Guidance on Eligibility of Software Projects for the SR&ED Tax Credits and Developing and Documenting Claims

Table of Contents

Part 1. Introductory Comments:

Background and Comment on the Use of this Guidance

The Scientific Research and Experimental Development (SR&ED) Program of the Canada Customs and Revenue Agency (the CCRA, formerly Revenue Canada) has joined with industry to build a partnership to improve the program. The following guidance is a product of this partnership. Specifically, two key elements of the CCRA's efforts to improve how the program works for the software industry are the development through the private sector of:

- additional guidance on the types of software development work that are eligible for the credits and which ones are not (i.e., clarifications); and
- additional guidance on how to properly prepare project descriptions and what documentation needs to be maintained to support a claim.

Development of this guidance was led by the **CATA*Alliance*** in association with the **Information Technology Association of Canada (ITAC); the Information Technology Association of Nova Scotia (ITANS); the B.C. Technology Industries Association of Canada (TIA); the Vancouver Island Advanced Technology Centre (VIATeC); the Canadian Information Processing Society (CIPS); and the CCRA.** Technologists, software development managers, consultants and tax practitioners from across Canada participated in the consultations that produced this guidance.
The consultative process initially involved a workshop of over 40 individuals from the community who set the priorities for the issues to be addressed in the guidance and how the guidance was to be developed. A consultant team of leading software technologists and policy experts served as the scribes for the project. Each individual piece was developed through a series of drafts, circulated initially to the workshop participants for their input and direction. Earlier drafts were posted for public comment and comment by the CCRA's software specialists. These drafts were finalized in their present form by the consultant team, based on the direction of the industry's Software Sectoral Working Group that provided oversight for the project.

Comments concerning the use of this guidance

This guidance is intended to be used in the context of joint CCRA / industry training on how the program works in the software sector. The guidance is not meant to stand alone. The CCRA has provided input throughout the development of the work and has generally agreed with the concepts. Nevertheless, the CCRA has expressed concern about several points. Specifically, the guidance on how to set the line for eligibility and how a business's context influences this decision should not be interpreted to indicate that the incentives are to promote training or promote learning outside the context of work primarily aimed at advancing a software technology or computer science. Learning how to use a specific development tool, etc., is not the objective of these tax credits per se.
Finally, the CCRA has expressed concern that on its own, and without the background information that would be provided in the training, that the examples could be misinterpreted. For this reason, it has been agreed that the CCRA's specialists are to develop

supplementary examples to be used in the joint CCRA/industry training and added to this guidance when the examples are completed by CCRA.
The relationship between the criteria of technological uncertainty and advancement was a critical consideration in the consultations. To many software engineers, the existence of a demonstrable technological uncertainty associated with the need for experimentation was enough to demonstrate eligibility. In the end, this approach, like the attempt in Information Circular IC97-1 to distinguish between what is standard practice and what is eligible work on the basis of the existence of experimentation, was seen as too imprecise. It was concluded that if a true technological uncertainty existed in the sense used by the software engineers, one should always be able to show that an advancement was being sought.
Finally, the Software Sectoral Working Group has expressed the concern that the material on documentation might be misused to establish unreasonable expectations. In this respect, the Working Group and the CCRA caution any user of this guidance to read closely the advice on sufficiency which cautions against this practice.

Partners' role

The lead for this work was with the business community who took on the task of developing the guidance. The CCRA provided support for this effort in the form of funding, and advice on the legislative and administrative constraints which set the boundaries for the clarifications. By administrative constraints we mean the development of "clarifications" that themselves can be interpreted in ways that result in confusion or in the abuse of the system, i.e., unintended results of proposed clarifications. The CCRA's inputs were coordinated through the National Technology Sector Specialists for Software in collaboration with the Manager, Client Liaison at the CCRA and included advice on the following:

- legislative boundaries - i.e., what the law allows for and what is clearly out or beyond the documented policy intent of Finance;
- administrative constraints;
- field experience - i.e. the CCRA's experience in terms of examples of eligible and non-eligible projects, well prepared claims, good documentation processes that work, those that do not work, etc;
- potential problems - e.g., imprecisions in definitions and practices;
- the implementation process - i.e., what is feasible or not in terms of changes to the program, and the time frame it may take

Part 2. Guidance on Issues Associated with Eligibility

Guidance to the Taxpayer

This document is written as general guidance to the taxpayer on how to decide whether or not to make an SR&ED claim for software development. To provide this guidance we first address the likelihood of project eligibility given the commercial context of the software development project. We next define the three criteria for eligibility and provide guidance as to how to identify convincing evidence of eligibility. Thirdly, we provide some guidance as to how to best package this evidence within the project description required by the T661. Finally, we address specific issues of eligibility and provide some example.

