110 T.C. No. 34

UNITED STATES TAX COURT

NORWEST CORPORATION AND SUBSIDIARIES, Petitioner <u>v</u>.
COMMISSIONER OF INTERNAL REVENUE, Respondent

Docket Nos. 13908-92, 20567-93,   Filed June 29, 1998.
            26213-93.[1]

        Between 1986 and 1991, N, a bank holding company
whose affiliates provide banking and other financial
services, developed or modified previously developed
software for the internal management and administration
of its businesses (internal use software). N seeks tax
credits for increasing research activities pursuant to
sec. 41, I.R.C., with respect to expenditures associated
with the development of its internal use software. The
parties selected 8 of 67 internal use software activities
to serve as representative or sample activities to
determine whether all or part of the 67 activities
constituted qualified research for purposes of the tax
credit.

---

[1]   These cases are consolidated for trial, briefing, and
opinion solely with respect to the issue of whether petitioner's
activities constitute qualified research pursuant to sec. 41,
I.R.C. (the research and experimentation credit, or R&E credit).

Sec. 41(d)(1), I.R.C., sets forth four tests for qualified research: (1) The expenditures must qualify as expenses under sec. 174, I.R.C.; (2) the taxpayer must discover information which is technological in nature; (3) the taxpayer must discover information the application of which is intended to be useful in the development of a new or improved business component; and (4) substantially all of the research activities must constitute elements of a process of experimentation.

In the conference report accompanying the Tax Reform Act of 1986 (TRA 1986), Pub. L. 99-514, 100 Stat. 2085, H. Conf. Rept. 99-841 (Vol. II), at II-73 through II-74 (1986), 1986-3 C.B. (Vol. 4) 1, 73-74, Congress set forth three additional tests for qualified research under sec. 41, I.R.C., for the development of internal use software: (1) The software must be innovative; (2) the software development must involve significant economic risk; and (3) the software must not be commercially available.

The parties agree that in order for the representative internal use software activities to constitute qualified research for purposes of the tax credit, all seven tests must be satisfied.

1. Held: The three additional tests for qualified research in the development of internal use software enunciated in the conference report accompanying the TRA 1986 require a higher threshold of technological advancement and functional improvement than is necessary in other fields of research.

2. Held, further: One of the eight representative internal use software activities, the development of the Strategic Banking System's (SBS) customer module, satisfies all seven tests and constitutes qualified research pursuant to sec. 41, I.R.C. SBS was a concerted effort at advancing the state of technology in the field of computer science and pushed existing technology to new heights.

3. Held, further: N failed to establish that the remaining seven representative internal use software activities constitute qualified research.

Mark A. Hager, Albert G. Lauber, Julie W. Davis, Carl S. Kravitz, James Sottile IV, John R. Kalligher, and Matthew W. Frank, for petitioner.

Barry J. Laterman, Paul Colleran, Michael Goldbas, Tyrone J. Montague, and Bruce G. Warner, for respondent.

CONTENTS

JACOBS, Judge: By separate notices of deficiency, respondent determined the following deficiencies in petitioner's Federal income taxes:

Docket No.	Year	Deficiency
13908-92	1983	$2,605,571
	1984	2,442,134
	1985	29,187
	1986	19,301,530
20567-93	1987	93,413
	1988	3,999,398

26213-93   1989   10,532,064

Respondent also determined additional interest under section 6621(c)[2] for 1983, 1984, 1986, 1988, and 1989.

Following the filing of petitions contesting respondent's determinations, petitioner engaged Coopers & Lybrand, LLP (Coopers & Lybrand), to ascertain whether it was entitled to credits for increasing research activities pursuant to section 41 (the research and experimentation credit, or R&E credit) between 1986 and 1993 with respect to certain internal use software development activities.[3] Coopers & Lybrand concluded that 67 of 118 internal use software development activities petitioner engaged in qualified for the R&E credit. On the basis of the Coopers & Lybrand study, petitioner sought and was permitted to amend its petitions to claim the R&E credit for 1986 through 1989.

Subsequently, petitioner again sought and was permitted to amend its petitions in order to carry back the R&E credit from 1990

---

[2]   Unless otherwise indicated, all section references are to the Internal Revenue Code as in effect for the years in issue, and all Rule references are to the Tax Court Rules of Practice and Procedure.

[3]   Petitioner did not claim the R&E credit on any of its U.S. Corporation Income Tax Returns for 1986 through 1991. The engagement of Coopers & Lybrand in August 1994 followed the Department of the Treasury's issuance of proposed regulations that further defined research and experimental expenditures under sec. 174. See sec. 1.174-2, Proposed Income Tax Regs., 58 Fed. Reg. 15821 (Mar. 24, 1993).

and 1991 to 1989 and to increase the R&E credit for 1986 through 1991.

For purposes of trial, briefing, and opinion, the parties have selected 8 of the 67 internal use software development activities to serve as representative or sample activities to determine the outcome of the other 59. The parties agree that $20,552,274 of the $22,658,829 in expenditures claimed by petitioner as associated with the eight internal use software development activities have been substantiated. In addition, the parties agree that the remaining 59 activities are deemed substantiated in the same ratio (90.7 percent) as the 8 sample activities. The parties further agree that if we determine that any of the 8 sample activities qualify for the R&E credit, the remaining 59 activities will be deemed qualified for the credit according to an agreed-upon formula.

The sole issue we must decide herein is whether Norwest Corporation & Subsidiaries (petitioner or Norwest) is entitled to the R&E credit with respect to any or all of the eight sample internal use software development activities conducted between 1986 and 1991. Resolution of this issue requires us to interpret and apply the four statutory requirements for the credit provided in section 41, and the three additional requirements provided by Congress in the conference report accompanying the Tax Reform Act of 1986 (TRA 1986), Pub. L. 99-514, 100 Stat. 2085, specifically

relating to internal use software development activities. H. Conf.
Rept. 99-841 (Vol. II), at II-73 through II-74 (1986), 1986-3 C.B.
(Vol. 4) 1, 73-74 (and as adopted by the Department of the Treasury
in its proposed regulations, section 1.41-4(e)(5), Proposed Income
Tax Regs., 62 Fed. Reg. 83 (Jan. 2, 1997)).

FINDINGS OF FACT

Some of the facts have been stipulated and are so found. The
stipulation of facts and the attached exhibits are incorporated
herein by this reference.

Norwest Corporation, a Delaware corporation, is the common
parent of an affiliated group of corporations that timely filed
consolidated U.S. Corporation Income Tax Returns (Forms 1120) for
the years in issue. It is a bank holding company whose affiliates
provide banking and other financial services.

At the time the petitions were filed, Norwest Corporation had
its principal place of business in Minneapolis, Minnesota.

1. Background

During the years in issue, petitioner engaged in numerous
projects involving the development of internal use software--that
is, software developed solely for petitioner's internal business
and management purposes. Most of these projects were managed and
executed by Norwest Technical Services, Inc. (NTS or Norwest

Technical Services),[4] a wholly owned subsidiary of Norwest Corporation, which employed between 600 and 800 technical personnel, such as computer programmers. NTS occasionally sought the assistance of outside consultants and contractors to provide technical support for its projects.

All of the eight sample activities at issue in this case involve software applications. Software enables a computer to perform specific functions by providing instructions to the computer in the form of source code. The source code is written in a programming language, often according to an architectural design, which commands the computer to perform the specific tasks. (A compilation of programs or tasks necessary to operate the software often is referred to as a system.) The source code is then converted, through the use of a compiler, into executable code that is readable by the computer.

2. Software Development Methodology

To maintain discipline, structure, and standardization in the software development process, Norwest Technical Services created a formalized approach known as the Software Development Methodology (sometimes referred to as SDM). SDM was developed to prevent the need to "reinvent the wheel" by providing managers with a documented process in developing software. The result was that SDM

---

[4] Norwest Technical Services, Inc., was formerly known as Norwest Information Services, Inc.

improved controls and managed risks. SDM was applied in developing new software by Norwest personnel and in modifying software purchased from vendors. The SDM procedures were applied to each project activity, with the exception of small maintenance projects that were grouped together for purposes of SDM.

SDM involved the following nine phases, which permitted an incremental approach to funding and development:

A. Phase 1: Request

In the request phase, those seeking the development or modification of the software evaluate the feasibility, cost/benefits, and risks associated with the project. During this phase, the business problem to be solved is described, and potential business and technical solutions are discussed.

B. Phase 2: Project Initiation

In the project initiation phase, expectations regarding project scope, deliverables, approach, costs, and schedules are agreed upon, documented, and approved. Determinations are made with respect to the skills required to complete the project, and a project team is assembled. Additionally, resource commitments are made for future phases.

During this phase, a Project Risk Assessment Form is completed. This form "is used to evaluate the risk associated with completing a project [and] * * * is intended to help the Project Manager determine the kinds of structure and controls that are

necessary to ensure project success." The form does not measure the project's technical risk. Rather, the form is focused on process risk; that is, whether the experience and skills of the project team will allow completion of the project within the project's parameters, as they have been determined, including time and cost.

C. Phase 3: Definition

In the definition phase, the business requirements and needs are analyzed and prioritized. Additionally, the project's impact on business and technical functions is reviewed.

D. Phase 4: Logical Design

In the logical design phase, the design (how it will be done) of the business solution is completed on the basis of the requirements (what needs to be done) set forth in the definition phase. The logical design phase performs a so-called external view of the project (that is, examining the project from a business person's perspective to determine whether the needs of the business can be met) before the development of the technical solution, because the design of the business solution will often constrain the technical solution. Also in this phase, consideration is given to whether the necessary software can be purchased. Finally, an investment decision is made, and the cost/benefit analysis (previously performed in the request phase) is verified.

E. Phase 5: Physical Design

In the physical design phase, which is sometimes referred to as the technical design phase, the technical solution to the business problem described in the logical design phase is determined. The technical approach to building the system requires consideration of the interfaces; that is, how the software operates in conjunction with other applications or systems. During this phase, the tools for building the system are described, as well as the nature of the interfaces between the software and other systems, the estimates of volume that requires processing, and the timeframe for processing.

F. Phase 6: Development

In the development phase, physical system components are constructed, documented, and verified. During this phase, the programmers write the source code that will allow the computer to perform the required tasks, and initial plans are made for testing the software.

G. Phase 7: Testing

In the testing phase, the technical development portion of the project is completed. During this phase: (i) Unit testing is performed to determine whether the individual components of a system are technically ready for production, (ii) integrated systems testing is performed to determine whether all components of

a system can operate together, and (iii) acceptance testing is performed to determine whether the business user's requirements have been satisfied. Further, during this phase, the programmers correct errors in the coding process (known as debugging) as well as in the design process.

Typically, the testing known as regression testing is performed on prerecorded data. The use of prerecorded data is important because the testers have known solutions and outcomes against which the software can be tested, and new methods, capabilities, and functions can be tested without risk to real (production) data.

H. Phase 8: Implementation

In the implementation phase, the software is placed into production (also known as going live), usually in a gradual process. During this phase, the software application is used on actual data by the end users (those who will be using the software product). It is also during this phase that the end users are trained in the use of the software.

I. Phase 9: Postimplementation

In the postimplementation phase, the software is maintained through continuous debugging, updating, and enhancing to meet new business needs.

J. Application of the Software Development Methodology

The development of software applications does not end with any particular phase of the Software Development Methodology. The process is often iterative in that unanticipated problems are discovered in one phase that require the developers to return to a prior phase. For example, during the testing phase, the developers may learn that the software is not sufficiently scalable; that is, that it cannot handle variances in the volume of data to be processed. In such a case, the project may be returned to the physical design or development phase to reconsider the technical approach or the source code used and, after modifications, resume the testing phase. This iterative process continues until either the software is operational to the satisfaction of those involved or the project is abandoned because of either technical or business constraints.

Throughout the SDM process, some form of technical risk (the risk that the project will not succeed for technical reasons) is present, even after the software is placed into production. (The risk of failure exists throughout a project because problems that arise because of volume or scale cannot be resolved until the

implementation phase.[5]) Norwest did not maintain any formal, documented method of reporting this technical risk.

3. The Eight Sample Internal Use Software Development Activities

During the mid-1980's, Norwest made a concerted effort to expand its banking and financial services business. Between 1986 and 1991, Norwest's assets grew from approximately $20 billion to more than $38 billion (and by 1995, Norwest's assets further grew to over $75 billion). Given the anticipated growth of Norwest's business that was forecast in 1986,[6] Norwest Technical Services was assigned the task of determining the necessary data processing support structure and technology to handle this expansion.

The eight sample activities, and the substantiated expenses incurred by Norwest with respect to each activity for the years in issue, are as follows:

---

[5]   The testing of software for volume or scalability cannot be completed before the implementation phase because of the cost required to replicate a full production environment.

[6]   In 1986, petitioner expected that most States would liberalize their branch banking restrictions and permit more interstate banking services.

Activity	1986	1987	1988	1989	1990	1991	Total[1]
Strategic Banking System	[2]$451,386	$1,367,679	$2,441,576	$2,696,099	$2,260,394	$3,073,334	$12,290,468
Trust TU	373,469	623,538	551,729	742,462	649,270	650,161	3,590,629
Success	---	---	129,320	138,196	410,030	303,452	980,998
General Ledger	173,596	136,735	94,744	384,237	175,020	183,468	1,147,800
Money Transfer	85,179	173,925	193,493	224,109	125,610	180,462	982,778
Cyborg Payroll	73,881	90,923	100,683	74,991	82,056	130,714	553,248
Trust Payment	93,265	171,334	182,047	101,704	---	---	548,350
Debit Card	---	---	22,687	435,318	---	---	458,005

1   The total expenses for the eight activities equals $20,552,276. The $2 discrepancy for the total amount the parties agreed in their stipulation was substantiated is unexplained.
2   The expenses incurred on the Strategic Banking System project for 1986 were contract expenses paid to Electronic Data Systems Corp.

A. Strategic Banking System

Strategic Banking System (SBS) was a large[7] joint effort by three entities to develop an integrated banking system in which a number of software applications would operate together. Electronic Data Systems Corp. (EDS) of Dallas, Texas, and Bank One, Columbus, N.A.[8] (Bank One), of Columbus, Ohio, began the SBS project on August 18, 1986. The purpose of the SBS project was to develop a software system in which all information regarding the bank's financial products centered around the customer (the customer

---

7   By 1993, the three entities involved in developing SBS had incurred approximately $125 million in expenses and written 10 million lines of code.

8   Bank One, Columbus, N.A., is an affiliate of Banc One Corp.

module) rather than a particular account. Before the development of SBS, information relating to individual bank accounts or other financial products was not linked but rather was maintained in separate systems. Thus, when information about one account was retrieved on a computer, the user would have no way of obtaining information about other accounts owned or maintained by the customer or a related customer. The SBS project sought to change that situation by integrating all account software systems around the customer. The customer module, which would contain basic information about the customer (e.g., name, address, Social Security number, and demographics), would be integrated with deposit and loan (or credit) modules so that information relating to customer deposit and loan accounts could be readily accessed and retrieved.

Because of concerns that SBS was becoming specific solely to Bank One's needs, in late 1986 Norwest was asked and agreed to join in the development of SBS to provide more of an industry perspective. In exchange for sharing in the development costs of SBS, Norwest received a perpetual license to use the system.[9]

---

[9]   Norwest paid a base development charge of approximately $6.5 million to Bank One (although the Participant Agreement between Norwest and Bank One required a $7 million payment) in exchange for a percentage of the royalties to be paid to Bank One by Electronic Data Systems Corp. (EDS). Further, Norwest paid $1,750,000 to EDS in exchange for a perpetual license to use SBS.

Before joining the SBS development project, Norwest considered
other alternatives. In this regard, Norwest examined commercially
available software in the market, considering the products'
functionality, economic benefit, scalability, and applicability to
interstate banking activities. In particular, Norwest focused on
a software product offered by Hogan Systems, Inc. (Hogan). Norwest
ultimately rejected the Hogan product because it did not meet
Norwest's volume and economic benefit criteria.

Norwest also considered building its own system. It abandoned
this idea, however, believing that the opportunity to enter into a
development project with EDS and Bank One would enable it to
achieve its goal and limit its risk. Accordingly, on December 16,
1986, Norwest entered into a Participant Agreement[10] with Bank One,
and a separate Participant License Agreement with EDS.

Section 3.2 of the Participant Agreement between Bank One and
Norwest provided for Norwest's role in the development of SBS:

> In connection with the development of the Financial
> System by EDS as contemplated under the [EDS-Bank One]
> Participation Agreement, Norwest will, on a timely basis

---

[9](...continued)
As part of a separate license agreement, Norwest paid EDS $1.2
million to use the software in connection with services provided
to nonaffiliated financial institutions.
    Under the participant and license agreements, EDS maintained
all ownership rights to the SBS system.

[10]   The Participant Agreement was amended on Sept. 8, 1988,
Feb. 8, 1992, and again in July 1992.

as requested by * * * [Bank One] so long as this
Agreement shall be in effect:

(a) Perform its tasks in connection with the
Development Plan including providing banking
expertise required from time to time by * * *
[Bank One] and provide management decisions
and support as reasonably required by * * *
[Bank One] to develop the Development Plan.

(b) Develop with the cooperation and
assistance of * * * [Bank One], test data and
test conditions which * * * [Bank One] will
use in connection with its obligations with
EDS in order to perform system testing
designed to establish that each component of
the Financial System functions in accordance
with the applicable Functional Specifications.

(c) Cooperate with * * * [Bank One], and, if
requested, with EDS, by, among other things,
making available as reasonably requested by *
* * [Bank One], office space and services at
Norwest's premises, personnel, information,
approvals, and acceptances in order that the
work required of EDS or of * * * [Bank One]
may be accomplished in accordance with the
Development Plan.

