Golden Gate University Law Review

Volume 13 | Issue 2 Article 5

January 1983

# Contracting For Performance in the Procurement of Custom Computer Software

L. James Lentz

# CONTRACTING FOR PERFORMANCE IN THE PROCUREMENT OF CUSTOM COMPUTER SOFTWARE

## I. INTRODUCTION

Rudimentary forms of the computer began to appear during the 1940's in scientific laboratories and governmental installations.[1] The proliferation of automation began later in the 1950's, but courts did not begin to realize the impact of computers on our way of life until the 1960's.[2] The growth of the computer industry has spawned the need for the development of computer law.[3] One area in particular, computer contracting, requires special legal attention.[4]

Existing computer contracting procedures often lead to improperly drafted agreements in light of existing contractual remedies.[5] Computer systems may prove unreliable if suppliers develop them based upon improperly drafted agreements.[6] Because courts are reluctant to go beyond the remedies specified in a freely entered contract, the dissatisfied customers often do not

---

1. COMPUTER LAW BIBLIOGRAPHY 1979 introduction (M. Scott and D. Yen eds. 1979).
2. *Id.*
3. *Id.* The 1979 BIBLIOGRAPHY *supra*, note 1, contains listings of articles and reference materials from over 300 publications in the legal profession, computer industry and related fields such as banking, electronics and business management. For an overview of computers and computer law, *see Block, The Expanding Role of Competition*, 25 INFOSYS., May 1978, at 80; *Book Review*, 6 RUTGERS J. COMPUTERS & LAW 297 (1978) (reviewing SEIPEL, COMPUTER LAW). *Id.* at 3. *See also* Tunick, *Computer Law: An Overview*, 13 LOY. L.A.L. REV. 315 (1980).
4. *See, e.g.*, Zammit, *Contracting for Computer Products*, 54 N.Y. ST. B.J. 128 (1982) and 54 N.Y. ST. B.J. 294 (1982); D. BRANDON & S. SEGELSTEIN, DATA PROCESSING CONTRACTS (1976); R. BIGELOW & S. NYCUM, YOUR COMPUTER AND THE LAW (1975).
5. Kirchner, *DP Contract Litigation Viewed as Booming Area*, Computerworld, Nov. 6, 1978, at 9, col. 1, noted in Tunick, *Computer Law: An Overview*, 13 LOY. L.A.L. REV. 315, 316 n.10 (1980). A properly drafted contract has been defined as one which assures that the intent and expectations of each party as to the subject covered, the terms and the product's or service's performance will be plainly stated. R. BIGELOW & S. NYCUM, YOUR COMPUTER AND THE LAW 99 (1975).
6. Edge, *Protecting the Computer User*, 131 NEW L.J. 1215 (1981).

have adequate recourse under their agreements.[7] In order to avoid such a result, the parties to each business arrangement need to clearly understand their rights and duties arising from the transaction, and those rights and duties must be incorporated into the written ageement.[8] This comment will attempt to demonstrate the problems and possible remedies involved with contracting for the development of custom computer software.[9]

## II. DEFINITIONS

The first step in each computer contracting situation is to make at least a preliminary attempt at computer definitions.[10]

---

7. *Id. See also* Smith, *A Survey of Current Legal Issues Arising from Contracts for Computer Goods and Services*, 1 COMPUTER L.J. 475 (1979).

8. R. BIGELOW & S. NYCUM, YOUR COMPUTER AND THE LAW 99 (1975).

9. This comment will not address the issue of whether tort remedies provide adequate and appropriate relief when a computer software development contract is breached. For discussions of that issue, *see* Triangle Underwriters, Inc. v. Honeywell, Inc., 604 F.2d 737 (2d Cir. 1979) (strict liability); LaRossa v. Scientific Design Co. 402 F.2d 937 (3d Cir. 1968) (negligence); Price v. Neyland, 320 F.2d 674 (D.C. Cir. 1963) (negligence); Clements Auto Co. v. Service Bureau Corp., 298 F. Supp. 115 (D. Minn. 1969) (torts, generally); Ultramares Corp. v. Touche, 255 N.Y. 170, 174 N.E. 441, 74 A.L.R. 1639 (1931) (negligence). *See generally* Prince, *Negligence Liability for Defective Software*, 33 OKLA. L. REV. 848 (1980). This comment also will not discuss the applicability of Article 2 of the Uniform Commercial Code. Article 2 deals with the sale of goods. Uniform Commercial Code § 2-101 (1978) (U.C.C.) It may or may not apply to the acquisition of custom software because of its services aspect. Even if it would apply, the parties to the agreement can contractually alter its application. U.C.C. § 1-102(3) (1978). *See* Computer Servicenters, Inc. v. Beacon Manufacturing Co., 328 F. Supp. 653 (D.S.C. 1970), *aff'd*, 443 F.2d 906 (4th Cir. 1971) (contract for data processing services not covered by the U.C.C.) *cited in* Nycum, *Computer Program Liability*, 7 RUTGERS J. COMPUTER, TECH. & LAW 1, 3 n.4 (1979).