We hope that through the following we can impart a broad understanding of the SR&ED Program as it relates to software projects. We hope that we can assist the taxpayer in quickly determining which of his projects are clearly eligible and which are clearly ineligible. For those projects for which eligibility is still questionable in the taxpayer's mind, because of the nature of the project, the taxpayer should study this document in its entirety, read and understand IC97-1, IC86-4R3, IT-151R4, seek expert professional opinion or seek an advance ruling from CCRA.

This guidance is intended to reflect the incentive nature of the SR&ED Program which is to promote advancements in a company's technologies through experimental development and research. While some eligible training and on-the-job learning may be incidentally associated with seeking these advancements, training and on-the-job learning are not the objective of these incentives and will not be associated with eligible work on their own.

The following generalities are based upon extensive audit experience and should be helpful in separating those projects that clearly qualify from those that clearly do not qualify.

In general:

To prove eligibility the taxpayer must show that the project meets all three criteria for eligibility:

sought a Technological Advancement and hence an advancement in our understanding of the technologies embodied Technological Uncertainty had Technological Content.

The following comments on the likelihood of eligibility are intended to help developers know where eligible projects are most likely to be found. They should not be interpreted to suggest that every project in a likely area is eligible, or for that matter that any project is eligible or ineligible in any area. Any project for which the taxpayer can demonstrate satisfaction of all three requirements for eligibility qualifies as SR&ED despite the following comments regarding likelihood of eligibility. If the three criteria cannot be demonstrated they are not eligible.

Projects aimed at developing the initial release of saleable software which is "leading edge" in some technological way are often largely eligible. To develop software at the leading edge of today's technologies generally requires the developer to come up with new constructs, such as new architectures, algorithms or database management techniques (i.e., make Technological Advancements), and there are then specific uncertainties as to the viability of these (i.e., Technological Uncertainty). If the software's competitive edge stems from advance in an area other than technology, such as business management, or improvements in financial management techniques, the project is unlikely to be eligible. Almost any software developed for sale is developed systematically and the uncertainties are systematically resolved (i.e., Technical Content).

Nevertheless, since some saleable products are simply the adaptation of established software or known developmental techniques to new situations the taxpayer must specifically identify the Technological Advancement and the Technological Uncertainties to successfully make a claim. He must have some evidence that shows that the development systematically approached problem identification, solution evaluation and testing. He should clearly

state at what point in time the project was complete i.e. when the Technological Uncertainties were resolved and a Technological Advancement was consistently demonstrable or when the project was abandoned. Simply claiming that he was "developing new saleable software" does not validate his claim. Similarly, new or novel functions or features that the software provides to end-users do not alone establish eligibility, because novel features and functions can be developed to provide a commercial advancement without attempting a Technological Advancement or encountering Technological Uncertainties.

Occasionally new saleable software is developed using combinations of constructs and tools that were generally known or were already known to the taxpayer from previous projects. This can frequently be the case when the new saleable software automates a well-understood or prescribed function in record keeping, data formatting or calculation. If this is the case, the project is not eligible unless the taxpayer can clearly identify an eligible Technological Advancement and show that the project uncertainties were Technological Uncertainties rather than routine design uncertainties.

Development of software for in-house use as the technological foundation of the taxpayer's business is often largely eligible, when the software contains new constructs (Technological Advancements) and/or new solutions to technological challenges (Technological Uncertainties).

Nevertheless, the taxpayer must specifically identify the Technological Advancement and the Technological Uncertainty to successfully claim the work for this is not always the case. He must show that the development was systematic. Simply claiming the he was "developing software for in-house use as part of the technological foundation of my business" does not validate his claim. Similarly, new or novel functions or features that the software provides to end-users' do not of themselves establish eligibility.

The development of feature extensions to proven software may not qualify. Such extensions are often developed within the architecture of the original package using the tools proven during the development of that package (no Technological Advancement and only routine design uncertainties).

Nevertheless, as is sometimes the case, when the new features challenge the original architecture or system performance in ways which cause Technological Uncertainties, the taxpayer may also be able to show Technological Advancement and thus qualify the project.