EDS' primary tasks in the development of SBS were to define

the technical environment for SBS, to provide the necessary tools

for its development and production, and to ultimately construct the

system (including the writing of the source code). Bank One's and

Norwest's primary tasks were to provide the business requirements

for the software (the logical design phase, which was the greatest

expense for the participant banks in the SBS project).   In

addition, Bank One and Norwest provided the technical expertise as

to how such systems should operate in a banking environment.

Ultimately, though, only EDS actually developed the technology and the source code.

Bank One and Norwest employees worked with EDS employees to determine the appropriate technical environment, tools, and products for SBS. (At any given time, approximately 100 people from each entity worked on the SBS project.) Employees of the three entities met approximately every 6 weeks to review and critique the work done to date and to recommend changes to the technical design.[11] As part of this process, NTS personnel conducted research and proposed solutions to design problems.

As part of the logical design phase, EDS technical personnel met with Bank One and Norwest employees to learn about the banking

---

[11] For example, Bank One and Norwest employees raised concerns about whether DB2, a relational database system EDS proposed using for SBS, could handle the volume of data the parties projected would be run on the system. Ultimately, Bank One and Norwest employees conceded that DB2 was appropriate when EDS demonstrated its success in other high-volume environments.

Another concern raised by Bank One and Norwest employees related to the use of so-called dummy terminals. EDS proposed maintaining all data in a mainframe computer (which would perform all data processing and run all applications) and placing nonintelligent dummy computers at user stations (e.g., bank teller windows and desks). The dummy computers could access the mainframe's data and software applications through a special menu screen but could not perform any processing functions on their own. Bank One objected to this proposal primarily because it was already using personal computers (PC's), which enabled its users to access data from a host computer, to input new information through software applications processed locally on the PC's, and to ship the new data back to the host computer for updating. (This is known as a client-server architecture.)

business requirements and needs for the system (a process known as data modeling). The bank employees (e.g., bank tellers) would explain to the EDS personnel the types of functionality (i.e., information) that they needed to access in using the SBS system. Additionally, business analysts at Norwest examined the various functionality requirements of the bank employees and drafted design documents indicating the recommended design approaches or changes.

After EDS completed the initial design and development phases, Norwest was used as the test bed for SBS. A large database of sample test data was created, and as early design releases arrived from EDS, Norwest ran SBS in its computer environment to test the software. These tests were conducted on multiple days of transactions to observe the performance of the system over a specific time span. After a test run, the results were analyzed, EDS employees were debriefed, and suggested changes to the design were made. These testing and redesigning phases continued for approximately 3 years in the case of the customer module and for approximately 4-1/2 years in the case of the deposit module.

Norwest discovered hundreds of problems in the modules it received from EDS. One problem that arose in the development of the customer module was its interface link to the deposit and loan application systems already in place at Norwest during that time, which was necessary until the new SBS deposit and credit modules

were developed. Another problem that arose was the development of
a so-called pointer system which allowed the end user to access
information about a customer through one of various identification
sources (e.g., name, account number, Social Security number, etc.).
The pointer system was one of the more critical functions of the
SBS software.

Many of the problems were reviewed and corrected during the
testing phase by Norwest technical staff, who would inform EDS
personnel of the problems and recommend corrective changes to the
software. (A small group of programming experts was assembled
specifically to handle the SBS problems through the use of a
specialized case tool called PACBASE, which was selected by EDS.)
Norwest employed a repeatable testing process to isolate problems
and find solutions.[12] A running list of these technical problems
was maintained in a system provided by EDS. Not all problems were
resolved in this process, and often new problems arose from the
correction of old ones. Some of the problems that arose in the
development of SBS were attributable to poor programming and the

---

[12]   This was a particularly difficult process because three
different entities were working on the development of SBS.
Norwest had to reconcile all of the changes and discover the
source of the problems. This process required the isolation of
each change made by each entity for testing purposes. Often the
attempts to correct one set of problems created new ones.

delivery of faulty code by EDS; others were attributable to poor technical design.

Ultimately, the customer module of SBS was successfully developed (although it contained many bugs). During 1989 and 1990, it was placed into production (the implementation phase) by Norwest in its banks and used for a number of years.[13] The deposit module was a technical failure. The failure was attributed to the deposit module's inability to produce the results sought by the users in

---

[13]   A test pilot program for the customer module began in Norwest's Duluth, Minnesota, bank in November 1989 and continued through early 1990. After the pilot proved successful, it was implemented throughout 1990 in Norwest's other banks around the country in what was known as release 1.2. During 1991, EDS issued a new upgraded release of SBS which required Norwest to "retrofit" the core functionality and design changes with the customization performed by Norwest in the interim period since the prior release. The upgraded release required a new round of testing and modifications by Norwest technical staff in conjunction with EDS. Many of these changes related to technical problems (such as the source code for the pointer system), and others related to nontechnical cosmetic problems (such as changing the name of a database field or the way a screen looks to the end user, or modifying reports produced by the software). A statement of work dated Oct. 10, 1992, reported:

> The migration of the Customer system from EDS
> Release 1.2 to Release 1.3.2 was completed
> for all Norwest banks using Customer in June
> of 1992. Installing Release 1.3.2 involved a
> massive effort of customization, testing and
> converting the existing data bases to the new
> release. Since the banks have been using
> Customer 1.3.2, problems have been
> identified.

terms of stability and reliability, as well as its inability to timely process or handle the volume of data required.

Because of the failure to adequately develop the deposit module, which was supposed to work in conjunction with the customer module, Norwest abandoned SBS in 1993 and turned to an alternative system created by Hogan. Following a 5-year joint venture with IBM, Hogan developed a customer module system which performed approximately 90 percent of the functions needed by Norwest and met Norwest's critical volume requirement.

After the period in issue, both the customer and deposit modules were eventually successfully implemented at Bank One. The record does not indicate the ultimate success or failure of the credit module and its implementation at Norwest or Bank One.

B. Trust TU

The Trust TU system consisted of a group of software applications designed to maintain trust accounts, including the tracking of assets, purchase and sale of securities, and collection and disbursement of income. The software for this system was purchased from a vendor in 1979 and installed in 1981. Norwest made numerous changes to the system in subsequent years.

Between 1986 and 1991, Norwest instituted hundreds of projects[14] to improve the Trust TU system's functionality (including compliance and regulatory changes) and to reduce processing costs. These projects involved all stages of the Software Development Methodology.

Between two and five senior programmers were assigned to accomplish the goal of reducing the monthly cost per trust account that was being maintained. Before 1987, the average monthly cost per account was approximately $9. By 1990, the monthly cost per account was reduced to approximately $4 or $5.

The programmers were also assigned the goal of increasing the volume of trust accounts maintained in the Trust TU system. In 1986, there were approximately 25,000 trust accounts running on the system, and the number of trust accounts was increasing by 10 to 12 percent per year. It was assumed that the existing system could not handle more than 30,000 trust accounts.

To accomplish both goals (reduction in cost and increased volume), NTS had to increase the speed of the so-called batch process[15] that occurred each night. (During the day, the Trust TU

---

[14]   Some of the smaller projects included building reports, updating software to reflect regulatory changes, converting acquired bank trusts customers, and changing screens.

[15]   The batch process consists of editing information, posting information, extracting information, and producing

system was on-line and transactions were stored. Beginning at 6 p.m., the on-line system was shut down, and the trust accounts were updated on the basis of the transactions that had occurred during the day.) There was a maximum 12-hour window to accomplish the batch process.

Before the start of the project to increase the speed of the batch process, approximately 10 of the 12 hours of available time for batch processing were being used. To reduce this time and to more efficiently use the mainframe computer's disk space (where the trust account data was stored), NTS needed to determine how to increase the number of processing jobs that could be run concurrently rather than serially. To accomplish this goal, NTS developed a module of as many as 20 processes to determine which batch processing jobs had to be run concurrently and which had to be run serially. This required a determination of those processing jobs that were independent and those that were interdependent. In this respect, the jobs had to be organized in a manner so that no two jobs were updating the same account at the same time.

After the module was designed and built, unit testing was performed on a subset of accounts on the mainframe computer. The test outcomes produced varied results. Some of the concurrent jobs

---

[15](...continued)
transactions for the following day.

ran more slowly than they did as serial jobs, and others ran
faster. To determine why some concurrent jobs ran more slowly, the
programs that constituted the module had to be reexamined and
reorganized using a different approach to the module's design. At
times, different modules were considered and used to increase
efficiency. Hundreds of test batch processes were performed over
a 3-year period to accomplish NTS' goals. Eventually, NTS was able
to reduce the total batch processing time to approximately 6 hours.

The Trust TU system was abandoned when the number of trust
accounts reached 40,000. In 1993, NTS switched to another system
(known as Compass) which used newer technology.

One of the numerous projects relating to the Trust TU system
during the years in issue involved the development of an interface
between Norwest's trust system and a vendor-supplied fiduciary tax
reporting system called Fasttax. The interface was necessary to
transfer trust account information to the tax system in order to
calculate the appropriate tax and thereafter send that information
back to the trust system in order that checks could be issued and
remitted to the various taxing authorities. The primary concerns
involved in this interface project were the accuracy of the
information coming from Fasttax to the trust system and the ability
to process that information in a timely manner for filing tax

returns.   The development of this project followed the SDM, including extensive testing on a separate test bed.

Yet another project, the Micro IS system, which was vendor supplied, was designed to automate purchases and sales of investment securities. This portfolio management system permitted a "what-if" analysis whereby the user could examine the impact of hypothetical purchases and sales. When Micro IS was purchased, it was written in Lotus (a spreadsheet program). The processing time, as much as 5 minutes, was unacceptable to NTS. Consequently, NTS suggested to the vendor that the application be written in a "higher" language, such as C, to increase the efficiency of the system.   The vendor accepted the suggestion and rewrote the application for Norwest. The result, after an iterative testing process by Norwest taking approximately 6 to 9 months, was the reduction in response time to less than 5 seconds.

The Micro IS system required two other notable changes. One was to increase the speed by which data could be extracted from the mainframe computer (which updated the data the night before) and sent to the Micro IS system. The trust portfolio data had to be extracted and loaded on to the Micro IS system from the mainframe each morning by 7 a.m., at which time portfolio managers arrived at work. As a result of this time constraint, the vendor had to rewrite a portion of the system in the C language. Norwest

conducted iterative testing on its accounts following the vendor's rewriting of the software.

The second notable change was to increase the speed by which the purchases and sales of securities by portfolio managers could be sent from the Micro IS system to the Trust TU system. This change required the writing of polling software that essentially checked the Micro IS system every 8 to 10 seconds in order to determine the occurrence of any trades that needed to be sent to the Trust TU system. The increased speed was achieved by writing the program in a language called Clipper, after the rejection of two other languages.

Each of these changes resulted in increased efficiency in the use of the Micro IS system both in terms of cost per account and speed of nightly batch processing.

C. Success

The Success system was designed in-house by Norwest to improve the speed and efficiency of the data processing for Norwest's equipment leasing business. It supported all aspects of leasing activity, including origination of contracts, tracking assets, invoicing customers, tracking collection, accounting, cash management, and off-line reporting. Success was developed between 1988 and 1991 by a group of 15 to 20 people within the Norwest Financial Information Services Group (NFISG), the data processing

department for Norwest Financial, Inc. (Norwest Financial), a Norwest Corporation subsidiary. Norwest Financial is in the consumer finance business and provides various types of lending products, sales finance, and revolving charge products, as well as data processing services to other unaffiliated consumer finance companies in the industry.

Before the development of Success, Norwest Financial Leasing (the subsidiary of Norwest Financial for which Success was developed) used a system known as Infolease for its equipment leasing business which was not meeting its needs. The Infolease system was causing delays for the users with respect to month-end data processing which in turn required the system to be off-line for approximately 3 days. During the time the system was off-line, users had to maintain records manually, and the data could be entered only after the system was on-line and running again. If problems arose in the month-end processing, the system could be down for 7 to 10 days.

NFISG lacked the expertise to maintain or enhance the Infolease system. As a result, NFISG was unable to make improvements to that system.

The development of Success required NFISG to use the networking of various operating system platforms. In Norwest's leasing offices, where the end users were situated, minicomputers

were in place. Data would be input into the minicomputers, which would then send the information to a mainframe computer, known as a Tandem system. (The Tandem system was a highly reliable and scalable data processor which stored static data; that is, data which would not change on a regular basis.) The Tandem system would then send a message back to the minicomputers indicating the receipt and update of the new data. If the entered information required number crunching, the Tandem system would send the information to the transaction processing facility (TPF) known as SWIFT.[16] These three systems--the minicomputers, Tandem, and the TPF--were all on-line systems, which meant that they were continuously updated and processed in real time, so that the results could be verified immediately.

After a transaction or data input was completed, a real-time activity (RTA) record would be created which provided a snapshot of what had occurred. The RTA records would then be written onto a tape which was used to create off-line reporting or month-end batch processing on an IBM MVS system.

During the first 6 to 9 months in the development of the Success system, NFISG personnel met with the intended end users of the system to determine their needs. These meetings, which

---

[16] Examples of the processing performed by SWIFT included billing, aging of accounts, earnings calculations, and depreciation.

continued throughout the design, coding, and testing phases, were designed to ensure a minimum level of functionality in the new system, as well as appropriate improvements from Infolease. As part of this development process, NFISG needed to learn many aspects of Norwest Financial Leasing's business. Additionally, in the early development phases, NFISG studied the Infolease system to determine what data records needed to be maintained.

The primary[17] technical concern at this point was the TPF system.   The existing system was designed to handle "short records"--referring to the size of the transactions--between the mainframe computer and the minicomputers. To obtain the efficiency needed, NFISG sought to create a system that could handle transactions with longer and varying sizes. As a result, NFISG examined the development of a message system that could pass information   efficiently.[18]   Further,   NFISG   considered   the accessibility of the data through direct-access indexing to minimize disk space and increase response time. Additionally,

---

[17]   Other issues included human interfacing--the user's ability to navigate through the system according to the screens set up on the system. Additionally, NFISG addressed off-line reporting capabilities of the system.

[18]   To increase efficiency in the TPF system, NFISG programmed the system using an "assembler-based" language, which is a lower level (i.e., each assembler instruction can be directly interpreted to machine language) and less descriptive language than most other programming languages.

there was concern over "error recovery" to maintain data integrity --the need to return the system to its state prior to the transaction in the event the transaction process failed. In order to address the problem of error recovery, the technical support staff developed a design through macros (code that is frequently used throughout the system and which, through standardization, ensures data integrity and reliability). All of these technical concerns were addressed and ultimately led to an initial design and conceptualization of the TPF system for Success.

Following the development of a basic design of the TPF system, the project was divided into smaller projects and assigned to different analysts. Code development then began, including the development of macros. (Following the coding process, the divided projects were reintegrated for the testing phase.) Additionally, some redesigning occurred upon the discovery of concepts which did not operate as anticipated. Throughout this development phase, unit testing occurred on two separate test beds (one for the TPF system and the other for the IBM MVS reporting system) followed by systems testing (e.g., the TPF with the Tandem system) and acceptance testing for the users. This coding and testing phase lasted 1-1/2 to 2 years.

Success was implemented and placed into production following the coding and testing phases, although there was some delay. As

part of the implementation phase, NFISG was required to convert the data from the Infolease system to the Success system. This required parallel operations of both systems to ensure that the data had been properly converted and that all aspects of the new system were working in accordance with the users' needs. After production began, NFISG sought to improve the response time of the minicomputers in retrieving information.

Ultimately, the Success system improved speed and efficiency in the month-end processing, reducing the processing time from 3 or more days to overnight.

The Success system was leased out to other leasing companies affiliated with Norwest Corporation.

D. General Ledger

The General Ledger (G/L) system was an integrated collection of software applications designed to support and maintain Norwest's general ledger accounts and to produce balance sheets, income statements, and other similar reports for Norwest. Approximately five NTS persons were assigned to work on the G/L system.

During the years in issue, the corporate controllers who used the G/L system annually met with NTS personnel to discuss their business goals. The primary purpose of these meetings was to discuss ways to improve the source code in order to expedite the month-end closing of the corporate books.

The G/L system was based on a widely used vendor-supplied package known as Financial Control System (FCS). After purchasing the FCS G/L system, Norwest customized approximately 25 percent of the vendor code.

Throughout the relevant time period, NTS needed to upgrade the system and install new vendor releases. This posed problems because of the customization work already conducted by NTS on the vendor code. The upgrading required NTS to study how to "retrofit" the new vendor code with both the old code and the customized code without sacrificing the functionality and efficiency needed. The retrofitting process involved the use of Norwest's SDM and included reiterative unit, integrated, and acceptance testing. The testing usually sought to isolate problems in the code that required debugging or rebuilding.

During the retrofitting process, NTS found that the vendor code was inefficient with respect to the corporate consolidation component (consolidating and eliminating transactions of the various corporate divisions) of the software. This problem required a redesigning of the software by NTS. To accomplish this redesign, NTS examined various structures (containers for holding the data) and tools for providing similar functionality as provided in the vendor code but with greater efficiency.

Another corporate consolidation project involved automating the corporate consolidation process. Before NTS' work, the corporate consolidation process was manual, requiring corporate personnel to examine cost center reports and determine the appropriate location of transactions to appropriate accounts. The automating of the corporate consolidation process through SDM involved four or five technicians conducting extensive testing over a 2-year period.

Another project with respect to the G/L involved the on-line component of the system. Every evening, the mainframe computer went off-line and updated the G/L system with transactions that had occurred during the day. However, often during the day the corporate controller's office or other Norwest personnel needed access to the G/L system to obtain on-line updates for ad hoc reporting. This presented a problem because the updating occurred at night--whereas users often needed real-time updates during the day. To accommodate the users' needs, NTS developed a "shadow file" system, which was a secondary set of files, copied from the primary files, that could be used for off-line reporting. As a result of the shadow file system, the users had access to shadow files with real-time updates.