10. Honeywell, Inc. v. Lithonia Lighting, Inc., 317 F. Supp. 406, 408 (N.D. Ga. 1970). The court declared:

> After hearing the evidence in this case the first finding the court is constrained to make is that, in the computer age, lawyers and courts need no longer feel ashamed or even sensitive about the charge, often made, that they confuse the issue by resort to legal "jargon," law Latin or Norman French. By comparison, the misnomers and industrial shorthand of the computer world make the most esoteric legal writing seem as clear and lucid as the Ten Commandments or the Gettysburg Address; and to add to this Babel, the experts in the computer field, while using exactly the same words, uniformly disagree as to precisely what they mean.

*Id.* The disagreement among the experts contributes to the misunderstanding of the parties and prevents the formation of proper contracts. Taylor, *Contracts, Responsibility Need Clarification*, Computerworld, May 1, 1978, at 33, col. 1. Therefore, the parties need to agree on definitions applicable to each business transaction and should incorporate the definitions as part of the written agreement. *Id.*

Certain words and phrases have meanings peculiar to the computer industry and misunderstandings of the terms used by the contracting parties may lead to improperly worded agreements.[11] Computer systems are a series of computer programs designed to accomplish specific business functions.[12] "Hardware" is a term specifying the naked, tangible parts of the machinery.[13] It is composed of the specifications, designs, production drawings and changes related to the computer components.[14] "Software" is a term specifying the set of instructions and programs that tell the computer what it is supposed to do and when.[15] "Application software" is a term specifying the software which performs specified business functions for a particular customer.[16] Such software is sometimes called "custom software."[17]

## III. Custom Software Contract Problems

Issues that arise when contracting for custom computer software involve warranties, disclaimers of warranties, limitations of liability and definitions of acceptance of the system or software.[18] A review of some recent cases will clarify these issues and define a background against which suggestions for improving computer contracting may be considered.

In *SHA-I Corp. v. City and County of San Francisco*,[19] the

11. Honeywell, Inc. v. Lithonia Lighting, Inc., 317 F. Supp. 406, 408 (N.D. Ga. 1970).

12. University Computing Co. v. Lykes-Youngstown Corp., 504 F.2d 518, 527 n.2 (5th Cir. 1974).

13. Honeywell, Inc. v. Lithonia Lighting, Inc., 317 F. Supp. 406, 408 (N.D. Ga. 1970).

14. Com-Share Inc. v. Computer Complex, Inc., 338 F. Supp. 1229, 1231 (E.D. Mich. 1971), *aff'd per curiam*, 458 F.2d 1341 (6th Cir. 1972).

15. Law Research, Inc. v. General Automation, Inc., 494 F.2d 202, 204 n.3 (2d Cir. 1974). Software has also been defined as the set of instructions, recorded electronically on such media as magnetic tape or disk, that is to be read into the computer's central processing unit through the peripheral devices, such as tape or disk drives. Teamsters Security Fund of Northern California, Inc. v. Sperry Rand Corp. 6 Computer L. Serv. Rep. 951, 957 (N.D. Cal. 1977). However, software is a term which has no generally accepted meaning within the computer industry. Tunick, *Computer Law: An Overview*, 13 Loy. L.A.L. Rev. 315, 317 n.15 (1980).

16. Teamsters Security Fund of Northern California, Inc. v. Sperry Rand Corp., 6 Computer L. Serv. Rep. 951, 957 (N.D. Cal. 1977).

17. Edge, *Protecting the Computer User*, 131 New L.J. 1215, 1217 (1981).

18. Langs, *Vendor's View of Computer Contracts*, in Computer Law 1981—Acquiring Computer Goods and Services 391 (D. Brooks ed. 1981). *See also* Brandon, *Checklists—Software Development Contracts*, in Computer Law—Purchasing, Leasing & Licensing Hardware, Software and Services 241 (D. Brooks ed. 1980).