Software developed for most in-house managerial reporting and efficiency support uses (accounting, performance reporting, shop management, industrial control) is often ineligible. This is because there is generally little doubt (only routine design uncertainty rather than Technological Uncertainty) that the specification can be met using available tools and well-known constructs in reasonably proven combinations or adaptations of them. Qualified professionals can usually determine the methods and solutions to be employed on such a project through discussion alone or through simple analyses (no Technological Advancement).

Nevertheless, if the in-house managerial reporting and efficiency support software departs from constructs already part of the industry's or the claimant's knowledge base, in ways that cause Technological Uncertainties, the taxpayer may also be able to show Technological Advancement and thus qualify the project.

In the broadest sense, the taxpayer can interpret the three essential criteria as follows:

Technological Advancement

The taxpayer must identify the new software construct(s), architecture or technique(s) sought or developed within the project which advances his understanding of information technology or computer science. The advancement need not be large.

Hint: As a means to identify the advancement(s), the taxpayer might try to identify that issue(s) which a member of the project's technical team should be willing to present as a "technological advancement" to an informal gathering of a group of peers. The taxpayer does not have to actually make such a presentation.

As a means of characterizing the Advancement for the purpose of making a claim, the taxpayer might identify the technological reason why his architecture or technique was not used before. How does it compare with earlier solutions or with the current solution of a competitor? What earlier technical constraint has been overcome?

Note: Simply claiming to have developed the first or best software suite for a given purpose does not in itself prove that the taxpayer has made a **technological advancement**. A new and unique software suite can be built using only well known combinations of constructs, tools and methods without technological advancement. This is analogous to designing and building a unique and complex office building without making any advancements in the field of civil engineering.

Note that an advancement in technology can rarely be described by listing software functions and features at an "end-user" level. Advances are typically made through innovation in software architectures, designs, algorithms, techniques or constructs.

Evidence of Technological Advancement could include credible third party literature or comparisons of the capabilities sought against those previously available from the taxpayer himself. As in a court of law, there are no rigid definitions of what constitutes credible evidence.

Technological Uncertainty

The taxpayer must show that there was at least one issue which raised a doubt as to whether the specification could be met within the technical constraints (not marketing, staffing or time-to-market constraints) OR there was at least one design issue or design alternative which had to be resolved through recorded investigation, analysis and/or prototype coding rather than through discussion alone.

Hint: In software development, technological uncertainty is most usually found in: resolving which is the best of several possible architectures, designs, algorithms, techniques or constructs; the extent to which performance can be achieved within defined hardware or software constraints (which might be commercially imposed); or in the interoperability of software entities which were not designed to work together.

There is always some uncertainty about anything. In software development, uncertainties that can be resolved through brief discussions with peers or simply through a few lines of analysis are routine design uncertainties rather than technological uncertainties. Likewise there is routinely the need to calibrate or optimize or clean up new software. These challenges are not technological uncertainties unless it can be shown there is a fundamental problem with the technologies that must be addressed.

As evidence, the taxpayer should retain the original analysis, design notes, relevant e-mail, engineering notebooks, the minutes of design meetings and test results from prototype coding (if he is claiming these activities) as proof of technological uncertainty. The taxpayer who generates none of these documents contemporaneously or who does not retain them, will have to rely upon publicly available evidence of technological uncertainty relevant to his specific claim. Persuasive evidence of this sort is usually difficult and costly to gather

Technological Content

The investigation must be systematic. Providing a brief chronology of the project may help to demonstrate this.

The chronology should highlight the design alternatives and the analysis or tests done to select between the alternatives. By showing the evolution of the design (confirmation of approach or changes in design approach or changes to the specification) resulting from each analysis, prototype or test, the taxpayer also demonstrates Technological Advancement.

The chronology should make reference to the Technological Uncertainties and show how activities were focused upon the resolution of these Uncertainties.

The chronology should show how the resolutions of the Uncertainties led to the Advancement sought or show that the sought Advancement could not be achieved within the constraints of the project.

Evidence that the chronology is accurate might include project schedules, relevant e-mail, dated meeting minutes or dated notebooks that record testing, observations and major design decisions.