The shadow file system presented technical issues for NTS. The shadow file system worked on a DB2 database system which, at

the time, was new to both the vendor (IBM) and to Norwest. The
Norwest users found DB2 to have a slow response time; consequently,
as users queried the database files for access to data, the queries
became backed up--each query waiting for the query ahead of it to
be completed. Although the resolution of this problem was
important, it involved no more than basic troubleshooting by IBM
and NTS personnel.

Another problem, which was similar to the one resolved by the
shadow file system, involved the great number of batch jobs
processed each evening. Because of the large number of users who
wanted reports processed following the evening updating, the
mainframe often became locked out (known as getting IO bound) so
that jobs requiring access to the same data at the same time could
not be completed. As a result, NTS both developed a system that
provided multiple data files for the jobs to run against and
created a mainframe scheduling system that allowed an organized
processing of the batch jobs.

There were several other, albeit smaller, projects that were
addressed by NTS personnel during the years in issue, ranging from
report writer customization to new releases to interface tasks.

E. Money Transfer

The money transfer, or wire transfer, system was a software
package that enabled Norwest to move funds electronically in real

time from its accounts, through the Federal Reserve System (the Federal Reserve), to the destination financial institutions.[19] (The software also permitted transfers from other institutions to Norwest.) During the years in issue, NTS was engaged in many projects[20] designed to address Federal Reserve regulatory changes and Norwest's business needs. (Norwest was experiencing a large growth in the wire transfer business--20 percent or more per year.[21])

---

[19] To effectuate a wire transfer of funds, Norwest set up a contra account (also known as a due-to/due-from account) with the Federal Reserve. When a Norwest customer wanted to make a transfer, Norwest would debit the customer's account and credit its own account at the Federal Reserve. The Federal Reserve would then debit Norwest's account and credit the destination institution's account. This procedure was used for all domestic wire transfers; international wire transfers did not go through the Federal Reserve.
    Initially, the wire transfer system used by Norwest did not provide for international wire transferring of funds. International transfers occurred through other, antiquated, systems which were manual and paper based. Norwest converted the international wire transferring to the MoneyNet system, which provided on-line, real-time transfers and validation. This was a large project that took several months of work.

[20] Some of the smaller projects included additions and changes to the screen and reporting functions of the system.

[21] Norwest was processing approximately $5 to $7 billion in wire transfers per day. The expansion of this business raised critical security issues that had to be addressed any time new technology was instituted. (In fact, Norwest was the victim, on at least one occasion, of a fraudulent wire transfer ring that infiltrated the system.) Thus, any software development was constrained by security concerns.

The money transfer system was based on vendor-supplied software, known as MoneyNet, purchased by Norwest during the early 1980's. Although the MoneyNet system did not provide all of the functionality Norwest needed, following an assessment of all other software on the market, NTS determined that MoneyNet was the most viable for its long-term needs. In anticipation that changes to MoneyNet would be necessary, Norwest acquired the system's source code from the vendor.

The first major regulatory change to affect Norwest was the Federal Reserve's so-called 5 p.m. rule. In general, this rule stated that all wire transfers were to be completed by 5 p.m. each business day. For years prior to 1986, the Federal Reserve did not enforce the rule, staying open as late as necessary to accommodate businesses whose systems failed or which otherwise could not complete their transfers before 5 p.m. But by the late 1980's, the Federal Reserve began strictly to enforce its rule, which required institutions such as Norwest to develop systems that were sufficiently reliable to ensure the timely transfer of the funds. The failure to complete transfers by 5 p.m. could result in severe consequences to Norwest: (1) It would be liable for interest payouts on the funds it was unable to transfer (an estimated $1 million per day); and (2) it would be potentially liable for any

business deals of its customers that were not completed because it was unable to transfer funds.

In response to the Federal Reserve's enforcement of the 5 p.m. rule, NTS examined the timeliness of the MoneyNet system in terms of logging and processing the wire transfers. To protect itself in the event the system crashed, NTS developed a contingency site[22] (also known as a disaster site or hot site) in which all logs of wire transfers were copied or mirrored from the production site to a second site in real time. Thus, if the production system failed, the users could simply move over to the contingency site and complete the transfers on that system. At the same time, NTS needed to quickly recover the production system from the point at which it crashed so that the users could continue logging transfers for its customers.

The second major Federal Reserve regulatory change affecting Norwest was the so-called daylight overdraft rule. Often, because of the large number of transfers to and from wire institutions, many of which were not settled until the end of each business day, some institutions maintained negative balances in their Federal Reserve accounts during some part of the day, which placed a risk

---

[22]   The wire transfer system's lack of a contingency site resulted in Norwest's being cited by the Office of the Comptroller of the Currency and Norwest's internal auditors because of the potential for large losses.

on the Federal Reserve if the wire transfers were not settled. In general, the daylight overdraft rule stated that the Federal Reserve would not permit wire transfer institutions to maintain balances in their Federal Reserve accounts below a specified minimum level. The same prohibition applied to the bank's customers with respect to their bank accounts from which the wire transfers were coming and going.

In response to this rule, Norwest developed (over a 9-month period) two programs (one each for the Federal Reserve accounts and for customer accounts) that monitored the wire transfers and essentially froze all transfers from an account once the minimum credit level was reached.

Another project, referred to as the distributed wire project, involved Norwest's attempt to consolidate all wire transfers among its branch locations to one bank location in Minneapolis. Norwest was concerned with the manual, paper-based systems used in its branch banks, particularly in the Des Moines and Omaha branches, which had a history of crashing (which meant the systems were down until repaired) because of too much volume.

NTS sought to run all wire transfers on the MoneyNet system. The bank branches were provided with a fault-tolerant system which provided protection against system crashes similar to that used in the 5 p.m. rule project's contingency site system. One technical

problem with the MoneyNet system addressed by NTS was the software's inability to take into consideration the multibanking environment of Norwest. In this regard, the software did not fully recognize all of the accounting relationships among the different Norwest banks, and thus the accounting records were often incorrect following a wire transfer.

NTS allocated five or six technicians to remedy the problem. The technicians mapped out the appropriate accounting entries in order to determine which entries were not properly made following a wire transfer. They then filled in the gaps in the code. Following each change to the code, the changed code was tested and retested. The distributed wire project improved efficiency through the elimination of much of the paper process. It also improved reliability through the fault-tolerant system.

During the years in issue, the vendor released upgrades of the MoneyNet software. The installation of these new releases posed compatibility problems for Norwest as a result of prior changes to the code made by NTS. The retrofitting process required a comparison of the new and old vendor codes with the NTS modifications to ensure that the required functionality was maintained.

For all of the MoneyNet projects, NTS employed the SDM process, continually testing the modifications to the system to

ensure that the code worked properly and that functionality was not compromised.

F. Cyborg Payroll

The Cyborg payroll software was a payroll management system designed to automate the calculation and payment of wages to employees and taxes to Federal, State, and local authorities. It also maintained and processed audit reports of those payments. The Cyborg software was vendor acquired in the late 1970's, although it was originally designed in the mid-1960's. Given the age of the system, much of NTS' effort was focused on "trying to keep it running". Eventually, after the period at issue here, Cyborg was replaced by a payroll system known as Peoplesoft.

Cyborg was customized by Norwest to meet its needs, including the need to make nonstandard payments to certain employees and to maintain different types of reports. NTS developed its own "method code" which permitted the users to customize payments or deductions for employees--a function not fully available on the original version of Cyborg.[23]

During the years in issue, there were as many as 50 discrete service projects with respect to Cyborg. They included: Payroll

---

[23] For example, customization of the various payments and deductions was necessary to reflect different tax withholding rates in different States and to reflect the imposition of local or county taxes in some jurisdictions.

method code JL--calculating the amount of paycheck deductions for savings and investments plans; Cyborg release 34--implementation of a new vendor release of the system; payroll work facility--a mail station identifier for the users; payroll general ledger reports-- changes to the reports made from the interface with the general ledger; and payroll Cyborg yearend--customizations with respect to the yearend production of documents.

Upgrades and new releases were common, particularly with respect to changes in the tax laws that required changes to the payroll system. The many customization projects involved changes to each of the three programs (the edit program, the calculation program, and the print program) that constituted the heart of Cyborg.

In using the Software Development Methodology to customize the Cyborg software, NTS reviewed the old and new source codes (obtained from the vendor) to determine where the customized code could be written without sacrificing required functionality. Following the design and development phases, NTS conducted test payroll and reporting runs in order to ensure that the users' needs were met. As part of this process, in order to verify that the results were consistent, NTS tested the system both before and after the changes were made.

One area that created a problem in customizing Cyborg for Norwest's needs was the programming language for the reporting function. The three main programs of Cyborg were written in an older COBOL language (a common mainframe programming language), but the reporting program was in a language specific to Cyborg which had severe limitations. Moreover, the Cyborg language was not designed for the mainframe systems used at Norwest, but rather for personal computers.

Besides customization, NTS was responsible for converting the payroll systems of the banks acquired by Norwest to the Cyborg system. The Cyborg software was not developed specifically to handle conversions of this nature. Consequently, NTS had to determine a means of converting the payroll data, or perform the conversions manually (in the case of small banks).

G. Trust Payment

The Trust Payment system was created in-house by Norwest in 1987 as a means for making disbursements from pension trust accounts to beneficiaries, maintaining trust account records, and reporting tax-related information. Norwest chose to build this system itself after a determination was made that commercially available trust systems were unable to meet its needs.

The Trust Payment system automated many manual processes as well as providing integrated check-cutting capabilities.

Initially, NTS developed the system using DB2 technology, as indicated supra, a relatively new relational database[24] offered and heavily promoted by IBM. Even though NTS had little experience with the technology, NTS used DB2 because it possessed many of the characteristics Norwest needed for its pension system.

NTS experienced various problems in building the Trust Payment system. First, the system ran slowly, taking several days rather than only a couple of hours to process data and produce a batch of checks. Second, occasionally, pension checks were issued in negative dollars; that is, pensioners received checks in negative amounts. Third, the system crashed when too many users were seeking to access the same data. Ultimately, these problems were solved by the NTS personnel assigned to this area.

After the Trust Payment system was implemented, NTS conducted (over a 2-year period) a thorough review of the system to determine its ability to handle timely the anticipated increase in the number of trust account checks that would need to be issued in the future. This review focused on the use of the DB2 technology and whether there was a means of making the system run faster. (The DB2 system had a reputation for being sluggish.) NTS considered several possible reasons for the slowness of the Trust Payment system but

---

[24] In general, a relational database allows the user to store data in one table which relates to (and can be used with) other tables for reporting and updating purposes.

was sure of neither the reasons nor the solutions. The process of improving the Trust Payment system required code review, the rewriting of several Trust Payment system functions, and then testing the system to observe the processing time. If the improvements were not satisfactory, NTS went on to other approaches. Throughout this process, NTS received assistance from IBM.

Although developed for in-house use by Norwest, the Trust Payment system was examined by two other companies interested in purchasing the system.

H. Debit Card

The Debit Card software system was built to support the issuance of Norwest's debit card product. A debit card is a financial product that permits payment of point-of-sale purchases directly from the card holder's checking (or other) account.[25] To accomplish payment, a number of transactions occur, as follows: Upon presentation of the debit card to a merchant, the transaction is sent to a card association network processor (in this case Visa);[26] Visa then seeks to identify the transaction as being made

---

[25]   This is as opposed to a credit card system in which the card holder makes purchases against a preestablished line of credit.

[26]   The card processor, in addition to routing transaction requests to the issuing bank, also performs card holder and
                                                        (continued...)

with a particular financial institution's (here Norwest's) product;
following recognition of Norwest as the appropriate financial
institution, Visa sends a communication to Norwest seeking
authorization of the transaction; Norwest then verifies that the
card holder has sufficient funds in his or her account to make the
purchase, authorizes the transaction, and finally debits the
holder's account.

Once Norwest decided to enter the debit card market in 1989,[27]
NTS was under pressure to complete the project by yearend so that
Norwest could market the product before its competitors. At the
time Norwest entered the debit card market, there were two
approaches to issuing the card. The first approach was through a
credit card processing system by which a third party handled all of
the transactions in a credit-card-like fashion and then sent the
transactions to the card issuer (Norwest) for payment and debit.[28]

---

[26](...continued)
account authentication and transaction authorization.

[27]   Typically, projects were identified in the year prior
to the time Norwest's business units wanted the projects
completed. However, in the case of the debit card software
project, the decision to enter the debit card market was not made
until early 1989, and the goal was to complete the project by the
end of that same year.

[28]   Under this approach, two messages are sent from the
merchant: One seeking authorization and one seeking settlement.
However, settlement is performed by the credit card processing
company in an off-line batch job (which is usually done at
                                                (continued...)

The second approach, the one used by Norwest, involved direct access to the checking (or other) account for debiting, skipping the processing middleman. This latter approach gave Norwest greater control over the product and its use. Although by the late 1980's the credit card processing version of the debit card product had been used throughout the United States, it was not available in the nine States in which Norwest competed--and thus, Norwest chose to use the direct access approach.[29]

To build the debit card system and links to Visa,[30] NTS, which applied Norwest's Software Development Methodology, relied on the Base 24 operating system that ran Norwest's automatic teller machines (ATM's) and the Visanet module, which was jointly developed by Visa and ACI, the vendor of Base 24. The Visanet module was necessary to connect the Base 24 operating system to the Visa debit card system. Norwest was the first bank to acquire,

---

[28](...continued)
night). This requires extension of short-term credit by the issuing bank to the customer, a risk banks generally consider appropriate for 60 to 80 percent of their customers.

[29]   First Data Resources (FDR), the credit card processor used by Norwest for its credit cards, lacked experience with the debit card product at the time Norwest entered the market. Further, Norwest was not comfortable with FDR's having direct access to its checking accounts.

[30]   As part of their partnership in developing the debit card system, Visa provided Norwest with technical manuals, transaction simulators, and an access point communications device.

install, and test the connection of the Visanet module to the Base
24 system for the debit card product.

NTS and Visa experienced problems with communications between
the two systems. (Two types of communications are necessary for
the debit card to operate properly: (1) A communication from Visa
to Norwest seeking authorization for a purchase (this is referred
to as an on-line transaction); and (2) a communication from Visa to
Norwest posting the transaction to the checking account and
Norwest's general ledger (known as settlement).) The most
significant communication problem was with respect to the
transaction posting communication, which often failed entirely--
potentially resulting in customers' purchasing goods and services
without having their Norwest accounts correspondingly debited.

Because the settlement information was not being received from
Visa, NTS had to create its own set of transactions and run them
through a test system to ensure that other systems (including the
checking account and general ledger systems) that were dependent on
the settlement communications operated correctly. This
communication problem required a Norwest team of four or five
technicians working for 4 or 5 months to redesign the code and
retest the system until communications were correctly established.

The introduction of the debit card product required several
changes to other systems that operated in conjunction with the

debit card system. Although these changes were relatively minor in comparison to the process of connecting the Visanet module to the Base 24 system, they still required a pattern of trial and error through testing and retesting. Further, because the debit card operated on the same system that operated the ATM card, it was important for the various Norwest systems to recognize that the debit card was a distinct product.

Ultimately, the debit card went into production and implementation in January 1990, as scheduled. Norwest entered the market with the debit card approximately 3 months ahead of its nearest competitor, First Bank.

OPINION

Our task herein is to determine whether any or all of the eight sample internal use software development activities of Norwest constitute qualified research for purposes of the tax credit provided by section 41. The parties agree that section 41 and its legislative history, combined, set forth seven tests to determine whether internal use software development activities constitute qualified research.

This is a case of first impression in this Court. However, we are mindful of the recent decision of the District Court for the Northern District of Illinois in United Stationers, Inc. v. United States, 982 F. Supp. 1279 (N.D. Ill. 1997), on appeal (7th Cir.,

Dec. 23, 1997), which addressed the application of section 41 to qualified research in the development of internal use software. We discuss that case infra.

1. Section 41

Section 41,[31] captioned "Credit For Increasing Research Activities", provides a nonrefundable credit against a taxpayer's U.S. income tax liability as provided in section 38 (general business credits). Any excess credit may be carried forward or carried back as provided in section 39. The credit is computed as an amount equal to the sum of 20 percent of the excess of the taxpayer's qualified research expenses over a "base amount", and 20 percent of the taxpayer's basic research expenses. Sec. 41(a)-(c). Qualified expenses include those amounts paid for "in-house" research services (i.e., wages) or supplies, or "contract" research by nonemployees. Sec. 41(b).

Section 41(d) defines qualified research and provides in part:

SEC. 41(d). Qualified Research Defined.--For purposes of this section--

(1) In general.--The term "qualified research" means research--

(A) with respect to which expenditures may be treated as expenses under section 174,

---

[31] The R&E credit under sec. 41 expires on June 30, 1998, with a limited exception. Taxpayer Relief Act of 1997, Pub. L. 105-34, sec. 601(a)(1), 111 Stat. 788, 861.

(B) which is undertaken for the purpose of discovering information--

(i) which is technological in nature, and

(ii) the application of which is intended to be useful in the development of a new or improved business component   of the taxpayer, and

(C) substantially all of the activities of which constitute elements of a process of experimentation for a purpose described in paragraph (3).

   *   *   *   *   *   *   *

(2) Tests to be applied separately to each business component.--For purposes of this subsection--

(A) In general.--Paragraph (1) shall be applied separately with respect to each business component of the taxpayer.

(B) Business component defined.--The term "business component" means any product, process, computer software, technique, formula, or invention which is to be--

(i) held for sale, lease, or license, or

(ii) used by the taxpayer in a trade or business of the taxpayer.