19. 612 F.2d 1215 (9th Cir. 1980).

Ninth Circuit held that the city's obligation to pay for a custom-designed computer system automatically arose when a thirty-day acceptance test was performed effectively.[20] The court held that the city's obligation to pay was not modified or canceled by the fact that subsequent to the test the system ceased to function satisfactorily.[21] The court reasoned that the computer system developer had assumed the risk that its system would not pass the acceptance test, but that once the system did pass, the risk that the system might not always function as tested shifted to the city.[22] The city then became obliged to pay the contract price and suffer the consequences if the system thereafter did not perform up to expectations.[23]

In *Triangle Underwriters, Inc. v. Honeywell, Inc.*[24] the Second Circuit held that a statute of limitations barred causes of action arising from a breach of a custom software development contract warranty.[25] The customer was a general agent for many insurance companies and relied on computers to process the quantities of paper that flowed through the organization.[26] Honeywell's system included both hardware and software and the software was custom-designed in part.[27] The hardware lease was signed and Honeywell began preparation of the custom software.[28] In December, 1970, Honeywell advised the customer that the system was fully operational.[29] The system was installed in January, 1971, under a contract of sale.[30] Thereafter, the customer complained each month about inaccurate processing results. Honeywell personnel attempted without success to correct the deficiencies in the software, and they continued to work on the system at the customer's site until some time in 1972 when they departed, never to return.[31]

---

20. *Id.* at 1218.
21. *Id.*
22. *Id.* The contract did not specify a different assignment of risks.
23. *Id.*
24. 604 F.2d 737 (2d Cir. 1979).
25. *Id.* at 741.
26. *Id.* at 739.
27. *Id.*
28. *Id.* at 740.
29. *Id.*
30. *Id.* Triangle had elected to purchase rather than lease the hardware.
31. *Id.*

The customer filed its initial complaint in August, 1975.[32] The district court dismissed all the counts as barred by time, and Triangle appealed.[33] The Second Circuit affirmed that the contract causes of action were so barred.[34] The Second Circuit based its decision on the conclusions that, first, the breach of warranty cause of action accrued when the system was installed and immediately proved itself incapable of functioning; and, second, Honeywell's attempts to repair the system did not toll the four-year statute of limitations.[35]

In *International Business Machines Corporation v. Catamore Enterprises, Inc.*,[36] the First Circuit held that a one-year statute of limitations agreed to by the parties, included in the purportedly comprehensive documents signed by both parties as representing their agreement, was valid, not bypassed and not easily given restrictive interpretations.[37] Although the contract and litigation histories of this case are very complex,[38] the basic issue was whether the limitations agreed to by both parties in writing were valid and governed all parts of their business arrangement.[39]

In December of 1969, Catamore had signed IBM's form "Agreement for Systems Engineering Services."[40] The agreement stated that it was to govern all IBM assistance in the installation and use of data processing products by Catamore.[41] It also purported to be the complete and exclusive agreement between the parties which superseded all proposals, oral or written, and all

32. *Id.*

33. *Id.* at 741.

34. *Id.* at 748.

35. *Id.* at 743.

36. 548 F.2d 1065 (1st Cir. 1976), *cert. denied*, 431 U.S. 960 (1977).

37. *Id.* at 1073-76.

38. The court stated, "This case is a paradigm of complex litigation, rivaling the complexity of the cybernetics era from which it arises." *Id.* at 1066-67. At the heart of the suit were the issues of scope and quality of IBM services. *Id.* at 1067. The relationships involved IBM, a supplier of electronic data processing equipment and services, and Catamore, a manufacturer and distributor of jewelry. *Id.* The events spanned the years between 1967 and 1971, and they concerned the efforts of Catamore to computerize its operations with leased IBM equipment and associated assistance of IBM services. *Id.*

39. *Id.* at 1073.