General Guidance on Assembling an SR&ED Claim

The onus is on the taxpayer to prove project eligibility by providing evidence of that eligibility. In assessing the eligibility of a project, a CCRA Science Advisor has to decide whether or not eligible activity occurred and the point in time when eligibility started and ceased. He also has to decide which activities contributed to obtaining a consistently demonstrable Technological Advancement and to resolving the Technological Uncertainties. The stronger the evidence provided by the taxpayer, the easier it is for the CCRA Science Advisor to recommend eligibility.

The role of the project's technical description is to summarize the taxpayer's evidence in a form that will be readily understood by a CCRA Science Advisor and to provide references to available project documents generated during the project which support the claimed eligibility. If available documentation does not provide direct evidence of eligibility, the taxpayer's case is weakened.

It is CCRA policy that a Science Advisor provides his rational for either accepting or rejecting a project. If through an understanding of the taxpayer's software development environment, available documentation and interviews, a Science Advisor concludes that eligible work was done, he can recommend eligibility of that work, but he is under no obligation to do so given only weak documentation.

The project's technical description need not be long (2-4 pages is usually sufficient), but it should make the case for eligibility convincingly so as to minimize the taxpayer's costs (compliance costs) in defending his submission. The project's technical description is the taxpayer's stated "expert opinion" as to why the claimed project is eligible.

The taxpayer should try to name his project for its Technological Advancement rather than for its end use or product name. This helps both the taxpayer and the Science Advisor to focus on the technological issues rather than on the commercial issues.

The claimable costs of a project are, in concept, the costs of resolving the Technological Uncertainties and thus either achieving the Technological Advancement or determining to what extent it is achievable within the constraints of the project. The project is inclusive of the initial planning for the project and concludes when the Technological Uncertainties have been essentially resolved or the project has been abandoned because the objectives were unattainable at acceptable costs.

To be claimable, some knowledge must have been obtained from the project and thus a project which is only planned and for which execution was never started, has no qualifying costs. Several rules have been established as means of determining an acceptable estimate of the eligible costs which in the case of software development are largely the salaries and overheads associated with the development team (See IT-151R4).

We strongly urge taxpayers to ensure that time-keeping records or personnel assignment records created contemporaneously with the project support their SR&ED labour costs. If these records are kept on the basis of a commercial project, they should be constructed to allow the easy separation of ineligible costs and activities. Ineligible costs include those associated with activities conducted to support the product (e.g. bug fixes, routine feature additions, customer support) after the Technological Advancement had been achieved (or abandoned) and the Technological Uncertainties resolved.

Additionally, capital costs for new equipment that is used for most of its useful life on eligible projects qualifies as SR&ED Capital. (See IT151R4 for a detailed discussion of what we mean by "most" and for a discussion of the treatment of shared use equipment).

Specific Eligibility Issues

The starting point for the policy and practices on the interpretation of which software projects are eligible for the SR&ED tax credits is Information Circular IC86-4R3. This document provides general explanation. More specific information on the application of the general concepts to software developments is contained in Information Circular IC97-1. These guides can be obtained from CCRA's web site at http://www.ccra-adrc.gc.ca.

The following 9 issue sheets and the Guidance on Documentation have been developed by The Software Sectorial Working Group at the request of CCRA's Steering Committee for the SR&ED program. It is intended to provide additional clarification of problem areas that have been identified since IC97-1 was issued.

This material is to be the basis for formal training for CCRA's staff and business on how the tax credits work in the software sector. This training is a joint effort of the CCRA and business.

Software Issue Sheets

Issue Title:Technological Advancement

Background: "Scientific or Technological Advancement", usually referred to as simply "Technological Advancement" is defined in IC97-1 as, "the discovery by the claimant of technical knowledge that advances the understanding of scientific relations or technologies. (Sec. 3.1)"
IC86-4R3 explains, "The search carried out in the scientific research and experimental development activity must generate information that advances our understanding of scientific relations or technologies. In a business context, this means that when a new or improved product or process is created, it must embody a scientific or technological advancement in order to be eligible."
In Section 2.13 of IC86-4R3 we find, "The definition of 'experimental development' in subsection 2900(1) of the Regulations requires that work be undertaken to achieve technological advancement when creating new, or when slightly improving existing materials, devices, products, or processes. For an experimental development activity to be eligible in terms of scientific research and experimental development, it must conform to the spirit of the legislation; that is, it must seek to advance the claimant's technological knowledge base. The technological advance achieved has only to be slight."

