, D. Architectural Programming: information management for design/ D. P. Duerk. – New York: Van Nostrand Reinhold, 1993. – 258 p.
13. Marmot, A. Programming/ briefing – programme review / A. Marmot, J. Eley, and S. Bradely// Assessing Building Performance/ W.F.E. Preiser, J.C. Visher (eds.). – Elsevier, Oxford, 2005. – pp.39-51.
14. Lawson, B. How Designers Think. The Design Process Demystified/ B. Lawson. – 4th ed. – Architectural Press – Elsevier, 2005. – 322 p.
15. Хартия ЮНЕСКО / МСА по архитектурному образованию. Пересмотренная версия. – МСА: Комиссия по архитектурному образованию, 2011. – 7 с.
16. UIA Accord on Recommended International Standards of Professionalism in Architectural Practice. Ammended August 2014 at the XXVI General Assembly, 2014. – UIA: Paris, 2014. – 19 p.

Это произведение доступно по лицензии Creative Commons «Attribution-ShareAlike» («Атрибуция — На тех же условиях») 4.0 Всемирная.



Кияненко Константин Васильевич
доктор архитектуры, профессор,
Вологодский государственный университет»,
Вологда, Россия, e-mail: design@mh.vstu.edu.ru

Статья поступила в редакцию 29.04.2016
Электронная версия доступна по адресу: http://archvuz.ru/2016_2/1
© К.В. Кияненко 2016
© УралГАХУ 2016

ARCHITECTURAL PROGRAMMING AS SOCIAL RESEARCH AND PREDESIGN MANAGEMENT

Kiyanenko Konstantin V.

Doctor of Architecture, Professor,

Vologda State University

Vologda, Russia, e-mail: design@mh.vstu.edu.ru

Abstract

One of the key design notions, «architectural programming», is commonly used in other countries and is not unknown in Russia to the extent that any analysis of its theoretical premises and challenges should be necessarily preceded by at least a summary of the latter. On the basis of classical texts on this theme, the in article considers the origins and evolution of this concept in transition from traditional society to postindustrial one, and compares different ideas concerning the levels and models of programming, assesses the role of the architect as “programmer”, follows possible relationships of programming with design, studies the structure and contents of the programming procedure, and outlines the forms of presentation of its results to the customer and other members of the “programming committee team». Conclusions are made concerning the current importance of theoretical elaboration of programming as a discipline and practical use of its methods in professional domestic practice.

Key words

architectural programming, social architectural surveys, design brief

References

1. Cherry, E., Petronis, J. (2009) Architectural Programming. In: Whole Building Design Guide. WDBG: A program of the National Institute of Building Science. 09.02.2009. [Online]. Available from: http://www.wbdg.org/design/dd_archprogramming.php
2. Rappaport, A.G. (1975) Design without Prototypes. In: Development and Implementation of Automated Design Systems in Design (Theory and Methodology). Moscow: Stroyizdat. (in Russian)
3. Bugni, V. Smith, R. (2002) A Look at Programming. In: Connections. September 2002.
4. Glazychev, V.L. (2008) Method of Designing. V.L.Glazychev’s web-site. [Online]. Available from: http://www.glazychev.ru/courses/2008_lecture_medot_proectirovania.htm (in Russian)
5. Popov, L. (2002) Evolution or Revolution: The Role of Facility Programming. Interactive paper presented at the 33rd Annual Conference of EDRA, Philadelphia, PA, May 22-26, 2002 (unpublished manuscript).
6. Hershberger, R. (1999) Architectural Programming & Pre-design Manager. New York: McGraw-Hill.
7. Lang, J. (1987) Creating Architectural Theory. The Role of the Behavioral Sciences in Environmental Design. New York: Van Nostrand Reinhold Company, Inc.
8. Cherry, E. (1999) Programming for Design: from theory to practice. New York: John Wiley.
9. Peña, W., Parshall, S.A. (2012) Problem Seeking: an architectural programming primer. 5th ed. New York: John Wiley & Sons.
10. Preiser, W., Schramm, U. (2005) A Conceptual Framework for Building Performance Evaluation. In: Preiser, W.F.E., Visher, J.C. (eds.). Assessing Building Performance. Elsevier, Oxford, pp.15-26.
11. Zeisel, J. (2006) Inquiry by Design: environment/ behavior/ neuroscience in architecture, interiors, landscape, and planning. New York, London: W.W. Norton & Company.

-
12. Duerk, D. (1993) *Architectural Programming: information management for design*. New York: Van Nostrand Reinhold.
 13. Marmot, A., Eley, J., and Bradely, S. Programming/ briefing – programme review. In: Preiser, W.F.E., Visher, J.C. (eds.). *Assessing Building Performance*. Elsevier, Oxford, pp.39-51.
 14. Lawson, B. (2005) *How Designers Think. The Design Process Demystified*. 4th ed. Architectural Press – Elsevier.
 15. *UIA/UNESCO Charter for Architectural Education. Revised version*. UIA: Architectural Education Commission, 2011.
 16. *UIA Accord on Recommended International Standards of Professionalism in Architectural Practice. Amended August 2014 at the XXVI General Assembly, 2014*. UIA: Paris, 2014