40. *Id.* at 1070. The form agreement is generally referred to as the SES agreement. *Id.*

41. *Id.*

other communications between the parties.[42] Its limitations included: (1) an exclusive remedy for Catamore limited to the amount paid for services under the contract; (2) a limit on IBM's liability which excluded the supplier's liability for lost profits or other claims, or for consequential damages; and, (3) a period of limitation with respect to actions arising out of the contract.[43] That limitation meant that no action arising out of the agreement's services could be brought by either party more than one year after the cause of action accrued.[44]

The contract specified times at which different actions would be considered to have arisen.[45] It was the difference between the times that caused the problems for Catamore.[46] All actions except IBM's action for non-payment had to be brought within one year from the date of the first due, not the last due, payment.[47] Therefore, so long as IBM sued within one year from the date of the payment most recently due, IBM could recover for all payments regardless of when their due dates occurred. On the other hand, Catamore could sue only within one year of the

---

42. *Id.*
43. *Id.*
44. *Id.*
45. The court quoted the agreement's limitation of liability clause:

> *Limitation of Liability*
>
> The Customer agrees that IBM's liability hereunder for damages, regardless of the form of action, shall not exceed the total amount paid for services under the applicable Service Estimate or in the authorization for the particular service if no Service Estimate is made. This shall be the Customer's exclusive remedy.
>
> The Customer further agrees that IBM will not be liable for any lost profits, nor for any claim or demand against the Customer by any other party.
>
> No action, regardless of form, arising out of the services under this Agreement, may be brought by either party more than one year after the cause of action has accrued, except that an action for nonpayment may be brought within one year of the date of last payment.
>
> IBM does not make any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.
>
> In no event will IBM be liable for consequential damages even if IBM has been advised of the possibility of such damages.

*Id.* at 1070 n.9. (Emphases omitted by the court).
46. *Id.* at 1070.
47. *Id.*

occurrence of a breach. The court concluded that neither of Catamore's breach of contract counterclaims had been made within one year after the causes of action had arisen.[48] The court barred the actions while applying the written agreement to the entire business arrangement between IBM and Catamore.[49] The court held as valid the provision limiting the period to one year in which IBM could bring suit against Catamore for non-payment.[50] Therefore, the court allowed the limitations clause to govern and IBM's suit to stand.[51]

If the parties agreed and understood the limitations clause, Catamore should have presented its claims within one year of the dates when the respective causes of action arose. Otherwise, Catamore should not have agreed to such terms.[52]

In *Allied Properties v. John A. Blume & Associates,*[53] the California Court of Appeal held that a professional does not impliedly warrant his design.[54] The court said that those who hire experts are not justified in expecting infallibility, but can expect only reasonable care and competence.[55] Allied Properties had hired John A. Blume & Associates to conduct an engineering feasibility study.[56] When the study resulted in construction of a structure which was little used, Allied Properties brought a suit alleging faulty design and negligence.[57]

---

48. *Id.* at 1074.

49. *Id.* The causes of action arose by June, 1971, and by December, 1971, at the latest. *Id.* at 1074 n.17. More than a year elapsed before Catamore filed its first counterclaim. *Id.*

50. *Id.* at 1076, citing Soviero Bros. Contr'g Corp. v. City of New York, 286 App. Div. 435, 142 N.Y.S.2d 508 (1955), *aff'd,* 2 N.Y.2d 924, 161 N.Y.S.2d 888, 141 N.E.2d 918 (1957).

51. *Id.* at 1076.

52. *Id.* Even though periods of limitation may appear equal in length, as in the written agreement between IBM and Catamore, their timing can be different depending upon the point in time at which each period begins. If a party needs the periods to coincide in both length and timing, then the contract language must allow for such timing coincidence by specifying the same beginning points in time for periods of equal length.

53. 25 Cal. App. 3d 848, 102 Cal. Rptr. 259 (1972).

54. *Id.* at 857, 102 Cal. Rptr. at 265. The court pointed out that its holding was consistent with the general rule that those who sell their services for the guidance of others in their economic, financial and personal affairs are not liable in the absence of negligence or intentional conduct. *Id.* at 856, 102 Cal. Rptr. at 264.

55. *Id.*

56. *Id.* at 850, 102 Cal. Rptr. at 260.

57. *Id.* at 850-55, 102 Cal. Rptr. at 259-63.

In its decision, the court emphasized that experts from John A. Blume & Associates were performing services, not furnishing insurance on the construction for Allied Properties.[58] The court then proceeded to point out that the well-settled rule in California is that where the primary objective of a transaction is to obtain services, the doctrine of implied warranty does not apply.[59]

In *National Cash Register Co. v. Marshall Savings & Loan Association,*[60] the Seventh Circuit found that a buyer of a data processing system was liable for payment even though the buyer's data were never converted for the system's software.[61] The contract in question provided that the buyer would pay for the system when invoices were rendered after the seller had delivered, installed and certified the system as ready for use.[62] The seller had sent the buyer a letter certifying that the system had been delivered, installed and made available for use.[63] The buyer had to pay for a system never received as expected.