Problem Statement: Project eligibility will be judged by an expert in the appropriate field of science/technology. Eligibility is therefore somewhat subjective, but in the face of strong evidence becomes less so. The onus is on the claimant to provide strong evidence of eligibility. At the same time, It is now to be practice of CCRA to try hard in their audits to understand the development process of the claimant and thus to understand where within the claimant's project documentation acceptable evidence might be found.
To some claimants unfamiliar with the SR&ED Program the term "project" implies a commercial development. The claimant then interprets the term "Technological Advancement" to simply mean "doing something new" such as developing a new product. The claimant then equates Technological Advancement to the successful completion of an "engineering task", a "commercial schedule" or to "commercial advancement/success". This is not a correct interpretation within the context of an SR&ED claim.
What is Technological Advancement, and how should it be determined and expressed in a commercial setting?

Discussion: A Technological Advancement exists when, starting from a declared technological base, a project investigates a technological solution that was initially unavailable to the claimant and was not generally known. The claimant thus gains technological knowledge through the course of the project.
The word "advancement" implies that there was a starting point from which the advance was made or at least sought. An advancement which is not open ended (and no SR&ED claim should extend indefinitely in time) then reasonably has a base from which to advance and a defined objective such as finding the technical means of achieving some objective.
A Technological Advancement brings new knowledge to the claimant. The word "knowledge" does not necessarily mean "publishable." The claimant gains knowledge through systematically making or attempting to make the Technological Advancement. He can use this new knowledge in developing software systems or embed this new knowledge in a new software system.
Often, but not always, the targeted forward objective can be defined quantitatively. The

objective may be measurable in some way or be otherwise demonstrable. The advancement sought might be to tighten a constraint on an earlier solution to a similar problem (e.g. move mean time to failure from 3 days to 30 days). The advancement could be to seek a solution within stated constraints (e.g. obtain a 3 second response from a software system on a specific platform).

Technological difficulties that provide the opportunity for a Technological Advancement may be identified at the start of a project or they may arise during the course of a project. The claimant should be able to cite the technological base from which he claims an advancement. A good reference for this base would be a credible third party statement of existing technological capabilities such as might be found in a contemporaneous technical paper, product statement or advertisement published in academic or trade literature. The reference must be credible evidence rather than boastful advertising. Otherwise, reference could be to the capabilities of a prior product of the claimant's as seen in the claimant's own literature or test results.

A good means of identifying the Technological Advancement is to ask the technical personnel involved what key point they would present as a Technological Advancement to an informally gathered group of peers external to their own environment. This question usually evokes the required technically oriented declaration of the Technological Advancement and avoids the definition of an advancement as simply, "We developed a new product." - which is unacceptable evidence of Technological Advancement. Of course, a negative technical result can also constitute a Technological Advancement.
It is rarely acceptable to equate functions and features of a software program, such as might be readily observable by its end-users, to the Technological Advancement being sought. Advances in the claimant's technical knowledge generally occur at architectural, algorithmic and design levels in software. These Advances which enable the development of the commercial functions and features should be the Technological Advancements cited by the claimant.

A commercial advancement (as distinguished from a Technological Advancement) in the software field can be made by resolving only routine design challenges (i.e. resolved solely through discussion of design issues, code production, debugging and acceptance testing). At the outset, the claimant had a valid conceptual model of the solution, the relevant data or a standard means of obtaining it, the tools needed to detail the solution and to test it for acceptability and a generally accepted process for obtaining the results. In this case, the solution and the means of detailing it was generally understood at the start of the project and the claimant has made no Technological Advancement, has gained no technological knowledge (in the sense of Advancement) as a result of the project. He did what he had to do using methods that were known to him at the outset. This type of project would be ineligible. In declaring the Technological Advancement in the project technical description, the claimant should, in a total of two or three sentences, answer the questions:

1. In what technological field are you seeking a Technological Advancement? [This defines the general qualifications of an expert who can assess the claim for a Technological Advancement. In the case of software development, the field is often Information Technology or Computer Science.]
2. What is the Advancement sought in terms of technical goals (quantified if possible) that you seek in this project? OR, put another way, what do you define as technological success (as opposed to commercial success)? [This establishes the probable forward limit of eligibility within this project i.e. when this Advancement is achieved or abandoned, the SR&ED project is complete. Commercial success often flows from technological success, but is separable from it.]