   *   *   *   *   *   *   *

(3) Purposes for which research may qualify for credit.--For purposes of paragraph (1)(C)--

(A) In general.--Research shall be treated as conducted for a purpose described in this paragraph if it relates to--

(i)   a new or improved function,

(ii)   performance, or

(iii)   reliability or quality.

(B) Certain purposes not qualified.--
Research shall in no event be treated as
conducted for a purpose described in this
paragraph if it relates to style, taste,
cosmetic, or seasonal design factors.

Section 41 excludes certain activities from qualifying for the

credit, including the development of internal use software--except

to the extent provided by regulations or to the extent the software

is developed for use in an activity or process that otherwise

qualifies for the credit. Sec. 41(d)(4)(E).

The R&E credit under section 41[32] was first created by Congress

as part of the Economic Recovery Tax Act of 1981 (ERTA), Pub. L.

97-34, sec. 221(a), 95 Stat. 172, 227, which provided for a

nonrefundable income tax credit for certain research and

experimental expenditures paid or incurred by a taxpayer during the

taxable year in carrying on a trade or business. The purpose of

the credit was to "stimulate a higher rate of capital formation and

to increase productivity", S. Rept. 97-144, at 76-77 (1981), 1981-2

C.B. 412, 438-439; H. Rept. 97-201, at 111 (1981), 1981-2 C.B. 352,

---

[32]   The tax credit was first designated sec. 44F by the
Economic Recovery Tax Act of 1981 (ERTA), Pub. L. 97-34, sec.
221(a), 95 Stat. 172, 227, and was then redesignated sec. 30 by
the Deficit Reduction Act of 1984, Pub. L. 98-369, sec. 471(c),
98 Stat. 494, 826, and then sec. 41 by the Tax Reform Act of 1986
(TRA 1986), Pub. L. 99-514, sec. 231(d)(2), 100 Stat. 2085, 2178.

358, and to "encourage business firms to perform the research necessary to increase the innovative qualities and efficiency of the U.S. economy", S. Rept. 99-313, at 694 (1986), 1986-3 C.B. (Vol. 3) 1, 694; H. Rept. 99-426, at 177 (1985), 1986-3 C.B. (Vol. 2) 1, 177.

ERTA adopted the definitions of "research" and "experimentation" as those terms are generally defined under section 174. ERTA sec. 221(a).[33] ERTA failed to address how

---

[33] ERTA sec. 221(a) defined qualified research as follows:

For purposes of this section the term "qualified research" has the same meaning as the term research or experimental has under section 174, except that such term shall not include--

    (1) qualified research conducted outside of the United States,

    (2) qualified research in the social sciences or humanities, and

    (3) qualified research to the extent funded by any grant, contract, or otherwise by another person (or any governmental entity).

In TSR, Inc. & Sub. v. Commissioner, 96 T.C. 903 (1991), we considered a taxpayer's claim to the R&E credit under then sec. 44F for 1981. The taxpayer, a manufacturer of historical and fantasy games, sought the credit for its historical research into the development of the games. In considering the taxpayer's claim, we applied the plain meaning to the terms "experimental" ("relating to, or based on [experiment]") and "laboratory" ("a place devoted to experimental study in any branch of natural science or to the application of scientific principles in testing and analysis"). We ultimately rejected the taxpayer's claim because we found that the R&E credit was originally intended by

research expenditures in developing software operated within the credit, but the House report accompanying ERTA indicated that the credit applied only where the costs were "incurred in developing new or significantly improved programs or routines that cause computers to perform desired tasks (as distinguished from other software costs where the operational feasibility of the program or routine is not seriously in doubt)". H. Rept. 97-201, supra at 114, 1981-2 C.B. at 360. This language was not adopted by the conference report which accompanied ERTA. H. Conf. Rept. 97-215, at 223 (1981), 1981-2 C.B. 481, 495.

In 1986, Congress became concerned with the lack of an express statutory definition of qualified research in the R&E credit and of taxpayers' abuse of the credit. The Senate Finance Committee report accompanying the Tax Reform Act of 1986 stated:

> After reviewing available information and testimony on the actual use of the credit to date, the committee believes that the statutory credit provision should set forth an express definition of qualified research

---

[33](...continued)
Congress to apply only to technological discoveries and not historical or other nontechnological research.

In Yellow Freight Sys., Inc. & Subs. v. United States, 24 Cl. Ct. 804 (1991), the Court of Federal Claims considered (on a motion for partial summary judgment) whether the taxpayer's development of software programs constituted qualified research under then sec. 44F for 1983 and 1984. That court found that material facts were in dispute, and thus it did not address whether the taxpayer's activities fell within the court's understanding of research and experimentation under sec. 174.

expenses for purposes of the credit. The committee believes that the definition has been applied too broadly in practice, and some taxpayers have claimed the credit for virtually any expenses relating to product development. According to early data on the credit, the Treasury has reported, many of these taxpayers do not engage in high technology activities.

S. Rept. 99-313, supra at 694-695, 1986-3 C.B. (Vol. 3) at 694-695;

see also TSR, Inc. & Sub. v. Commissioner, 96 T.C. 903, 919 (1991);

H. Rept. 99-426, supra at 178, 1986-3 C.B. (Vol. 2) at 178.

Congress amended the R&E credit by TRA 1986 sec. 231(b), 100 Stat. 2173, to provide the definition of qualified research that now is codified in section 41(d).

2. Internal Use Software

As part of the 1986 amendment to the R&E credit, Congress addressed the application of the credit to the expenditures paid or incurred in the development of internal use software. Congress did not define what it meant by internal use software, other than to indicate that the phrase refers to software that supports general and administrative functions (such as payroll, bookkeeping, or personnel management) or provides noncomputer services (such as accounting, consulting, or banking services). H. Conf. Rept. 99-841 (Vol. II), at II-73 (1986), 1986-3 C.B. (Vol. 4) 1, 73. (The parties herein agree that each of the eight sample activities constitutes internal use software.)

In the conference report accompanying the TRA 1986, the
conferees gave the Department of the Treasury responsibility for
promulgating regulations relating to the application of section 41
to research with respect to the development of internal use
software. In this regard, the conferees stated:

> The conferees intend that these regulations will
> make the costs of new or improved internal-use software
> eligible for the credit only if the taxpayer can
> establish, in addition to satisfying the general
> requirements for credit eligibility, (1) that the
> software is innovative (as where the software results in
> a reduction in cost, or improvement in speed, that is
> substantial and economically significant); (2) that the
> software development involves significant economic risk
> (as where the taxpayer commits substantial resources to
> the development and also there is substantial
> uncertainty, because of technical risk, that such
> resources would be recovered within a reasonable period);
> and (3) that the software is not commercially available
> for use by the taxpayer (as where the software cannot be
> purchased, leased, or licensed and used for the intended
> purpose without modifications that would satisfy the
> first two requirements just stated). The conferees
> intend that these regulations are to apply as of the
> effective date of the new specific rule relating to
> internal-use software; i.e., internal-use computer
> software costs that qualify under the three-part test set
> forth in this paragraph are eligible for the research
> credit even if incurred prior to issuance of such final
> regulations.

Id. at II-73 through II-74, 1986-3 C.B. (Vol. 4) at 73-74.
Congress provided that these rules would be effective to guide
taxpayers until the Department of the Treasury issues regulations
pursuant to the quoted language. The Department of the Treasury
issued proposed regulations with respect to the R&E credit for

internal use software in January 1997. The regulatory language was nearly identical to that provided in the conference report. Sec. 1.41-4(e)(5), Proposed Income Tax Regs., 62 Fed. Reg. 83 (Jan. 2, 1997).

Ordinarily, proposed regulations carry no more weight than a position advanced on brief by the Commissioner. F.W. Woolworth Co. v. Commissioner, 54 T.C. 1233, 1265-1266 (1970). However, with regard to the R&E credit under section 41, Congress has specifically expressed its position with respect to the three internal use software tests. Thus, we look to the tests as an expression of legislative intent rather than a position of the Commissioner.

The proposed regulations add two features worthy of note that are not expressly provided in the conference report accompanying the TRA 1986. First, the regulations provide that all facts and circumstances shall be considered in determining whether a taxpayer satisfies the requirements for qualified research in the development of internal use software. Second, the regulations require that the software meet a "high threshold of innovation" to obtain the credit under section 41. Sec. 1.41-4(e)(6), Proposed Income Tax Regs., 62 Fed. Reg. 83 (Jan. 2, 1997).

3. The Seven Tests

The parties agree that the combination of section 41 and the conference report accompanying the TRA 1986 results in a total of seven tests that taxpayers must satisfy to obtain the R&E credit with respect to qualified research in the development of internal use software:[34]

(1) The section 174 test (the research expenditures must qualify as expenses under section 174);[35]

(2) the discovery test (the research must be undertaken for the purpose of discovering information which is technological in nature);

(3) the business component test (the research must be undertaken for the purpose of discovering information the application of which is intended to be useful in the development of a new or improved business component of the taxpayer);

(4) the process of experimentation test (substantially all of the activities which constitute elements of a process of experimentation must relate to a new or improved function, performance, reliability, or quality);

(5) the innovativeness test (the software must be innovative (as where the software results in a reduction in cost, or improvement in speed, that is substantial and economically significant));

---

[34] Tax credits are a matter of legislative grace, and the taxpayer bears the burden of proving entitlement to the credits. Rule 142(a); Interstate Transit Lines v. Commissioner, 319 U.S. 590, 593 (1943); Segel v. Commissioner, 89 T.C. 816, 842 (1987).

[35] Sec. 41(d)(1)(A) provides that "qualified research" means research "with respect to which expenditures may be treated as expenses under section 174". We believe that the statutory phrase "may be treated as expenses under section 174" means that the expenditures must "qualify" for the deduction under sec. 174. See infra.

(6) the significant economic risk test (the software development must involve significant economic risk (as where the taxpayer commits substantial resources to the development and also there is substantial uncertainty, because of technical risk, that such resources would be recovered within a reasonable period)); and

(7) the commercial availability test (the software must not be commercially available for use by the taxpayer (as where the software cannot be purchased, leased, or licensed and used for the intended purpose without modifications that would satisfy tests (5) and (6))).

These seven tests, however, must be applied with an understanding of some of the terminology used by Congress which is not defined. To understand these tests, we will rely on the ordinary meaning of the language used in the statute, see Commissioner v. Soliman, 506 U.S. 168, 174 (1993); United States v. American Trucking Associations, Inc., 310 U.S. 534, 543-544 (1940), as well as the legislative history surrounding the promulgation of the TRA 1986, see Landgraf v. USI Film Prods., 511 U.S. 244 (1994); Trans City Life Ins. Co. v. Commissioner, 106 T.C. 274, 299 (1996).

A. The Section 174 Test

The section 174 test requires that the research expenditures qualify as expenses under section 174. Section 174 generally allows as a current deduction research and experimental expenditures which are paid or incurred by the taxpayer in connection with the operation of a trade or business. Section 174 does not define the phrase "research and experimental", but the applicable regulations require that the expenditures represent

research and development costs "in the experimental or laboratory

sense". Sec. 1.174-2(a)(1), Income Tax Regs. The regulations

further provide:

> Expenditures represent research and development costs in
> the experimental or laboratory sense if they are for
> activities intended to discover information that would
> eliminate uncertainty concerning the development or
> improvement of a product. Uncertainty exists if the
> information available to the taxpayer does not establish
> the capability or method for developing or improving the
> product or the appropriate design of the product.
> Whether expenditures qualify as research or experimental
> expenditures depends on the nature of the activity to
> which the expenditures relate, not the nature of the
> product or improvement being developed or the level of
> technological advancement the product or improvement
> represents.

T.D. 8562, 1994-2 C.B. 30, 31.[36] "Product" refers to any pilot,

model, process, formula, invention, technique, patent, or similar

property. Sec. 1.174-2(a)(2), Income Tax Regs.

On brief, respondent concedes that the expenditures associated

with all eight of the sample internal use software activities "may

be treated as expenses under I.R.C. § 174" (emphasis added), the

language used in section 41(d)(1)(A). Respondent asserts that it

is the Commissioner's policy not to disturb a taxpayer's method of

accounting with regard to computer software expenditures if the

---

[36] The 1994 regulations under sec. 174 are effective for
all taxable years after Oct. 3, 1994. Because the amendments
provided in the 1994 regulations only clarify the definition of
research or experimental expenditures, retroactivity was
unnecessary. 59 Fed. Reg. 50159, 50160 (Oct. 3, 1994).

requirements of Rev. Proc. 69-21 are satisfied. Rev. Proc. 69-21,

1969-2 C.B. at 303, states that "The costs of developing software

* * * in many respects so closely resemble the kind of research and

experimental expenditures that fall within the purview of section

174 * * * as to warrant accounting treatment similar to that

accorded such costs under that section."[37] Respondent notes that

his concession of the section 174 test "does not mean, however,

---

[37]   The 1983 proposed regulations under sec. 174 provided
extensive language and examples relating to the treatment of the
development of computer software. The proposed regulations
provided that the term "research or experimental expenditures"
included the costs for developing "new or significantly improved
computer software. The term does not include costs paid or
incurred for the development of software the operational
feasibility of which is not seriously in doubt." Sec. 1.174-
2(a)(3), Proposed Income Tax Regs., 48 Fed. Reg. 2799 (Jan. 21,
1983). This language appears to have been derived from the House
report accompanying ERTA, which discussed the treatment of
computer software development under the R&E credit (under then
sec. 44F). H. Rept. 97-201, at 114 (1981), 1981-2 C.B. 352, 360.
     The treatment of computer software in the 1983 proposed
regulations was criticized by commentators because the proposed
regulations treated computer software differently than other
products, imposing a more difficult test. 54 Fed. Reg. 21225
(May 17, 1989). The 1989 proposed regulations removed the
controversial language and stated that computer software was to
be treated the same as other products. Id. at 21227.
     Although Notice 87-12, 1987-1 C.B. 432, indicated that the
IRS intended to issue regulations further explaining the
treatment of computer software under sec. 174, no regulations
were forthcoming. The 1993 proposed regulations only indicated
that no additional conditions were to be imposed on computer
software development activities, and that taxpayers could
continue to rely upon Rev. Proc. 69-21, 1969-2 C.B. 303. The
1994 final regulations are silent altogether. Sec. 1.174-2,
Proposed Income Tax Regs., 58 Fed. Reg. 15821 (Mar. 24, 1993).

that respondent concedes that petitioner has met the requirements of I.R.C. § 174, which is simply not an issue in this case."

Petitioner objects to respondent's claim that its expenditures "may be treated" as expenses under section 174 without in fact meeting the section's requirements for a deduction. Petitioner hazards a guess that respondent wants to have it both ways; that is, respondent wants to avoid an adverse ruling on the issue (and its potential impact on other tests under section 41) without conceding that the section 174 requirements are actually satisfied.

We believe that the phrase "the research expenditures may be treated as expenses under section 174" is meant to require the taxpayer to satisfy all the elements for a deduction under section 174. The legislative history of section 41 supports this requirement. See H. Conf. Rept. 99-841 (Vol. II), supra at II-71, 1986-3 C.B. (Vol. 4) at 71 ("the conference agreement limits research expenditures eligible for the incremental credit to 'research or experimental expenditures' eligible for expensing under section 174. * * * Under the conference agreement, research satisfying the section 174 expensing definition is eligible for the credit").

Thus, on the basis of respondent's concessions, each of petitioner's eight sample activities satisfies the elements for a deduction under section 174.

B. The Discovery Test

The discovery test requires that the research be undertaken for the purpose of discovering information which is technological in nature. In the conference report accompanying the TRA 1986, Congress explained the discovery test as follows:

> The determination of whether the research is undertaken for the purpose of discovering information that is technological in nature depends on whether the process of experimentation utilized in the research fundamentally relies on principles of the physical or biological sciences, engineering, or computer science[3]-- in which case the information is deemed technological in nature--or on other principles, such as those of economics--in which case the information is not to be treated as technological in nature. For example, information relating to financial services or similar products (such as new types of variable annuities or legal forms) or advertising does not qualify as technological in nature.

> [3] Research does not rely on the principles of computer science merely because a computer is employed. Research may be treated as undertaken to discover information that is technological in nature, however, if the research is intended to expand or refine existing principles of computer science.

H. Conf. Rept. 99-841 (Vol. II), supra at II-71 through II-72 & n.3, 1986-3 C.B. (Vol. 4) at 71-72 & n.3.

The purpose of the discovery test is to limit the type of information discovered to that which is technological in nature, as opposed to nonscientific information, such as in the fields of the social sciences, arts, or humanities, that is specifically excluded by section 41(d)(4)(G). See TSR, Inc. & Sub. v. Commissioner, 96

T.C. 903 (1991). Additionally, the test is intended to limit the form of discovery to the "process of experimentation", which is defined elsewhere by the conference report accompanying the TRA 1986 and which is discussed infra.

Congress did not statutorily define the word "discovery". Petitioner asserts on brief that "discovery" has the same meaning as in the regulations under section 174, see sec. 1.174-2(a)(1), Income Tax Regs., directing the Court to the structure of section 41(d)(1)(B). Consequently, petitioner contends, if Norwest's activities satisfy the section 174 test, there is no need to address whether Norwest performed discovery in the context of the second test.