As can be inferred from these case reviews, the supplier and customer involved in contracting for custom software should endeavor to write as complete a description as possible of the particular underlying business arrangement in which they understand themselves to be involved.

## IV. Remedies: Minimizing Risks When Contracting For Performance

When parties are contracting for custom software development, they should emphasize specific software requirements, acceptance test criteria, warranties and disclaimers of warranties, and limitations of liability including periods of limitations. The parties should define clearly and allocate all risks and costs associated with the particular business transaction in contract terms applicable to their understandings. Customers and their legal representatives must consistently follow a procedure which will

58. *Id.* at 856, 102 Cal. Rptr. at 264.

59. *Id.* at 855, 102 Cal. Rptr. at 264. The court also noted that the doctrine of strict liability would not apply. *Id.* Further, there is no "computer malpractice" tort. *See* Chatlos Systems, Inc. v. National Cash Register Corp., 479 F. Supp. 738 (1979).

60. 415 F.2d 1131 (7th Cir. 1969).

61. *Id.* at 1133.

62. *Id.* at 1132.

63. *Id.* at 1133.

best accomplish all the goals in each computer contracting situation. Therefore, it is important to have defined not only adequate checklists of contract issues and suggested contract terms but also procedures of a methodology that the parties can follow to minimize both mistakes and omissions and maximize the benefits for each computer contracting situation.

*SHA-I Corp. v. City & County of San Francisco*[64] furnishes an example of what can result from incomplete performance goals and a short acceptance period. The thirty-day acceptance test was not long enough to test periodic, or monthly, processing. The basis of the city's contracting difficulties lay within the definition of the success criteria of the acceptance test.[65] A detailed description of the acceptance test should have specified reliability criteria that included *at least* the requirement that the test be executed for a longer period of time prior to acceptance in order to better challenge not only the on-going reliability of the software but also the important periodic processing, such as month-end analysis and reporting.[66]

*Triangle Underwriters, Inc. v. Honeywell, Inc.*[67] provides an example of what can result when the contract deliverables, the curing process and understandings of when statutory periods begin to run are not defined in detail in the agreement according to the understandings of the parties.[68] The parties relied on the

64. 612 F.2d 1215 (9th Cir. 1980).

65. *Id.* at 1218. The success criteria of a test means the criteria that will be measured to judge successful completion of the test. G. MYERS, SOFTWARE RELIABILITY PRINCIPLES AND PRACTICES 243 (1976). Acceptance testing is a validation process that tests the system against initial requirements. *Id.* at 244. It should be conducted by the customer, not the software developer. *Id.* For new, custom-designed software, the customer must define and write an acceptance test with the intent of showing how the product does not meet requirements. *Id.* at 245. The test needs to exercise every decision branch in the software. *Id.* at 243. The test must attempt to prove that the software does not meet its original reliability objectives such as mean-time-to-failure, goals for the number of errors and functions for the detection, correction tolerance of software errors. *Id.* at 233.

66. The city had not tested all of the decision branches of the software if it had not tested such periodic processing. It had not adequately tested the system's performance if it had not validated the on-going reliability requirements in its own environment. *See* G. MYERS, SOFTWARE RELIABILITY PRINCIPLES AND PRACTICES 230 (1976). The customer is able to test month-end, year-end and other periodic processing without waiting for the associated calendar times to elapse. The customer is able to simulate the periods by manipulating the input and processing dates within the test facility. *Id.* at 216-18. *See also* T. DEMARCO, STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION 325 (1979).

67. 604 F.2d 737 (2d Cir. 1979).

68. *Id.* at 740-43.

law of Massachusetts as stipulated in the contract.[69] Triangle also relied on the contract being a contract for services rather than a contract for goods and thereby relied on a longer statute of limitations.[70] The court on the other hand decided that New York law applied under a center of gravity test.[71] The court then went on to point out that under New York law a contract is a contract for services rather than a contract for sale when service predominates.[72] The court concluded the converse in Triangle's case, that the contract was for the sale of goods.[73] The parties had also failed to choose governing law of a state which bore a reasonable relation to the transaction.[74] Triangle's most costly mistake was relying upon the belief that the contract was one for service without expressly and reasonably stating its characterization in the written contract with Honeywell.[75] Triangle could have prevailed in its actions based on breach of warranty if the milestones of time and their respective deliverable products, associated with payments due under the contract, had been defined and incorporated into the contract.[76]

*Allied Properties v. John A. Blume & Associates*[77] typifies disagreements that can develop between two parties when the warranties being relied upon by one party are not incorporated into the written agreement. Similarly, a company contracting for computer experts to furnish services in the development of custom software should insist that all warranties covering the services be clearly incorporated into the written agreement between the parties.