3. What was the state of the relevant art in this field of technology known to you at the start of the project? [The difference between what could be achieved using technology (architecture, constructs, algorithms, etc.) proven prior to the start of this project and what is achievable by incorporating the Advancement is evidence that an Advancement was sought. It also shows that the claimant had the understanding of the field expected of a competent participant in an SR&ED project.]

For clarity, we note that project eligibility requires that the claimant provide evidence not only of Technological Advancement but also of Technological Uncertainty and Technical Content.

Finally, we note that there is inherently a relationship between Technological Uncertainties and a true Technological Advancement. One making a claim should always be able to identify the technological advancement in his knowledge that is associated with solving a Technological Uncertainty, i.e. what he learned through experimentation. See the issue sheet, *"Experiments within Software SR&ED"* and the *"Guidance to the claimant"* for further discussions of the relationship between true Technological Uncertainties and uncertainties that are simply routine design uncertainties, and the Technological Advancements that are associated with true Technological Uncertainties.

Example of a well-stated Technological Advancement:

We seek a Technological Advancement in the field of Information Technology. In September of last year our competitor, MedsInc, announced its Framis encoding software capable of encoding a gigabyte in seven seconds on a 400 Mhz Pentium III processor. We seek to determine the means of constructing a Framis which will encode a gigabyte in less than five seconds on the same processing platform and which will scale to higher speed platforms more gracefully than does the MedsInc offering.
[For clarity we emphasize that while this is an acceptably declared Technological Advancement, the claimant must also provide evidence of Technological Uncertainty and Technological Content. This evidence will be readily definable if the Technological Advancement is robust and succinctly declared.]
Eligible or Ineligible Dependent Upon the Facts of the Case
Consider the following claimant submission:
In September of last year our competitor ABC-IT Inc. released a new tool suite for the compression and modification of electronic maps and overlays. The tools are designed for use on small platforms, (PDA's, Palm Pilots, Palmtops) They allow the user to make notes, and modify the electronic maps as they are doing field work. They can then upload the changes to their desktop PC, back at the office, where a full GIS package resides. Our product is very similar to that of ABC-IT Inc, however due to limited memory and battery life only limited size maps could be loaded, and a limited number of notes could be attached. Our competitor has released their software with a 50% further decrease in their compressed maps, i.e. they can compress a 1 meg. map to less than 40K. Our current best compression is to get a 1 meg image down to 90K. In order to maintain market share we must at least meet their performance, and develop a new compression technique. Our eventual goal is to be down to 30K.
Technological Objective:
To develop a new compression tool for GIS information with the capability of compressing a 1 meg map down to 30K. This has to be accomplished with less than 2% data loss.
Technological Advancement:
Through development and experimentation with several approaches, we have managed to develop a compression tool using a data communication standard (X2 standard for hardware compression), and a method of analyzing the

maps and overlays, synchronizing them into a single image and then using a modified version of MPEG 3 compression. The modified software compression allows for easier separation of the map from the overlay once the data is transferred from the hand held unit to the desktop PC. This method shows promise and to date has resulted in compressing 1 meg files to 60K. We are convinced that we can improve this to our intended goal.
(Notice that in this submission the section headed 'Technological Advancement' instead lists the results achieved; however the technological advance being sought is identified in the first paragraph and the objective). Generally, this Advancement would qualify, but it would not qualify in either of the following two situations.

1. *While doing the preliminary technical feasibility work we discovered a company in the US that has a tool ideally suited for our needs. We are currently working out a licensing agreement for resale. With a couple of parameter changes, their tool will give us our target compression.*
2. *In the early part of the technical feasibility study portion of the project, we learned that one of the senior software engineers had resigned from ABC-IT Inc. We hired him and he is redeveloping their algorithm for our application. We have decided that matching the ABC-IT Inc. performance will be adequate.*

Issue Title:Business Environment

Background: IC97-1 Sec. 3.2: A technological uncertainty arises when the solution is not readily **apparent to appropriately skilled and experienced software developers.** Uncertainties that arise from lack of diligence or lack of appropriate expertise, such as the failure to use **commonly available information**, or lack of programming knowledge are not technological uncertainties and relevant to eligibility.
As well, the tax credits are not to serve as incentives simply for training or learning on the job but for efforts to develop new or improved technologies through experimental development or research.

Problem Statement: Does a company's business environment affect what is eligible by modifying what is judged to be an "appropriately skilled and experienced" developer or what is judged to be "commonly available information"?