Respondent contends that the discovery tests under sections 174 and 41 are different. For several reasons, we agree. First, the discovery test under the section 174 regulations was not adopted until 1994, 8 years after the discovery test in section 41 was created in the TRA 1986--thus, we find it difficult to conclude that Congress intended the tests to have the same meaning. Second, the discovery test under the section 174 regulations relates to "uncertainty concerning the development or improvement of a product". The discovery test under section 41, on the other hand, relates to information which is "technological in nature" and which "fundamentally relies on principles of the [hard sciences]". Thus,

the tests relate to the discovery of different information. Finally, we note that the legislative history of section 41 reveals that in 1986, Congress sought to tighten the requirements for obtaining the R&E credit because taxpayers were not using the credit for high technology purposes. H. Rept. 99-426, at 178 (1985), 1986-3 C.B. (Vol. 2) 1, 178; S. Rept. 99-313, at 694 (1986), 1986-3 C.B. (Vol. 3) 1, 694-695. Congress effectuated this goal by adding three tests to section 41, including the discovery test, but did not change the meaning of section 174 (and presumably the discovery test subsequently created under the section 174 regulations). H. Conf. Rept. 99-841 (Vol. II), supra at II-76, 1986-3 C.B. (Vol. 4) at 71, 76; see also S. Rept. 97-144, at 81 (1981), 1981-2 C.B. 412, 441. We therefore conclude that Congress intended to treat the discovery test under section 41 more narrowly than the discovery test created under section 174.

The dictionary definition of "discover" is "to make known or visible" or "to obtain sight or knowledge of for the first time". Webster's Ninth New Collegiate Dictionary (1985). The legislative history of section 41 dictates that the knowledge gained from the research and experimentation must be that which exceeds what is known in the field in which the taxpayer is performing the research and experimentation--in this case, the computer science field. The fact that the information is new to the taxpayer, but not new to

others, is not sufficient for such information to come within the
meaning of discovery for purposes of this test. The purpose of the
R&E credit was to stimulate capital formation and improve the U.S.
economy--not merely the taxpayer's business. See H. Rept. 97-201,
at 111 (1981), 1981-2 C.B. 352, 358; H. Rept. 99-426, supra at 177,
1986-3 C.B. (Vol. 2) at 177.

The legislative purpose of the discovery test in section 41
was to narrow the scope of the research and experimentation under
section 174 to that which is technological in nature. The
technological-in-nature requirement is consistent with Congress'
concern that the R&E credit was not being used for high technology
research before the 1986 amendments. S. Rept. 99-313, supra at
694-695, 1986-3 C.B. (Vol. 3) at 694-695; H. Rept. 99-426, supra at
178, 1986-3 C.B. (Vol. 2) at 178. Thus, Congress specifically
stated that the discovery process must fundamentally rely on
principles of the hard sciences--namely, physical or biological
sciences, engineering, or computer science.

The parties dispute the significance of note 3 of the
conference report, H. Conf. Rept. 99-841 (Vol. II), supra at II-71
n.3, 1986-3 C.B. (Vol. 4) at 71 n.3, accompanying the TRA 1986.
That footnote states that the research must "expand or refine" the
principles of the hard sciences. Respondent contends that the
phrase "expand or refine" is meant to explain the nature of the

discovery required by the taxpayer, noting the legislative history of section 41. On the other hand, petitioner contends that "expand or refine" is only illustrative and is intended to contrast the mere use of a computer which in and of itself does not necessarily involve fundamental reliance on the principles of computer science.

We believe the purpose of the first sentence of note 3 ("Research does not rely on the principles of computer science merely because a computer is employed.") is to emphasize that Congress intended the R&E credit not for the use of the hard sciences per se, but for the use of the principles of the hard sciences in conducting research. "Principle" is defined as "a comprehensive and fundamental law, doctrine, or assumption". Webster's Ninth New Collegiate Dictionary (1985). Clearly, the use of a computer does not necessarily require reliance on any fundamental laws, doctrines, or assumptions.

The purpose of the second sentence of note 3 ("Research may be treated as undertaken to discover information that is technological in nature, however, if the research is intended to expand or refine existing principles of computer science.") is to indicate that the taxpayer must discover information with respect to the principles of the hard sciences on which it fundamentally relies. Note 3 sets forth two means of discovering information about the principles of the hard sciences--either by "expanding", or by "refining"--either

or both of which allows the taxpayer "to make known or visible" or "to obtain sight or knowledge for the first time", the definition of discovery discussed above. Congress' goals of stimulating a higher rate of capital formation and improving the U.S. economy cannot be achieved unless the taxpayer goes beyond the preexisting knowledge of the principles of the hard sciences. Expanding or refining those principles are two, but not the exclusive, ways of satisfying these goals.[38]

C. The Business Component Test

The business component test (which is actually part of the discovery test) requires that the research be undertaken for the purpose of discovering information the application of which is intended to be useful in the development of a new or improved business component of the taxpayer. This test was addressed by Congress as follows:

> Under the conference agreement, research is treated as conducted for a functional purpose only if it relates to a new or improved function, performance, reliability, or quality. (Activities undertaken to assure achievement of the intended function, performance, etc. of the business component after the beginning of commercial production of the component do not constitute qualified

---

[38]   Respondent relies on the language in TSR, Inc. & Sub. v. Commissioner, 96 T.C. 903 (1991), in which we stated that the technological-in-nature requirement amounted to a requirement of a "technological breakthrough". Id. at 920. Inasmuch as TSR, Inc. did not involve the R&E credit as it exists under sec. 41, we do not consider that language an element of the discovery test.

experimentation.) The conference agreement also provides
that research relating to style, taste, cosmetic, or
seasonal design factors is not treated as conducted for
a functional purpose and hence is not eligible for the
credit.

H. Conf. Rept. 99-841 (Vol. II), supra at II-72, 1986-3 C.B. (Vol.

4) at 72.

The parties do not seriously dispute the meaning of the

business component test or the legislative history. Indeed, it is

evident that Congress intended only that the taxpayer's activities

provide some level of functional improvement, at a minimum.

D. The Process of Experimentation Test

The process of experimentation test requires that

substantially all of the activities which constitute elements of a

process of experimentation relate to a new or improved function,

performance, reliability, or quality. The process of

experimentation test, which is referenced in the discovery test, is

explained by Congress as follows:

> The term process of experimentation means a process
> involving the evaluation of more than one alternative
> designed to achieve a result where the means of achieving
> that result is uncertain at the outset. This may involve
> developing one or more hypotheses, testing and analyzing
> those hypotheses (through, for example, modeling or
> simulation), and refining or discarding the hypotheses as
> part of a sequential design process to develop the
> overall component.
>
> Thus, for example, costs of developing a new or
> improved business component are not eligible for the
> credit if the method of reaching the desired objective
> (the new or improved product characteristics) is readily

discernible and applicable as of the beginning of the research activities, so that true experimentation in the scientific or laboratory sense would not have to be undertaken to develop, test, and choose among the viable alternatives. On the other hand, costs of experiments undertaken by chemists or physicians in developing and testing a new drug are eligible for the credit because the researchers are engaged in scientific experimentation. Similarly, engineers who design a new computer system, or who design improved or new integrated circuits for use in computer or other electronic products, are engaged in qualified research because the design of those items is uncertain at the outset and can only be determined through a process of experimentation relating to specific design hypotheses and decisions as described above.

H. Conf. Rept. 99-841 (Vol. II), supra at II-72, 1986-3 C.B. (Vol. 4) at 72.

Unlike the regulations under section 174, which are silent about the means of discovering information, the conference report accompanying the TRA 1986 made it clear that a more structured method of discovery is required with respect to section 41. By requiring that at the outset uncertainty exist about the ability to develop the product in the scientific or laboratory sense, the process of experimentation test is aimed at eliminating uncertainty about the technical ability to develop the product--as opposed to uncertainty as to whether the product can be developed within certain business or economic constraints, even though the taxpayer knew that it was technically possible to develop it.

As evidence of the required uncertainty, Congress mandated the evaluation of more than one alternative, which in turn may require

the use of a structured process of experimentation through the continuous development of hypotheses that require testing and analysis until the method for reaching the objective is discovered. Congress did not specify that any particular number of hypotheses be developed by the taxpayer, but the more hypotheses that are developed, tested, and analyzed, the more likely the project will satisfy the process of experimentation test.

This test also requires that "substantially all" of the activities constitute elements of a process of experimentation. This requirement raises two questions: (1) What does the term "substantially all" mean? and (2) what activities come within the elements of a process of experimentation?

Respondent contends that "substantially all" means at least 80 percent, referring to section 1.41-2(d)(2), Income Tax Regs., which defines the term "substantially all" with respect to qualified wages under section 41 as meaning at least 80 percent of the wages paid or incurred by the taxpayer for the employee. Petitioner does not dispute respondent's proposed 80-percent test.

We agree with respondent and hold that in the context of section 41, the term "substantially all" refers to at least 80 percent of the activities that constitute elements of a process of experimentation. This interpretation is consistent with the

existing definition of "substantially all" in the regulations under section 41 with respect to qualified wages.

Congress indicated in the conference report accompanying the TRA 1986 those elements which constitute a process of experimentation. They include the developing, testing, and analyzing of hypotheses. They do not include activities performed after commercial production or implementation or otherwise set forth in section 41(d)(4). See H. Conf. Rept. 99-841 (Vol. II), supra at II-72, 1986-3 C.B. (Vol. 4) at 72. However, in the case of internal use software, exceptions are made for the modifications of commercially available software. See infra.

Thus, at least 80 percent of the activities engaged in by a taxpayer with respect to the preproduction or implementation development of a product must involve the development, testing, and analysis of hypotheses that are designed to eliminate technical uncertainty as to the development of that product. This then raises the issue of which activities in a project are to be examined together and which are to be examined separately for purposes of section 41. Congress has provided us with some guidance in a so-called shrinking back test:

> The term business component means a product, process, computer software, technique, formula, or invention that is to be held for sale, lease, or license, or is to be used by the taxpayer in a trade or business of a taxpayer. If the requirements described * * * [in the first four tests under section 41] are not met with

respect to a product, etc. but are met with respect to
one or more elements thereof, the term business component
means the most significant set of elements of such
product, etc. with respect to which all requirements are
met.

Thus, the requirements are applied first at the
level of the entire product, etc. to be offered for sale,
etc. by the taxpayer. If all aspects of such
requirements are not met at that level, the test applies
at the most significant subset of elements of the
product, etc. This "shrinking back" of the product is to
continue until either a subset of elements of the product
that satisfies the requirements is reached, or the most
basic element of the product is reached and such element
fails to satisfy the test. Treasury regulations may
prescribe rules for applying these rules where a research
activity relates to more than one business component.

H. Conf. Rept. 99-841 (Vol. II), supra at II-72 through II-73,

1986-3 C.B. (Vol. 4) at 72-73. We conclude that the shrinking back

test must be examined on a case-by-case basis to determine which

activities are part of the same product or process, and which are

so discrete as to warrant a separate evaluation.

E. The Innovativeness Test

The innovativeness test requires that the software be

innovative "as where the software results in a reduction in cost,

or improvement in speed, that is substantial and economically

significant". Id. at II-73, 1986-3 C.B. (Vol. 4) at 73. The

parties disagree over the meaning of the innovativeness test.

Respondent contends that the legislative history of section 41

mandates that we require a "high threshold of innovation", the

phrase that appears in the House and Senate reports accompanying

the TRA 1986, S. Rept. 99-313, at 694-695 (1986), 1986-3 C.B. (Vol.

3) 1, 694-695; H. Rept. 99-426, at 178 (1986), 1986-3 C.B. (Vol. 2) 1, 178, and as used by the Department of the Treasury in its proposed regulations, section 1.41-4(e)(6), Proposed Income Tax Regs., 62 Fed. Reg. 83 (Jan. 2, 1997). Further, respondent asserts that we should read this and the other internal use software tests narrowly inasmuch as they are exceptions to the general rule that internal use software activities are not eligible for the R&E credit. Sec. 41(d)(4)(E).

Petitioner argues that the language in the innovativeness test is straightforward and that we should focus on its plain meaning. Petitioner analyzes the meaning of "substantial" and "significant" to reach the conclusion that these words connote a range of 5- to 20-percent improvement in the product or process. In support of this conclusion, petitioner cites several statutes, regulations, and <u>Nabisco Brands, Inc. & Consol. Subs. v. Commissioner</u>, T.C. Memo. 1995-127 (applying 25 percent in the context of section 1253(b)(2) and discussing the various statutes and regulations which define the term "substantial").

We do not believe that quantifying by way of a percentage that which is "substantial" and "significant" will materially assist us in determining whether the innovativeness test is satisfied. Suffice it to say, the extent of the improvements required by Congress with respect to internal use software is much greater than that required in other fields. The business component test (the third of the seven tests, which applies to all research and

experimentation under section 41) requires only a "new or improved" function, whereas the innovativeness test (which applies only to internal use software development) requires change "that is substantial and economically significant". H. Conf. Rept. 99-841 (Vol. II), supra at II-73, 1986-3 C.B. (Vol. 4) at 73 (emphasis added). We therefore agree with respondent that only a "high threshold of innovativeness" will satisfy this requirement.

F. The Significant Economic Risk Test

The significant economic risk test requires that the software development involve significant economic risk "as where the taxpayer commits substantial resources to the development and also there is substantial uncertainty, because of technical risk, that such resources would not be recovered within a reasonable period". Id. Petitioner contends that the significant economic risk test requires that only a 20-percent risk need exist, because of technical uncertainty, to prevent the taxpayer from recovering its investment within a reasonable period of time. Petitioner reaches this conclusion on the basis of the same analysis it used with respect to the innovativeness test. Respondent, on the other hand, emphasizes the magnitude of the technical risk as a key factor in analyzing the internal use activities under this test.

Again, we find significant Congress' requirement of a substantial uncertainty. In both the contexts of the regulations under section 174 and the explanation of the process of experimentation test provided in the conference report accompanying

the TRA 1986, only <u>uncertainty</u> is required. The use of the word <u>substantial</u> here requires a further step in the context of the development of internal use software. As in the case of the innovativeness test, we believe the significant economic risk test requires a higher threshold of technological advancement in the development of internal use software than in other fields.

G. The Commercial Availability Test

The commercial availability test requires that the software not be commercially available for use by the taxpayer "as where the software cannot be purchased, leased, or licensed and used for the intended purpose without modifications that would satisfy [tests (5) and (6)]". <u>Id.</u> Although this language is fairly self-explanatory, respondent argues that a taxpayer's expenses incurred in the implementation of vendor releases of commercially available software can never satisfy this final test because the releases are commercially available.

We refuse to adopt such a bright-line rule. Congress clearly anticipated that some modifications to commercially available software can satisfy the fifth and sixth tests of our seven-test analysis. We must examine those modifications, including any modifications resulting from the implementation of commercially available software, on a case-by-case basis.

4. Summary of Internal Use Software Requirements Under the Seven
Tests

The higher threshold of technological advancement and
functional improvement in the development of internal use software
vis-a-vis other fields of research is consistent with the general
rule that qualified research under section 41 excludes internal use
software development. Sec. 41(d)(4)(E). Although the reasons for
such discrimination are not readily apparent, they nonetheless do
exist. The conclusion we draw from these higher threshold
requirements is that Congress sought to limit the development of
internal use software under section 41 only to those endeavors that
ventured into uncharted territory.[39] We are mindful, however, that

---

[39]   We recognize that commentators to the 1983 and 1989
proposed regulations under sec. 174 were concerned with language
that could discourage the "evolutionary" nature of research. The
Explanation of Provisions in the 1989 proposed regulations to
sec. 174 states in pertinent part:

    A number of commentators suggested that the [1983]
    proposed regulations could be read to require a
    significant improvement for an activity to qualify
    under section 174. They suggested that such a reading
    would be overly restrictive because research and
    development activities may in many instances be part of
    an evolutionary process involving a series of minor
    improvements that, when taken together over a period of
    time, lead to a significantly improved product. The
    regulations proposed by this document do not include
    the reference to "routine" or "periodic" improvements.
    * * *

54 Fed. Reg. 21225 (May 17, 1989).

    The Explanation of Provisions in the 1993 proposed
regulations to sec. 174 states in part:

we should apply these higher threshold requirements reasonably and practically and assume that Congress set standards that are not impossible to meet. See <u>United States v. American Trucking Associations, Inc.</u>, 310 U.S. 534, 543 (1940); <u>Venture Funding, Ltd. v. Commissioner</u>, 110 T.C. 236, 264 (1998) (Ruwe, <u>J.</u>, dissenting). In applying each of the seven tests under section 41 to the facts herein, we shall consider the facts and circumstances in determining whether the taxpayer performed qualified research.

<u>5. United Stationers, Inc. v. United States</u>

Both parties rely on the District Court's decision in <u>United Stationers, Inc. v. United States</u>, 982 F. Supp. 1279 (N.D. Ill. 1997), in advancing their differing positions as to whether Norwest engaged in qualified research under section 41. In <u>United Stationers</u>, the taxpayer, a large office supply wholesaler, sought the R&E credit with respect to eight software projects. To automate its business operations, the taxpayer acquired from a vendor a software package which it then customized to meet its particular needs. The court principally relied upon the report and recommendation of a magistrate judge to find the facts in the case, see <u>United Stationers, Inc. v. United States</u>, 79 AFTR 2d 97-1761,

_____

[39](...continued)
Commentators argued that the "time-line" approach of the 1989 proposed regulation was unrealistic because progress in research and development is often achieved only in small, incremental steps. * * *

58 Fed. Reg. 15819 (Mar. 24, 1993).

97-1 USTC par. 50,457 (N.D. Ill. 1997) (report and recommendation of magistrate judge), although the court did not accept all of the magistrate judge's findings. The Government therein, as in this case, conceded that the taxpayer satisfied the section 174 test.

In analyzing section 41, the District Court disallowed the credit on several grounds. First, the court concluded that the taxpayer did not discover information that was technological in nature. Applying a dictionary definition for technological ("resulting from improvement in technical processes that increases the productivity of machines and eliminates manual operations or the operations done by older machines") and discovery ("to make known" or "to obtain for the first time sight or knowledge of"), the court found that the taxpayer did no more than use the software package as a building block and that it failed to discover information that was technological in nature. In that regard, the court found that the taxpayer did not expand or refine existing principles of computer science, stating: "Rather, Stationers merely applied, modified, and at most, built upon, pre-existing, technological information already supplied to it. This is a far-cry from what Congress contemplated when it spoke of research directed at the `principles of computer science'." 982 F. Supp. at 1284.