*International Business Machines Corporation v. Catamore Enterprises, Inc.*[78] focuses on incorporating into the written agreement mutually understood and agreed-to remedies and statutes of limitations. In light of the agreement it signed, Cat-

69. *Id.* at 741 n.6.
70. *Id.* at 742.
71. *Id.* at 741 n.6.
72. *Id.* at 742, citing North American Leisure Corp. v. A & B Duplicators, Ltd., 468 F.2d 695, 697 (2d Cir. 1972).
73. *Id.* at 742-43.
74. *Id.* at 741 n.6.
75. *Id.* at 742-43.
76. *Id.* at 740-48.
77. 25 Cal. App. 3d 848, 102 Cal. Rptr. 259 (1972).
78. 548 F.2d 1065 (1st Cir. 1976), *cert. denied*, 431 U.S. 760 (1977).

amore should have presented its claims against IBM in time to satisfy the limitation on which both parties had agreed in writing. Moreover, Catamore should not have agreed to the limitation if either it did not fully understand the limitation and its implications, or it could not ascertain that a breach had occurred within the period of time specified by the limitation. Finally, Catamore should have insisted on incorporating into the agreement a limitation which would have been more appropriate for the particular services and products Catamore was expecting IBM to furnish under its general contract.

*National Cash Register Co. v. Marshall Saving & Loan Association*[79] presents a situation where the enforcement of a payment liability clause by the court caused a customer to pay for a system it never accepted.[80] The customer was liable for payment when the seller certified that the system was installed and ready for use. The customer should have insisted on modifying the contract to allow for customer acceptance of the system before any payment liability accrued. The customer should also have reserved the right in the contract to review all invoices for completeness and accuracy before payments were due.

The time periods associated with the underlying business transactions should be defined in detail and incorporated into the contract as another method of minimizing the risks in the procurement of custom computer software. Periods of limitation must be defined, together with points in time at which causes of action will be considered to have arisen. The timing of invoices should be delineated so that the customer does not become liable for any payment until the software has satisfied the customer's experience acceptance test criteria, unless the parties agree to partial payments while work is in progress, so-called progress payments.

If they agree to such progress payments, the customer's liability to make each payment should be linked to the customer's acceptance of a milestone deliverable, where such a deliverable is some work product prepared by the supplier in accordance with a delineation of milestones and associated schedules at-

79. 415 F.2d 1131 (7th Cir. 1969).
80. *Id.* at 1132-33.

tached to the contract and incorporated into it. If the customer makes progress payments but is never able to accept the system in full, due to uncorrected deficiencies, then the supplier should be compelled to refund automatically any and all payments, or an agreed-to portion of them, made by the customer under the agreement.

In order to best define software requirements in a contract, extensive systems analysis must be conducted.[81] Systems analysis, or the study of a computer system,[82] involves four procedures: (1) selecting an optimal target; (2) producing detailed documentation of that target in such a manner that subsequent implementation can be evaluated to determine whether the target has been attained; (3) providing accurate predictions of costs, benefits, schedules and performance characteristics; and, (4) obtaining concurrence on each of these items from all parties involved in the particular business transaction.[83] In order to carry out these tasks, an analyst must be responsible for liaison between the user, or customer, and the supplier, as well as for software specifications, cost-benefit analyses, feasibility studies and estimations.[84]

The analyst can best satisfy these responsibilities and complete the tasks of software specification by following a particular type of analysis approach.[85] This approach can be called a structured methodology.[86] It provides for tools the analyst can use to produce products that are highly maintainable, effectively partitioned for size, complete with graphics wherever possible, and part of a logical systems model of the eventual software package.[87] The structured methodology approach, if not the tools themselves, would be useful in conducting contract negotiations and drafting written agreements or amendments that adequately define particular business arrangements.