Discussion: An appropriately skilled individual is one whose qualifications and/or experience enable him or her to operate competently within a particular field . Such a person could be reasonably expected to produce a functional solution to a problem if such a solution exists within the constraints imposed upon the solution. The skills expected for a company's developers are those commonly found within the context of a company's competitive business environment, and appropriate to the work being undertaken.
The three criteria for eligibility are to be applied with reference to the circumstances of the organization that executes the project. The judgement as to what knowledge of solutions a company should have or how far a company should search for solutions is to be assessed within the context of what it is reasonable to expect of a company conducting the project in a similar business environment. If a third party executes a project on behalf of a claimant, the applicable business environment against which eligibility will be evaluated is that of the third party. Nevertheless, a claimant should not be denied eligibility simply because third parties exist for whom the same project would be standard practice, .if the claimant did not have reasonable access to that expertise.
As noted, the meaning of "commonly available" has to be sensitive to the business' environment and hence what can be expected of the tax payer's developer's project employees

All information technology workers are expected to have knowledge of the commonly available evolving tools of their trade. Workers are expected to make reasonable efforts to learn of new tools that might be applicable to the task at hand, but can be excused for failure to discover a relevant tool not commonly available in the public domain. Simply learning to solve the problems that are commonly encountered during the first time use of a commonly available tool which happens to be new to the claimant's business environment does not on its own involve the resolution of a Technological Uncertainties and hence produce a Technological Advancement. In the absence of any other fundamental technological problems, this is simple on-the-job learning.
The redevelopment of a solution which was generally available will usually fail the tests of both Uncertainty and Advancement. When making a claim based upon such a project, the onus will be on the claimant to show that (a) he was in fact unaware of the existing solution, and that (b) he made reasonable efforts to search for available solutions before embarking upon his project.

Hint: Often it is a company's ability to show how he normally searches for generally available solutions or that he did search for them or that he has established make/buy assessment practices that provides useful evidence that a "generally available" solution was not available to him in his business environment. The business environment in which the software is developed or is to be used can give rise to technological challenges and hence Technological Uncertainties, but business constraints per se are not of themselves Technological Uncertainties.

For example, one claimant may choose for business reasons (e.g. time to market, a perceived need to differentiate himself from a competitor's product) to attempt technical solutions involving Technological Uncertainty which another claimant might choose to avoid. A small business may try to minimize the development costs of a software system by taking some shortcuts that increase his risks. A larger firm might not take these same shortcuts because of the increased commercial risks. In this scenario, the claimant may be able to provide evidence that the shortcut was the source of a technological challenge and hence of a Technological Uncertainty and hence he sought an advancement.

For example, acceptable evidence might be advanced by showing that component hardware or software chosen because of cost considerations were less capable than those required for the system's certain success. The resolution of the resulting Uncertainties, if done experimentally, could lead to an Advancement and thus to project eligibility.

Example 1

A company developed applications in COBOL/CICS for deployment over a wide area network wholly controlled by the company. The development group had no prior experience with object oriented development using the C++ language, where the application would be deployed over the Internet. As these two skill sets are so radically different, an eligible SR&ED project in the latter area would have to include staff with directly relevant skills (OOPS, C++, HTML, CGI etc.). These individuals may be hired specifically for the project, contracted, or trained before the project started; whichever course is chosen, the company's original lack of experience is not relevant to the eligibility of the project.

Example 2

(a) A browser plug-in was developed by a software company specializing in browser development. While performance testing and de-bugging were required, the planned functionality and performance were achieved without any significant uncertainty, using routine product engineering. The work was not eligible as SR&ED.
(b) A similar plug-in was required by a clothing retailer. The work was contracted out to the browser developer, and was ineligible as SR&ED to the clothing retailer and the browser develop alike.
(c) The clothing retailer decided to have its business systems development group do the work in-house. Since no-one in the group had relevant experience, the work was time-consuming and demanding, and the developers were uncertain whether they would be successful. As a result of doing this work the group developed new expertise through the training which this development project provided. Nevertheless, as others were available to do the work, and for them the work would not be qualifying SR&ED, this was simply on-the-job training for the clothing retailer and the work was not eligible as SR&ED.
(d) A developer of a software environmental management system tried unsuccessfully to purchase existing software that could be adapted for his new project, or to find a contractor or personnel for hire able to do the work. The company was left with no option but to have its business systems development group do the work. The objective

was not training for the development team but rather the development of knowledge which was not available in the public domain. The work met the three criteria and was eligible as SR&ED.
NOTE: While a company's technological knowledge-base can be enhanced through formal training, on-the-job-training or through work on eligible SR&ED projects, the SR&ED program is not a training incentive. A claimant's initial lack of knowledge or an attempt to re-develop knowledge that is available in the public domain does not give rise to technological uncertainty.