The court next considered the process-of-experimentation test. The court defined experimentation as "the act, process, or practice of making experiments" and defined an experiment as "a test or

trial" or "a tentative procedure or policy; [especially] one adopted in uncertainty as to whether it will answer the desired purpose or bring about the desired result". The court then cited the legislative history of the test and found that it was necessary to determine the extent of uncertainty that existed in the taxpayer's projects. The court concluded that "while the aspired benefits of the projects were in doubt, the development of the means that would allow Stationers to potentially achieve those benefits was not." Id. at 1285.

Despite the court's finding that the taxpayer failed to satisfy both the discovery and process-of-experimentation tests, it considered whether the taxpayer could have satisfied any of the internal use software tests. In doing so, the court rejected the magistrate judge's findings that the taxpayer did not satisfy the innovativeness test. Instead, the court noted that the "Magistrate Judge was correct in reasoning that Stationers' projects `simply increased efficiency and revenues for Plaintiff'," but nonetheless concluded that "[the projects] all fall under the plain meaning of the definition included in the legislative history" and were thus innovative. Id. at 1287-1288.

The court did, however, agree with the magistrate judge and find that the taxpayer failed to satisfy the significant economic risk test because "`the ability to implement [the projects] was clear from the outset. The only risk or uncertainty was whether

the [projects] would produce the desired efficiency, not whether
they could, in fact, be developed.'" Id. at 1288.

The District Court's opinion is, unfortunately, of little
benefit to us because of the lack of a detailed record from which
we can compare the facts therein to the facts before us. Moreover,
we are not bound by the District Court's analysis. See A.E. Staley
Manufacturing Co. & Subs. v. Commissioner, 105 T.C. 166, 208
(1995), revd. on other grounds and remanded 119 F.3d 482 (7th Cir.
1997); Estate of Schwartz v. Commissioner, 83 T.C. 943, 952 (1984).
Consequently, we will not rely on or otherwise refer to United
Stationers in evaluating the present case.

6. The Experts

Both parties rely upon the opinions of experts in support of
their respective positions. Both parties' experts prepared initial
and rebuttal reports and testified in support of those reports.
The experts reviewed thousands of pages of documents and
interviewed many Norwest employees in the process of preparing
their reports and testimony.

We look to the parties' experts to aid us in applying the
facts relating to Norwest's eight internal use software development
activities to the seven tests for internal use software under
section 41.

We evaluate the opinions of an expert in light of the expert's
qualifications and all other evidence in the record. Estate of
Christ v. Commissioner, 480 F.2d 171, 174 (9th Cir. 1973), affg. 54

T.C. 493 (1970); <u>Parker v. Commissioner</u>, 86 T.C. 547, 561 (1986). We are not bound by the opinions of an expert, especially when they are contrary to our own judgment. <u>Orth v. Commissioner</u>, 813 F.2d 837, 842 (7th Cir. 1987), affg. <u>Lio v. Commissioner</u>, 85 T.C. 56 (1985); <u>Silverman v. Commissioner</u>, 538 F.2d 927, 933 (2d Cir. 1976), affg. T.C. Memo. 1974-285. Instead, we may reach a decision based on our own analysis of all the evidence in the record. <u>Silverman v. Commissioner</u>, <u>supra</u> at 933. We may accept the opinion of an expert in its entirety, <u>Buffalo Tool & Die Manufacturing Co. v. Commissioner</u>, 74 T.C. 441, 452 (1980), or we may be selective in the use of any portion of such an opinion, <u>Parker v. Commissioner</u>, <u>supra</u> at 562. We may also reject the expert's opinion in its entirety. <u>Palmer v. Commissioner</u>, 523 F.2d 1308, 1310 (8th Cir. 1975), affg. 62 T.C. 684 (1974).

A. Petitioner's Expert--Dr. Drew McDermott

Petitioner offered the opinion of Drew McDermott, Ph.D., a professor of computer science at Yale University. Dr. McDermott's area of expertise is artificial intelligence, and he has widely published on that topic. He also has an extensive knowledge of programming language design and implementation, formal learning theory, and philosophy of mind. However, Dr. McDermott readily admitted that he does not maintain much familiarity with the banking industry or banking software in particular.

Dr. McDermott opined that all eight of the sample internal use software activities he examined qualify for the R&E credit based on

his understanding of the seven tax law tests. However, in his

rebuttal report and at trial, he expressed reservations as to

whether the Cyborg Payroll project so qualified. When asked to

rank the eight activities from most to least characteristic of

research, Dr. McDermott provided the following list: Success and

SBS; Trust TU; MoneyNet, Trust Payment, and General Ledger; Debit

Card; and finally Cyborg Payroll.

Dr. McDermott summarized his general findings as follows:

There is usually little room for debate about whether a
project passes tests T4 and T7, having to do with
"improved business component" and commercial
availability. The main goal of most projects was a piece
of software that automated a process that was previously
done by hand, or that did essentially the same job as an
earlier piece of software, but had more features and
better performance. Even when the project failed, a goal
of this kind was usually clearly present and explicitly
stated. As far as commercial availability is concerned,
I was impressed by how thoroughly Norwest searched for
commercial products, proceeding to develop software
internally only when it had to, and usually by beginning
with a commercial product and adding functions to it that
were crucial to the banking business. * * *

Another criterion that is usually met fairly easily is
T3, the use of a process of experimentation, involving
the development and testing of hypotheses. There was
always some process of experimentation involved in the
eight sample projects. The process was generally not as
systematic as one would find in a physics or chemistry
lab. I think that reflects the state of practice in
computer science, where effects are usually less subtle
than in physics, and require less rigorous experimental
methodology.

     *   *   *   *   *   *   *

For these eight projects, it is clear that there was a
process of experimentation to reduce significant
computer-science uncertainties in every case. There are
some areas of uncertainty that were not involved in any

of these projects, although I saw evidence that they were addressed in other Norwest projects.

Dr. McDermott defined computer science as "the study of what can be accomplished by various classes of algorithm on various classes of computer architecture in a certain amount of time, or using memory in a certain way."[40] He stated that these limitations make computer science a "science". Dr. McDermott testified that in software development the issue is rarely whether something can be done at all,[41] but rather, whether something can be done given constraints, particularly in the computer environment, e.g., the type of hardware, the programming language, the degree of reliability, or the level of security.

In each of the Norwest activities (other than the Debit Card project), Dr. McDermott found that the programmers were attempting to "push a little bit beyond the current state of the art in order to produce their next product", and the question was "whether [fairly familiar elements] * * * could be put together in a

---

[40]   By "algorithm" Dr. McDermott referred to the steps a computer is supposed to execute; and by "architecture" he referred to the types of elementary steps that are available. The "time" referred to both the time necessary to develop a program and the time necessary for a program to process the selected task.

[41]   Dr. McDermott agreed that none of the Norwest projects confronted the question of whether they could be done at all. He stated that Norwest was more concerned with whether it was "technically possible to do this with the resources available, that is, with controllable development costs, manageable schedule delays, and acceptable performance when completed".

slightly new combination".[42] These types of activities, Dr. McDermott claimed, are distinguishable from what he identified as "cookbook" results--"where past practice has codified a way of solving problems of a certain kind, and it requires no creativity or experimentation to apply that method to a new problem of that kind."

Dr. McDermott claimed that a computer science project is considered research if it has "a significant chance of failure due to uncertainty regarding questions of computer science". He further identified eight types of uncertainty that when present can result in a project's characterization as research: (1) Ill definedness--the inability to formally define a problem to be solved; (2) time and space complexity--lack of sufficient computing power due to growth in data that requires an exponential growth in computing power; (3) intractability--the inability of a program to work with many different data sets; (4) software engineering--the management of complex programming projects; (5) architectural constraints--the process by which the computer completes its tasks; (6) asynchronousness--the organization of several computers operating in widely separated places; (7) security--the proper authority to enter a system; and (8) user engineering--the friendliness of a computer to the user. Dr. McDermott estimated

---

[42]   Dr. McDermott conceded that the work performed by Norwest on the eight sample activities would not produce publishable results for a textbook on algorithms.

that, to the extent that it is possible to quantify, a 20-percent level of uncertainty (in the ability to predict a program's behavior) in a project constitutes "technical risk"[43]--which, in turn, would result in a significant chance of failure.[44] Dr. McDermott was of the belief that Norwest would not have engaged in any project in which there was a greater than 50-percent chance of failure in the first place.

Finally, Dr. McDermott noted that the field of computer science does not engage in research in the same manner as other fields--i.e., the "white lab coat" experiments. Rather, he asserted that computer science research is less formal.

B. Respondent's Experts

i. Dr. Randall Davis

One of respondent's experts was Randall Davis, Ph.D., a professor of management in the electrical engineering and computer science department at Massachusetts Institute of Technology (MIT). He previously served as an associate director at the Artificial Intelligence Laboratory at MIT. Dr. Davis has been a consultant

---

[43] In his rebuttal report, Dr. McDermott defined the type of technical risk that arises in most cases as "the risk that a given computing configuration, or `architecture,' might not be programmable to perform a task within realistic time and space bounds, assuming that there are compelling reasons to use that architecture."

[44] Dr. McDermott noted, however, that software projects fail for many reasons that have nothing to do with research or technical risk, but rather with a vendor's failure to deliver a product on time, changing conditions, or the incompetency of the programmers.

for numerous corporations and has served as a court-appointed expert.

Dr. Davis concluded that all eight of the sample internal use software activities failed to satisfy one or more of the seven tests for the R&E credit.[45] He summarized his findings as follows:

> It is my opinion based on the sources [provided] * * * that the work performed by Norwest involved normal and routine software development. The software produced, in terms of the products and services provided, and the technology used to support it, was all within the then current state of the art in the industrial work of management information systems. None of the documents provided suggest that any of the software developed by Norwest was, among other things, innovative or involved a significant degree of technical risk.

Dr. Davis described five types of projects associated with software development: (1) Design and implementation (the de novo creation of a body of software); (2) installation and testing (the purchase and installation of software from a vendor); (3) maintenance (ongoing adjustments to the code); (4) enhancement (adding of functionality to the program); and (5) research (attempting to do something for the first time). Dr. Davis found that each of Norwest's activities involved at least one of the first four types of projects, and generally characterized Norwest's work as installation, interfacing, and testing. When asked to rank the eight activities in order from most to least characteristic of research, Dr. Davis provided the following list: SBS; Success and

_____

[45]   However, Dr. Davis opined that one of the activities not included in the eight sample activities, known as Expert Systems, qualified as research and experimentation under sec. 41.

Trust TU; Debit Card, Trust Payment, and Money Transfer; and
General Ledger and Cyborg Payroll.

Dr. Davis stated that routine software development must be
distinguished from software research efforts. He contended that
software research is characterized by the search for information
(as opposed to the production of code),[46] the use of test data (as
opposed to production data), and the presence of technical risk.
By "technical risk", Dr. Davis referred to the development of novel
tasks, the use of familiar technology in a new manner,[47] or the size
or complexity[48] of the project. However, according to Dr. Davis,
whether a software project is research cannot be cast in terms of
black and white: the fact that the task has been done before is

---

[46] Dr. Davis dismissed Norwest's activities as not
qualified research because Norwest produced operational software
and not information about principles.

[47] As an example of this research, Dr. Davis referred to
the development of spreadsheets in C language as opposed to the
lower level assembly-based language. At the time this was first
done, both the C language and spreadsheets were commonly known
and understood, but C had never been used to develop a
spreadsheet. The use of C reduced the amount of memory spent by
the computer in running the spreadsheet program, but it was
unclear until the project was completed that C would also be fast
enough to operate on the then-current generation of personal
computers.

[48] As an example of a large project that constitutes
research, Dr. Davis noted the attempted development of a single,
comprehensive reservation system among an airline, a hotel, and a
car rental company which spanned three different businesses,
their operating divisions, and thousands of sites. As an example
of a complex project that might constitute research, Dr. Davis
referred to the efforts of running multiple programs on multiple
machines over a network.

not controlling as to whether it is necessarily research because of the different environments in which the tasks are attempted. Further, Dr. Davis contended that technical risk[49] cannot entirely be eliminated from any project, even up to and through the time of production.

Dr. Davis explained that routine software development is characterized by the use of commercially available tools or known methodologies, both applied within their expected limits, and skilled practice.[50] He stated that routine tasks often include the moving of an existing application to a new operating system or to a new machine, translating code from one programming language to another, or putting a new interface on an existing code. Dr. Davis asserted that these projects, although difficult and challenging and requiring considerable time, effort, and skill, are not research but merely the typical part of any development effort.

Further, Dr. Davis maintained that although routine software development often involves uncertainty, trial and error, and experimentation, such factors do not convert the projects into

---

[49] Dr. Davis defined technical risk in his initial report as arising "when we don't know whether it's possible to accomplish the task in the current state of the art." However, at trial, Dr. Davis amended his definition by stating that technical risk can arise due to constraints in, for example, the type of hardware used or the resources available. In this regard, once again Dr. Davis suggested that technical risk, like his definition of research, was a matter of degree.

[50] As an analogy, Dr. Davis referred to the building of a skyscraper which, although a large and difficult task, involves the application of known methodologies and skilled practice.

research efforts. He stated that within the computer science community there is general agreement as to the basic elements of the routine software development process: Problem definition and specification, design, implementation, integration and system testing, installation and field testing, and maintenance. The development involves a constant, cyclical process of designing, testing, and modifying.[51]

Finally, Dr. Davis claimed that failure in the software development process is usually not attributable to technical risk, but is more often due to people and project management concerns.[52] Additionally, he insisted that Norwest's use of the SDM was meant to minimize risk because Norwest was developing "mission-critical" software where it could ill afford to redesign a system late in the development process. Dr. Davis conceded, however, that the activities generally appeared to provide a new or improved business function.

---

[51]   Dr. Davis testified that in the early stages of development, the modification efforts are generally characterized as redesigning, while in the later stages, as the development approaches production, the efforts are generally characterized as debugging.

[52]   Dr. Davis stated that as much as 30 percent of all software projects fail or are canceled before completion.

ii. The Tower Group[53]

Respondent's other expert was Diogo Teixeira, president of the Tower Group, a research and consulting firm specializing in information technology in the financial services industry. Mr. Teixeira is widely published on issues involving software applications in the banking industry. He lacks formal education or training in computer science or software development, or research in either of those fields.

The Tower Group report, although not specifically concluding that the eight sample internal use software activities failed to qualify for the R&E credit, found that Norwest did not develop any technology that was not within the then-current state of the art of the banking industry. The Tower Group summarized its findings as follows:

> It is our opinion, based on the documentation submitted by the petitioner and on our knowledge of the US banking industry, that the work performed by Norwest was the normal and routine data processing activities of a bank. The software produced by Norwest, in terms of the products and services provided and the technology used to support them, were all well within the then current state of the industry. We find no characteristics which would distinguish the overall work performed by Norwest within the claim from the nearly $17.5 billion spent by US

---

[53] Before trial, petitioner sought to disqualify Diogo Teixeira and the Tower Group from serving as an expert witness at trial because of a prior relationship between the Tower Group and Norwest. After hearing arguments at trial and, upon due consideration, we denied petitioner's motion for reasons expressed on the record.

Mr. Teixeira was assisted in the preparation of the Tower Group report by George T. Kivel, the group director of Wholesale Banking and General Technologies.

commercial banking industry between 1986 and 1991 on routine data processing implementation. Norwest, as a result of the work performed within the claim, developed no approaches or components that were conceptually or fundamentally different from those already in use within the banking industry.

In its report, the Tower Group stated that nothing Norwest developed could be considered unique because Norwest already possessed the necessary information.[54] The Tower Group report also stated that three factors determine whether a banking software application is unique: Product (e.g., checking account, credit card), channel (e.g., bank branch, ATM), and volume. Mr. Teixeira asserted that Norwest did not offer any products or services that were unavailable in the banking industry, and the volume of transactions was typical of banks of Norwest's size.[55] Consequently, Mr. Teixeira insisted that Norwest had numerous sources from which it could learn about software applications, and that any uncertainties Norwest faced were unique to it but not to the banking industry.[56]

_____

[54]   According to Mr. Teixeira, the sources of this information are vendors, consultants, conferences, and newsletters.

[55]   Mr. Teixeira stated that Norwest was never a top 15 bank (in terms of asset value) but fluctuated between 17th and 33d. Banks such as Citibank, Chase Manhattan, Wells Fargo, and Security Pacific were all two to three times larger than Norwest. Citibank had total assets of approximately $220 billion in 1991, while Norwest's total assets reached approximately $40 billion in 1991. In this regard, Norwest's transactional volumes never reached the levels of the larger banks'.

[56]   Mr. Teixeira opined that, on the basis of the channels,

Mr. Teixeira characterized the nature of Norwest's work as the automation of processes that previously were performed manually. In this regard, he claimed that Norwest's work was oriented primarily toward routine maintenance (correcting problems with existing applications), enhancement (adding new features to existing applications), and the implementation (deploying) of those projects, but it was not research and experimentation.[57] Further, Mr. Teixeira emphasized that Norwest's use of the Software Development Methodology was designed to limit risk and prevent the undertaking of any implementation projects with significant uncertainty.[58]

---

[56](...continued)
products, and volumes supported within the banking industry at the time, no significant technical risk existed in the eight sample activities. He found that after Norwest investigated its needs and the information available to it, the only risk that remained related to business and ability to execute; but no technical risk existed. He also indicated that generally even where technical risk exists in a bank technology project and causes its failure, the failure is usually attributable to the cost of correcting the problem--not the ability to correct it. Thus, the solution is often the purchase of more expensive technology.