The tools an analyst can use include, among others, data

81. G. MYERS, SOFTWARE RELIABILITY PRINCIPLES AND PRACTICES 47 (1976).
82. T. DEMARCO, STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION 4 (1979).
83. *Id.*
84. *Id.*
85. *Id.* at 15.
86. *Id.* at 16.
87. *Id.* at 15-16.

flow diagrams,[88] a data dictionary,[89] structured English,[90] decision tables[91] and decision trees.[92] The analyst may use the tools of structured analysis in accordance with any of several procedures or structured methodologies available today.[93] Once the software requirements are written and both parties agree with them, they should be incorporated into the written agreement covering the underlying business transaction associated with the particular custom software. The products of the tools may also be incorporated if adequately referenced in order to clarify and further enhance the specifications.

The acceptance test criteria need to be described in detail that is sufficiently objective to enable the customer to accept the system if it satisfies the criteria or to reject the system if it does not satisfy the criteria. The contract should specify the following terms: (1) who will conduct the test on the customer's behalf, and under what conditions the test will be carried out; (2) a cur-

---

88. Data flow diagrams are representations of networks of interrelated processes. *Id.* at 16. They are composed of named vectors representing data flows, circles or bubbles representing processes, straight lines representing files and boxes representing data sources. *Id.* at 51. Complete data flows of the computer system referred to in SHA-I Corp. v. City & County of San Francisco, 612 F.2d 1215, would have shown the required monthly or other periodic processing and reports, as well as the sources of data for each.

89. A data dictionary is a repository of data about data, which includes the set of procedures used to build and maintain the repository itself. T. DEMARCO, STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION 125 (1979). It is the set of rigorous definitions of all the data flow diagram elements. *Id.* at 126. A data dictionary can be used as a measure of the completeness and accuracy of services furnished by experts such as those discussed in Allied Properties v. John A. Blume & Associates, 25 Cal. App. 3d 848, 102 Cal. Rptr. 259 (1972).

90. Structured English is a highly orthogonal set of constructs that makes use of a limited vocabulary and a limited syntax in order to minimize the ambiguities that are regularly a part of the English language. T. DEMARCO, STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION 179-80 (1979). A review of structured English by the user's computer personnel could provide an early warning of an incompletely or inadequately defined system such as the one referred to in International Business Machines Corporation v. Catamore Enterprises, Inc., 548 F.2d 1965.

91. Decision tables are tools that can be used to distinguish among different sets of subpolicies, only one of which applies in any given situation. T. DEMARCO, STRUCTURED ANALYSIS AND SYSTEM SPECIFICATION 217 (1979). They are combinations of condition matrices and rules. *Id.* at 219-20.

92. Decision trees are graphic representations of decision tables. *Id.* at 222. A review of decision trees, or tables, together with the data flow diagrams and data dictionary could have revealed clearly and early in the performance period that the system being furnished the supplier in Triangle Underwriters, Inc. v. Honeywell, Inc., 604 F.2d 737, would not meet all the user's requirements.

93. *See, e.g.*, C. GANE & T. SARSON, STRUCTURED SYSTEMS ANALYSIS: TOOLS AND TECHNIQUES (1979); E. YOUGDON & L. CONSTANTINE, STRUCTURED DESIGN (1975).

ing process which the parties agree to follow if the customer rejects the system; (3) how notices involving acceptance, rejection and the curing process are to be communicated; (4) how the costs of the acceptance testing are to be allocated between the parties; and, (5) what remedies will be available if the system does not satisfy the acceptance test criteria even after the curing process has been followed. Specifying those remedies minimizes the inappropriateness of remedies a court might seek to set forth in the absence of remedies included in written agreement between the parties.

## V. Conclusion

Contracting for performance in the procurement of custom software requires the parties to be aware of problems and remedies from the history of software development. Effective contracting requires the parties to include in their written agreements more definitions of terms and conditions of acceptances than are required in other contracts. Effective contracting also requires the parties to include in their written agreements complete and accurate descriptions of the software goals, objectives and expected reliability. In order to avoid having a court decide on an appropriate remedy, the parties must include in their written agreement acceptable remedies for each type and event of non-performance or breach. Courts generally consider only those remedies specified in the contract in suits based on contract-related causes of action. It behooves the parties to ensure that they have incorporated into their contracts the most complete selection of remedies available at the time of contracting.

*L. James Lentz**

---

* Fourth year night student, Golden Gate University School of Law.