Further, Mr. Teixeira described Norwest as a conservative user of technology that generally spent its time attempting to catch up to the technology already deployed by other U.S. banks.

[57]   Mr. Teixeira believed that Norwest's expenditures on maintenance and enhancement projects generally reflected industry trends.

[58]   Mr. Teixeira contended that he "[found] no evidence that technical risk was factored into the [return on investment] calculation of the projects claimed indicating that the expectation that it would impact delivery of the project was minimal."

Mr. Teixeira maintained that Norwest did not engage in research and experimentation because "At all times after the design phase, Norwest was working with <u>just one option--not a set of alternatives</u>." Mr. Teixeira identified two main attributes of technology research in the banking industry:[59] The "primary deliverable" and the "project approach". According to Mr. Teixeira, the primary deliverable in technology research in the banking industry is information which helps make decisions about other information technology projects rather than a production system. He further stated that the project approach[60] in technology

---

[58](...continued)
Further, through his understanding of the Software Development Methodology, Mr. Teixeira claimed that most risk would be resolved before the logical design phase (the business requirements phase), which is where only 20 to 30 percent of all expenditures in information technology projects occur in average U.S. banks. Mr. Teixeira asserted that very little of the work prior to the logical design phase involved research or experimentation.

[59] Mr. Teixeira noted that many top U.S. banks fund a technology research entity within their information technology organization for the purpose of "understanding when a technology is sufficiently mature for use within the bank and working with the lines of business to determine where the technology may be deployed effectively". These research groups account for less than one-half of 1 percent of the total information technology organization's budget, and even less at a bank of Norwest's size. Given these percentages, Mr. Teixeira projected that Norwest's expenditures for research and advanced development were over 50 times greater than he would have expected from an entity of its size.

[60] The technology research project approach has a pattern of: (1) Posing a question (stating what information the project is trying to determine); (2) performing targeted work (e.g., model, prototype, or literature review aimed at resolving the
(continued...)

research in the banking industry involves identifying the critical elements and capabilities of the technology under study and then modeling them in context. In the case of Norwest, Mr. Teixeira found that the work was intended to deploy a production system, not to provide information or identify the elements of technology.

Mr. Teixeira distinguished experimentation from the testing performed by Norwest. He asserted that experimentation addresses the issue of how to achieve a goal, whereas testing shows whether the goal has been reached. In this regard, he stated that most banks perform feasibility experiments which ask the question "can it be done at all?". However, at trial, Mr. Teixeira testified that these experiments also involved questions of whether the goal can be achieved given certain constraints in the business environment.

Finally, Mr. Teixeira attributed much of the inability of Norwest to complete its projects on time and within budget to management issues, not technical difficulties or risk. He defined technical risk as the probability that the chosen technological architecture combined with the user's determined features, functions, and volumes would not go into production. He believed, that on a percentage basis a 10- to 20-percent chance of failure would constitute technical risk. And with respect to Norwest's

---

[60](...continued)
question); and (3) applying the resulting information to other projects (e.g., to implement or not).

projects, he found the technical risk very low; however, he did not determine any specific percentage in his analysis. Mr. Teixeira conceded that all of the activities at issue provided a new or improved business function to Norwest, although not to the banking industry.

7. Analysis of the Eight Sample Activities

The parties' experts aided the Court in understanding research in the context of the computer science field and the banking industry. We did not find any one of the experts more helpful than another. Mr. Teixeira assisted the Court by explaining the existing state of technology in the computer science field as related to the banking industry. Drs. McDermott and Davis offered particular insight into software development issues, although they were often abstract or vague. Unfortunately, with respect to all of the experts, much of their reports and testimony was of limited use because they applied definitions and standards that are inconsistent with our interpretation of the seven tests that must be satisfied to obtain the R&E credit.[61] See <u>Alumax, Inc. v.</u>

_____

[61]   There were several problems with the definitions provided by each of the experts. For example, Dr. McDermott appears to apply a 20-percent test to the definition of substantial uncertainty--which we have rejected for the reasons expressed. Additionally, his example of a hypothesis, as in the case of the SBS project, is overly simple and not workable: "This collection of algorithms, run on such-and-such a hardware configuration, can perform such-and-such an account-management task with no errors, in such-and-such a period of time." This hypothesis cannot be used to develop true alternatives which can be examined and considered by the taxpayer.

(continued...)

Commissioner, 109 T.C. 133, 171-172 (1997); Phi Delta Theta Fraternity v. Commissioner, 90 T.C. 1033, 1041 (1988), affd. 887 F.2d 1302 (6th Cir. 1989). Thus, while we will rely on the experts' technical findings, we will generally discount their conclusions with respect to the seven tests.

A. Strategic Banking System

Petitioner's expert, Dr. McDermott, contended that at the time SBS, the integrated banking system, was developed, no existing product could have accomplished the increase in data processing capability Norwest required. He insisted that SBS was subject to several uncertainties, particularly those relating to time and space complexity, software engineering, and user-friendliness. He concluded that "The painful complexities and ultimate failure of SBS ought to be evidence that there was significant risk due to

---

[61](...continued)
Dr. Davis' definitions were too academic and did not conform to the language used by Congress. For example, he stated that "discovery" is the result of an experimental or laboratory effort, which he defined as "the creation of an isolated situation intended to mimic the real world in some respects, but tightly controlled in all other respects." We recognize that Dr. Davis was attempting to explain his understanding of research and experimentation as understood in the computer science community--but in reaching judicial decisions the definitions used by Congress are controlling.

Mr. Teixeira and the Tower Group, as well as Dr. Davis, also assumed that the ultimate goal in research is information rather than a product. This is inconsistent with the language of sec. 41, which clearly permits the ultimate goal to be a product. Also, both of respondent's experts used definitions of innovativeness that, although more familiar to us, are not consistent with the language used by Congress in the conference report accompanying the TRA 1986 on the innovativeness test.

technological uncertainty." This conclusion, Dr. McDermott claimed, is bolstered by a statement of Brian Phillips, former president of NTS, that SBS had a 50-percent chance of failure at the outset of the project.

Dr. Davis stated that the SBS project was within the then-current state of the art and asserted that any uncertainties could be eliminated through information that was reasonably available to Norwest. Further, he claimed that any risks that existed during the project were attributable to business risk, not technical risk, despite the large-scale nature of the project.

Mr. Teixeira, in the Tower Group report, contended that the SBS customer module was first implemented by GMAC, a division of General Motors, in 1990, and involved nearly five times as many accounts. He further explained that the failure of SBS was due to continual changes in the banking industry and the growth of banks such as Norwest and Bank One and was not due to technical risks. Mr. Teixeira attributed innovative qualities in the SBS project to the building of the customer module as the centerpiece of an integrated banking system and the use of the PACBASE development environment. A July 1993 Tower Group report was more generous in describing the SBS project, noting it as a "monumental effort * * * based on providing a bank with state of the art technology". The July 1993 report also stated that the customer module had the ability to contain up to 12,000 pieces of data, or six times more than any other system available.

In both the Davis and Teixeira reports, it appears that respondent's experts found that Norwest was not engaged in qualified research in the SBS project because the majority of the work was performed by EDS. This is only relevant, however, if EDS was not performing contract research on behalf of Norwest. Respondent contends that EDS did not perform contract research.

We find that together Norwest and EDS engaged in qualified research in the SBS project with respect to the customer module. See sec. 1.41-2(e)(5), Examples (5) and (6), Income Tax Regs. (allowing R&E credit where in-house and contract research are performed together).[62] In making this finding, we are influenced by the writings of respondent's expert, the Tower Group, which were prepared before its involvement in this case and are a part of the record.

The SBS project was a massive effort at developing an integrated banking system that could interact with several other systems and handle a tremendous volume of data. Although some integrated systems existed in 1986 (about 25 percent of the top 100 U.S. banks had such systems according to Mr. Teixeira), none of them could handle the volume of data in the SBS system, nor were they customer based (rather, they were product or account based).

---

[62] Although petitioner claimed that it was a "development partner" with EDS in the development of the SBS system, neither petitioner nor respondent suggests that Norwest and EDS were engaged in a partnership that would implicate the rules under sec. 1.41-2(a)(4), Income Tax Regs.

The Tower Group expected that the customer-centric module of SBS would become the "core foundation around which a bank can integrate all of its customer information".

The development of the deposit and credit modules, however, does not constitute qualified research. The record is void of any testimony or evidence (other than reports dated after 1991, after the timeframe in issue) from which we can evaluate the nature of the work performed on the deposit and credit modules. We note however Mr. Teixeira's indication that the deposit and credit modules provided "no significant functional innovations to the industry".

The SBS customer module project involved the discovery of information which was technological in nature and expanded principles of computer science--namely, the ability to create a customer-based system that could integrate with other banking systems and handle large volumes of data. In this regard, we note that qualified research for purposes of section 41 is not limited to the development of new technology but also encompasses the use of existing technology in new and dynamic ways.

There is no doubt that SBS provided a new or improved business component of Norwest in terms of customer service and growth opportunities. The Tower Group described the GMAC customer module system as providing "quicker, more efficient underwriting, and a faster, accurate understanding of changes in automobile purchase behavior". The sheer volume capacity of SBS, as compared with the

preexisting banking systems, resulted in a new or improved business component at Norwest.

The SBS project went through a process of experimentation. EDS, Bank One, and Norwest personnel met regularly to review and critique the EDS technical development of SBS and recommend changes to the system's design. Specific concerns were raised, for example, with respect to the volume capacity of the database system, DB2, which was one of the critical elements of SBS. In addition, concerns were expressed as to the user architecture and the use of the PACBASE tool. Alternatives were proposed and discussed for each of these concerns (although apparently not adopted by EDS). These alternative suggestions, together with Mr. Phillips' claim (to Dr. McDermott) of a 50-percent chance of failure of the SBS project, buttress a finding that there was uncertainty at the outset of the SBS project. Other issues concerning functionality, through data modeling, were also discussed. These issues and others were tested extensively at Norwest with respect to the customer module over a 3-year period, resulting in the discovery of hundreds of problems, some of which were attributable to poor technical design and others to poor programming of the source code. This process of developing, reviewing, testing, and analyzing the various approaches proposed by EDS constituted at least 80 percent of the development of the SBS system and satisfies the formal standards of experimentation sought by Congress.

In concluding that the process of experimentation test is satisfied, we believe the activities in the development of the SBS system should be examined in toto, rather than separately. The activities were interdependent and built on each other. Separately, the activities were of no utility.

The SBS system was an innovative effort that had the potential to result in substantial efficiency and significant economic benefit to Norwest. McKinsey & Co., Inc. (McKinsey), which was hired by EDS to conduct a valuation of the SBS system for Norwest in September 1987, found that SBS would provide better cross-selling of products to customers, improved relationship pricing, and the completion of backlogged development projects. McKinsey concluded that these improvements would result in an increase in pretax profits for Norwest of approximately $24 million for its retail banking business. Further, it was believed that Norwest would save approximately $6 million in data processing expenses as a result of the use of fourth-generation technology tools, integration, and business model design. Finally, another $1 million in pretax savings would result from better collection procedures and personnel productivity. The McKinsey report stated that "the system's major value will be allowing banks to develop innovative products and target their customers more effectively. Maximizing these capabilities gives a participating bank a clear advantage over its bank and `near bank' competitors." McKinsey's findings were unchallenged by respondent and lead us to the

inescapable conclusion that Norwest would have obtained substantial competitive and fiscal benefits from the SBS customer module had it and the other modules been successfully implemented.

The SBS system also involved significant economic risk to the three entities that participated in the development of the project. The Tower Group reported in 1993 that an estimated $125 million had been spent at that time by the three participants. Further, substantial uncertainty existed because of technical risk that those resources would not be recovered within a reasonable period of time. In 1993, the Tower Group characterized EDS' efforts as "venturing into new territory for both itself * * * and its prospects". In 1994, the Tower Group, in looking back on the prospect of failure of the SBS system, stated: "The world of banking is moving faster than EDS and its two development banks could move SBS into production. This fact is a commentary on the time and resources required to implement a complex new system in a large bank environment." A January 1989 internal audit at Norwest also revealed strong concern about the ability to complete the SBS project: "In a project of this complexity and magnitude, involving major new concepts, original database designs and state-of-the-art technological application, by its very nature will result in unforeseen problems."

Dr. Davis stated in his initial report that size or complexity of a project can create technical risk. We agree and find that SBS falls within that category. The development of the customer-based

system, the creation of a large volume capacity, and the implementation and integration of new systems into a large banking environment each separately may not be technically difficult. But together these activities pose serious technical challenges. We believe SBS was a technically risky venture for the participants, as the ability to accomplish all three goals and recover the costs within a reasonable period of time was substantially uncertain. This is confirmed, in hindsight, by the Tower Group in an American Banker article in which respondent's expert discussed the reason why systems such as SBS have failed:

> As the size of core application systems has grown, it has become impossible for teams of mere mortals to understand and control the almost infinite number of details. The world won't hold still while every detail is isolated, structured, and efficiently related to every other detail. The number of relationships goes up exponentially with the number of details - and so does the number of points where an error can occur. The net result: a high likelihood of failure.

Teixeira, "Why Big Bank Core-Processing Systems Will No Longer Be Built", Am. Banker (hereinafter Am. Banker article) 7A (Apr. 11, 1994).

Finally, the SBS customer module built for GMAC could not have entirely helped in the development of the modules for Norwest and Bank One because the module for GMAC was not operational until 1990--whereas the rollout for Norwest began in 1989. Further, systems such as Anacomp, which was acquired by EDS, and others developed by EDS' competitors, did not offer the functionality or volume capacity that was critical to SBS' potential success. Thus,

no commercially available products were available at the time Norwest entered into the SBS project with EDS and Bank One.

In sum, the SBS customer module project satisfies all seven tests for qualified research. The project, at the time, represented a concerted effort at advancing the state of technology in the field of computer science, pushing existing technology to new heights. Its ultimate failure, at least at Norwest, does not diminish its contribution to the field but only demonstrates the rapid pace of technological advancement in the industry.

Respondent contends that even assuming arguendo that the SBS customer module project satisfied the definition of qualified research in section 41(d), the work performed by EDS did not satisfy the requirements of contract research on behalf of Norwest, and thus Norwest is not entitled to the R&E credit. Specifically, respondent claims that Norwest did not retain a right to the research results, and further, that the payments made to EDS were contingent on the success of the research. We disagree.

Section 41(b)(3) allows the R&E credit for 65 percent of any expenses paid or incurred by the taxpayer to any person (other than an employee of the taxpayer) for qualified research. Section 1.41-2(e)(2), Income Tax Regs., requires that the contractor of the qualified research perform such research "on behalf of the taxpayer" and that the payments not be contingent on the success of the research. Section 1.41-2(e)(3), Income Tax Regs., further states: "Qualified research is performed on behalf of the taxpayer

if the taxpayer has a right to the research results. Qualified research can be performed on behalf of the taxpayer notwithstanding the fact that the taxpayer does not have exclusive rights to the results."

Under the Participant License Agreement between Norwest and EDS executed on December 16, 1986, the SBS system and each release of that system were the property of EDS, except that Norwest was entitled to a "perpetual, nontransferable, nonexclusive, and * * * royalty-free license to use the [SBS system]". The license to use the SBS system was subject to several conditions, and the failure of Norwest to adhere to those conditions gave EDS the right to revoke the license if the noncompliance was not cured within 10 days. The license was later extended to permit Norwest to use SBS in providing banking services to nonaffiliated financial institutions.

Respondent contends that this limited license to use the software is not sufficient as a "right to the research results" to warrant the treatment of EDS as a contractor of qualified research. Respondent differentiates between research results and the final product and argues that had EDS failed to deliver a product, Norwest would have been entitled to nothing. Respondent asserts that if research results and the final product are synonymous, then the term "research results" would be rendered meaningless and every time a person purchased a product, the purchaser might be entitled to a qualifying research credit under section 41.

Petitioner argues that Norwest, Bank One, and EDS were "development partners" in the SBS project, noting that this was how Mr. Teixeira characterized the relationship of the three entities in the Am. Banker article. Further, petitioner contends that the language of the regulations anticipates no more than this type of relationship where each entity shares in responsibilities and the rights to use the software upon completion of development. We agree with petitioner that Norwest had a right to the research results within the meaning of section 1.41-2(e)(2)(ii) and (3), Income Tax Regs.

The regulations provide that a contractor may obtain the R&E credit if it retains "substantial rights" in the research. Sec. 1.41-2(a)(3)(ii), Income Tax Regs. A taxpayer does not retain substantial rights in the research if the taxpayer must pay for the right to use the results of the research. Sec. 1.41-5(d)(3)(i), Income Tax Regs. (applying to the pre-1986 version of the R&E credit but cross-referenced by section 1.41-2(a)(3), Income Tax Regs.). We believe that the right to use the results of the research without paying for that right is at least a right to the research results as that term is applied in section 1.41-2(e)(2)(ii), Income Tax Regs.--although it may or may not constitute "substantial rights in the research" within the purview of the regulations.

Norwest retained a right to the research results in this case through its perpetual license from EDS, the contractor, to use the

SBS system without paying any royalties. This is sufficient to satisfy the requirements of section 1.41-2(e)(2)(ii), Income Tax Regs.

Respondent contends that Norwest's payments to EDS after May 25, 1987, were contingent on the success of the SBS system. Under the Participant Agreement between Norwest and Bank One, Norwest was required to pay $7 million to Bank One for its participation in the SBS project. The payments were made in installments, beginning with an initial payment of $694,440 upon execution of the agreement, followed by monthly payments of $138,890 until February 25, 1988, and followed by monthly payments of $250,000 until July 25, 1989. (The last two payments were deferred pursuant to an amended agreement between the parties because of delays by EDS in releasing SBS.) However, under the terms of the agreement, Norwest could stop making payments and terminate the agreement at selected times during the interim periods in which development milestones of the SBS project were being completed. The first development milestone was completed on May 25, 1987.

Petitioner argues that although Norwest was not obligated to make payments to Bank One (and EDS) after each interim development milestone period, there was never an agreement that any amounts that it chose to pay were recoverable if the research was not successful. We agree with petitioner.

As to respondent's argument that the payments made to EDS were contingent on the success of the research, we note that the

participant agreements between Norwest and Bank One, and Norwest and EDS, are void of any reference to such a contingency. Had Norwest been dissatisfied with a development milestone, it could have terminated the agreements at that point and taken the SBS release as it then existed--but Norwest could not have recovered the payments it had already made. Thus, the requirements of section 1.41-2(e)(2)(iii), Income Tax Regs., are satisfied.

We hold that the expenses paid or incurred by Norwest in the development of the SBS customer module constitute qualified research under section 41. The only remaining issue then is which expenses of EDS and Norwest in the SBS customer module project constitute qualified research. In this regard, respondent contends that Norwest's installation and customization activities, and the subsequent testing after it received the releases of SBS from EDS, do not constitute qualified research. Respondent specifically relies on section 41(d)(4)(D)(v), which excludes from qualified research any routine or ordinary testing or inspection for quality control. Respondent asserts that installation and customization, and the testing activities related to those processes, are routine or ordinary testing procedures in the software field and thus must be excluded from qualified research.

While respondent's characterizations may be accurate in some contexts, they are not accurate in the context of the SBS project. The customer module's unique feature was its ability to interact with other modules to permit the easy sharing of data and

information. Thus, the installation process, including the interfacing of SBS with Norwest's existing deposit and credit modules and other systems, and the subsequent testing, was critical to the success of SBS. This was all part of the research process. It was far from routine. Mr. Teixeira, in a book published in 1990, indicated that installation and conversion of large integrated software systems like SBS "can be risky".

We agree with respondent, however, that customization activities relating to SBS by Norwest after delivery by EDS that were unrelated to the installation and interfacing activities are not qualified research as those activities relate only to style, taste, cosmetic, or seasonal design factors. See sec. 41(d)(3)(B). We also agree with respondent that any activities performed by Norwest after the first SBS release was installed at each bank is not qualified research. Section 41(d)(4)(A) excludes from qualified research any research conducted after the beginning of commercial production of the business component. Although Congress anticipated that some postproduction activities, including internal use software activities after the installation of commercially available software, could be treated as qualified research, H. Conf. Rept. 99-841 (Vol. II), at II-73, II-74 n.4 (1986), 1986-3 C.B. (Vol. 4) 1, 73, 74 n.4, Norwest has not shown that the subsequent releases of SBS by EDS involved qualified research by EDS or Norwest, Rule 142(a).

To summarize, we conclude that all expenses paid or incurred by Norwest in the development of the SBS customer module constitute qualified research through the first deployment of the customer module to each of Norwest's banks, and that expenses relating to customization activities by Norwest after receipt of the first SBS release do not.

B. Trust TU

Dr. McDermott found that the NTS staff used "considerable ingenuity" in keeping the Trust TU system running as long as possible. He stated that even though the project was eventually abandoned, had Norwest's Trust TU system failed before new technology (the Compass system) was available, Norwest would have incurred significantly greater costs in running its trust department. Thus, he concluded that the project was a "mixed success".

Dr. Davis characterized this project generally as maintenance and enhancement of the source code and claimed that the efforts to increase volume and efficiency were "one of the most common, routine tasks in information processing". He found that the volumes sought by NTS were well within the then state of the art.

Mr. Teixeira also characterized the Trust TU project as generally maintenance and enhancement efforts. He found that the changes did not add any new functionality that was not already available at any other trust department. In this regard, he noted that the volume sought by NTS was well below the volume already

processed by the larger trust departments such as at State Street Bank and Bank of New York.

We agree with respondent that the Trust TU activities do not constitute qualified research. None of the activities engaged in by NTS in this project involved anything more than "cookbook" approaches to software development and the basic "skilled practice" of computer programmers. Indeed, Dr. McDermott admitted as much when he stated that the changes to Trust TU "required the application of basic principles of software engineering and testing" and that "This is not cutting-edge science, but it falls exactly within the bounds of computer science as it is taught in textbooks."

Routine software development is characterized by the use of cookbook approaches, as described by Dr. McDermott, in which at the outset the ability to accomplish a task is known because of the presence of skilled practice and known methodologies,[63] even though

---

[63] We accept petitioner's assertion that standard methodologies (such as Norwest's SDM) are employed because researchers find that they are the best way to discover new information. These are not the types of methodologies we are concerned with in qualified research; the key is whether the practices and methodologies provide the researchers with the known capabilities (e.g., formulas) for accomplishing the task-- and in routine software development, that is often the case.
We think Dr. Davis' testimony on this subject is instructive:

> A   One of the things that I think is worth
> distinguishing is the notion of research
> activity from what I'll call the skilled
> practice.

(continued...)

the exact benefits to be derived are in doubt. Cookbook approaches
to software development preclude any finding of a discovery of
information that is technological in nature, or a process of
experimentation. Cookbook approaches to software development do
not result in new knowledge about the principles of computer
science or technical uncertainty that requires the consideration of
alternative hypotheses. In our opinion, Congress did not intend
cookbook approaches to software development to come within the
bounds of section 41 when it excluded from the R&E credit the
duplication of existing business components, or routine data
collection or testing. See sec. 41(d)(4)(B), (D)(iv) and (v); H.
Rept. 99-426, at 182 (1985), 1986-3 C.B. (Vol. 2) 1, 182; cf.

---

[63](...continued)
        Anybody engaged in, well, almost any
    profession -- but let me make it in the
    engineering profession for the moment --
    accumulates a body of skilled practice over
    time. These are things that you know how to
    do because you're in the business.

        Some of those things are difficult to
    do. Some of them require a substantial
    amount of skilled practice to do them. So to
    say something is not research, or even to
    call it routine software construction is no
    way to denigrate it or to say it doesn't take
    a substantial body of skill to do.

        What it says is that, in the community
    of people who do this kind of thing, the
    knowledge of how to do it is out there, as
    opposed to it has to be discovered or
    revealed in some fashion. Okay? It's what
    you would expect a competent professional in
    the field to know and be to [sic] able to do.

Mayrath v. Commissioner, 41 T.C. 582, 590-591 (1964), affd. 357
F.2d 209 (5th Cir. 1966) (rejecting section 174 deduction for
development of experimental house which involved the use of
standard construction principles).

The Trust TU project was not one that expanded or refined the
principles of computer science. Nothing new was accomplished. The
capacity to develop the project was already a part of the skilled
practice of NTS and had been achieved elsewhere. To the extent
risks were present in this project, they related solely to business
risk, not to technical risk that involved substantial uncertainty.

C. Success

Dr. McDermott described the technical efforts of developing
the Success equipment leasing system as "getting TPF [the
processor] to do things slightly beyond what it was designed to do"
(although later in the same report he indicated that the Success
project pushed the TPF system "well beyond its known limits"). He
found serious questions regarding whether the "complex data
structures" required by Success could be handled within the TPF
framework, and the solutions developed by NFISG "made a definite
contribution to computer science". He further found that Success
provided a substantial improvement over its predecessor, Infolease,
in terms of speed, functionality, and data integrity.

Dr. Davis found that the development of the Success system
involved nothing more than routine practice, and that none of the
activities required the use of a process of experimentation. He

also found that no significant degree of technical risk was present in any of the various activities.

Mr. Teixeira reported that the Success system offered no new or unique features to the banking industry, and that although Success was an improvement from its predecessor, it did not provide industry-leading capabilities. Mr. Teixeira noted that although some functions, such as the use of e-mail and the ENFORM program to generate reports, improved efficiency, NTS did not discover any new or unique ways to exploit those functions. He concluded that although no commercially available software could provide exactly the functionality needed by Norwest, there was no doubt about the ability to develop viable equipment leasing software in Norwest's environment.

We agree with respondent that the development of the Success system does not constitute qualified research. All three experts noted the lack of documentary evidence for the Success project, and we note that the record is specifically lacking with respect to the technical risks involved. It appears that the only project activity which may have involved technical risk was with respect to the ability to develop Success on the TPF platform. However, Dr. McDermott was vague about the nature of the technical risk in this regard, and Mr. Teixeira reported that this was a small part of the overall project. We also note the testimony of Dan Franker, manager of the Success project, who indicated that he never encountered insurmountable issues with the project and that he

anticipated that the project could be accomplished with the available resources. Further, the major risks to the project appear to have related to the ability to develop the software within tight time constraints--a business risk, not a technical risk. In sum, the project simply falls short of one that could be characterized as venturing into uncharted territory.

D. General Ledger

Dr. McDermott identified the "shadow files" system as the major technical challenge of the general ledger project, allowing virtual access to real-time data while continuing to run batch processing only once per day. He stated that the logical problems that had to be solved for this system were "complex and hard to get right".

Dr. Davis described the changes to the general ledger system as "remarkably small-scale, routine modifications". He stated that the issues raised by the performance problems were routine, and that the solutions were well known and could be accomplished through good coding practices and established methodologies.

Mr. Teixeira stated that the general ledger system from FCS, which Norwest purchased, was used at over 30 of the top 100 U.S. banks during the relevant period. Moreover, he stated that the work performed by NTS did not provide any significant technical innovations beyond those which the vendor had designed and delivered.

We agree with respondent that the general ledger project activities do not constitute qualified research. The only activity that could have involved technical risk was the shadow files system. However, we accept Mr. Teixeira's assertion that the development of a shadow files system was common at large U.S. banks and required only the following of known methodologies and approaches--routine software development that does not expand or refine the principles of computer science and does not involve the engagement of technical risk.

E. Money Transfer

Dr. McDermott opined that the money transfer system project involved qualified research due to the financial and security risks involved in complying with the regulatory changes imposed by the Federal Reserve and the expected growth in the industry. He wrote of NTS' efforts of extensive testing to assure the elimination of software bugs. Finally, he noted that NTS rewrote as much as 50 percent of the source code to the system to provide the functionality needed by Norwest.

Dr. Davis admitted lack of knowledge of wire transfer systems; he deferred to Mr. Teixeira for elaboration. He commented that the development of systems through the use of COBOL language and the Tandem computer (which were apparently used in the money transfer system) do not constitute the discovery of information that is technological in nature or which could involve a process of experimentation.

Mr. Teixeira reported that by late 1986, the money transfer system used by Norwest, MoneyNet, was being used by 46 of the top 100 U.S. banks. He stated that Norwest's modifications to the system to satisfy regulatory and business requirements were consistent with the changes made by other users of the same system. He found that although the technology may have been new to Norwest, it was not new to the industry.

Although a taxpayer's making changes that are being made concurrently by other users of a system does not mean that the taxpayer cannot be engaged in qualified research, Norwest has not demonstrated that any of its work on its money transfer system involved technical risks or a process of experimentation--and thus the activities do not constitute qualified research. There were obvious business risks involved with this project, e.g., failing to implement critical security measures that ensured the safe and sure transfer of funds, but there was no evidence of uncertainty about the ability to complete the project. The project did not involve alternative designs or hypotheses but merely required conducting good coding and eliminating bugs through testing--issues resolved through cookbook approaches and skilled practice, not research and experimentation.

F. Cyborg Payroll

Dr. McDermott strained to conclude in his rebuttal report that the Cyborg payroll system constituted qualified research. He insisted "it is possible that the process was one of mechanical

trial and error rather than true research." However, he refused to concede that this project did not qualify. Elsewhere, Dr. McDermott appeared to regard as significant the use of the arcane report writer language of the Cyborg payroll system, as well as the "heroic efforts" of NTS to sustain this "increasingly inadequate" system.

Dr. Davis classified the task of maintaining and enhancing the payroll system as "one of the oldest and most familiar in information technology". He found that everything performed by NTS was routine and well within industry practice.

Mr. Teixeira also found NTS' efforts entirely routine and noted that the functionality added to Cyborg already existed at every other major U.S. bank.

The Cyborg payroll system activities clearly do not fall within the realm of qualified research. Dr. McDermott stated that the key issue "was how long the system could be made to survive". Cyborg was an outdated system. It is evident that the goal of NTS was not to advance the principles of computer science, but rather was to maintain a 1970's system running into the early 1990's. It was this type of activity that Congress had in mind when it sought to narrow the definition of qualified research and expressed its concern that taxpayers were claiming the R&E credit even though they were not engaged in high technology activities. See S. Rept. 99-313, at 694-695 (1986), 1986-3 C.B. (Vol. 3) 1, 694-695; H. Rept. 99-426, at 178 (1985), 1986-3 C.B. (Vol. 2) 1, 178. Heroic

as the efforts of NTS may have been, such efforts involved nothing more than the skilled practice of those in the information technology industry.

G.   Trust Payment

Dr. McDermott identified two complex issues in developing the Trust Payment system which related to performance and reliability. These involved "optimizing" the programs to make them run faster and more efficiently and attempting to avoid "locking" of the system when too many users were accessing data. He found that the changes to this system were "more difficult than usual" because small changes could disrupt the logical structure of a larger system.

Dr. Davis stated that the task of developing the Trust Payment system, and the technology used, were familiar and well established in the industry. He found nothing uncommon about interfacing a new system to other older systems. Dr. Davis conceded that uncertainty over NTS' ability to develop a system to meet Norwest's requirements may have existed, but he denied that any new principles of computer science were discovered through the routine use of standard tools that had been around for many years. Finally, he found that the problems and solutions encountered by Norwest in the development of this system were routine.

Mr. Teixeira, as in the case of the Trust TU system, made reference to the fact that Norwest was a mid-tier trust service provider, and that other trust service providers processed as many

accounts as Norwest. Indeed, some providers processed as many as

five times more. Mr. Teixeira found that Norwest's development of

the Trust Payment system did not result in any new or unique

functions or technology to the industry, and that commercially

available software offered comparable functionality. He concluded

that the technologies used in developing the system were well

established in the banking industry and were familiar to Norwest

personnel.

We agree with respondent that the development of the Trust

Payment system does not constitute qualified research. Although

technical risks arose in the development process, those risks were

limited and did not involve substantial uncertainty that Norwest's

investment could be recovered within a reasonable period of time.

The technical risks were clearly solvable: the only issue to

Norwest was whether they could be solved in time so as to maintain

or advance Norwest's competitive standing in the trust service

provider industry. Indeed, Dr. McDermott admitted this when he

reported:

> The key point to repeat here is that the risk to Norwest
> was that a project would be <u>delayed</u> because of technical
> uncertainties. Even if a project could clearly be done
> eventually, the results might be useless if they came too
> late. The cost to Norwest would be wasted development
> effort, inefficiency in using old solutions, and
> backwardness with respect to their competitors. It seems
> clear that in the case of * * * [the Trust Payment
> system] these risks were real, and actually materialized,
> although not to the extent that the project failed
> altogether.

A "reasonable period of time" for the development of a software system does not relate to self-imposed business time constraints, but rather to the reasonable time of those in the field of computer science.

H. Debit Card

Dr. McDermott found that the Debit Card project had a "significant chance of failure" due to software engineering, one which the project team placed as high as 50 percent. However, Dr. McDermott admitted that "With all resource constraints removed, there is little doubt the project would have eventually succeeded." He indicated that the software engineering question was whether it was "possible to make hundreds of modifications to several existing systems in the time allotted". In the end, the entire success of the project, i.e., becoming the first bank in Norwest's market to deliver a debit card product, was dependent upon Visa's ability to develop the appropriate interface with Norwest's existing ATM system on time. "This was out of Norwest's control, and was a risk the Norwest programmers could do nothing about." Dr. McDermott concluded in his rebuttal report that there was no high degree of uncertainty about the particular algorithms used in the project, and that the only uncertainty was the ability to complete the project in the short time provided.

Dr. Davis concurred that the only risks in this project related to business or economic risks, not technical ones. Further, he found that the processes engaged in by Norwest were

routine and consistent with standard practice in the computer science field.

Mr. Teixeira reported that debit cards were common by 1989, and several banks had transaction volumes significantly larger than Norwest's. Further, by 1990, 673 banks were issuing Visa logo debit cards. He also found that Norwest's experience with ATM's, which Norwest used to operate the debit card system, provided a solid understanding of the technical issues involved, including interfacing, card issuance, and capacity planning. Additionally, Mr. Teixeira stated that Norwest's activities in interfacing the new debit card system with its other systems were consistent with the deployment of the products by other institutions. Finally, Mr. Teixeira rejected any claim by Norwest that technical alternatives were considered in the process of developing the debit card system and insisted that the alternatives related only to business issues.

We agree with respondent that the development of the Debit Card system does not constitute qualified research. Dr. McDermott's own findings that the project's only risk was its ability to be completed and deployed before those of Norwest's competitors undermines any claim to the R&E credit. None of the experts reveal any technical risks, and it is apparent that the debit card was a fairly common product (nearly 40 percent of the top 75 banks had the debit card by late 1989) by the time Norwest entered the market. No new principles of computer science were discovered by this project, and although alternative approaches

existed (e.g., credit card processor versus direct access approach), they related entirely to business concerns and not technical risks.

Conclusion

In summary, we hold that the development of the SBS customer module constituted qualified research under section 41, to the extent indicated <u>supra</u>, and that the remaining seven internal use software projects failed to satisfy the seven tests required to obtain the R&E credit.

To reflect the foregoing,

<u>Appropriate orders</u>

<u>will be issued</u>.