Example 3

The browser development company heard that a competitor was working on a new streaming technology to support full-screen real-time video. The new technology was of course proprietary to the competitor company and inaccessible to the browser company so the existence of the competitive work would have no relevance to any claim made by the browser development company.
It was later shown that a paper explaining just how to do this had appeared in a Swedish-language journal on Information Theory some months before the start of the project within the browser company. This did not affect eligibility, as the test is **reasonable** accessibility.

Issue Title: Experiments within Software SR&ED

Background: Subsection 248(1) of the Canadian Income Tax act defines scientific research and experimental development (SR&ED) as requiring a "systematic investigation or search carried out in a field of science or technology through experiment or analysis."
This statement is echoed in Section 1.2 of IC97-1. In Exhibit D of the same document, we find that "experimental development aimed at filling technological knowledge gaps, as necessary to develop a software program or system", is listed as frequently eligible.
We also find the following ideas in Section 3.1 of IC97-1. "An effort to achieve technological advancement will be accompanied by experimentation or analysis in a situation where there is technological uncertainty about whether or how the technological advance can be achieved. It is the **attempt** to achieve technological advancement that is important in determining eligibility."

Problem Statement: Some claimants believe that debugging is experimental work as called for by the definition of SR&ED in that it involves a series of trials and observations. Some claimants believe that all software development is experimental for the purposes of the Act by the very nature of the activity.
The CCRA 's and the Department of Finance's position is that some software development does not meet the SR&ED guidelines because it is not basic research, or applied research or experimental development. Software that is developed through routine design activities in projects which have only routine design challenges does not meet the SR&ED criteria of Technological Advancement and Technological Uncertainty.

Discussion: An experiment within the context of the SR&ED Program involves setting up test conditions and making observations or measurements aimed at filling gaps in our technical knowledge. The result of the experiment, whether it is successful or unsuccessful, provides an increase in knowledge of software systems relative to the Technological Advancement sought and/or the Technological Uncertainties. The new knowledge is applicable beyond the system under test.
Thus inherently, Technological Uncertainties are associated with advancements in technology knowledge. One making a claim should always be able to identify the technological advancement in his knowledge that is associated with solving a technological uncertainty, i.e. what was learned through experimentation.
In software development, immediate problems are usually solved by "trial-and-error" rather than by experiment in the sense of the Income Tax Act. Trial-and-error involves executing a series of probes that were not sequenced in a systematic pre-plan. The objective here is to resolve a functional problem (as in routine debugging) rather than to gain understandings that are expected to be more widely applicable. The lesson learned in each iteration of "trial and error" is simply "that an option didn't work" and they are not applicable in a much broader sense. For each iteration the probe is chosen that is now judged to be the most efficient in resolving the immediate problem. The process proceeds quickly from iteration to iteration.
Resolving problems through the "trial-and-error" approach is eligible support work, but it is not the basis for a Technological Advancement, as the knowledge gained does not produce a true improvement in our understanding of the technologies.
In the context of software development and the legislation, experiments might be aimed at resolving design or architectural alternatives or systematically probing an inadequately specified interface. The experimental approach itself should be designed.

Debugging activity aimed at discovering coding errors and misunderstandings of interface requirements (call sequences, parameters, data formats, etc.) do not lead to Technological Advancements and embody only routine design uncertainties. Tests (in the trial-and-error mode) are designed and executed in a rapidly evolving sequence rather than being pre-planned to settle a design alternative. These activities are not "aimed at filling technological gaps" within the meaning of those words in IC97-1 and are thus ineligible in their own right. Nevertheless, this debugging activity is part of any software development. It can be claimed as eligible supporting activity when it is essential to the demonstration of a Technological Advancement and the resolution of Technological Uncertainty.

When faced with Technological Uncertainties many software developers design experiments using prototype code. Prototype code is not fully featured and is quickly built to check out critical performance issues or uncertain interfaces (Technological Uncertainties). The code is often abandoned following testing for the unknowns, but successful architectures and constructs derived through the work with the prototype are carried forward to the final design. When this design methodology is used, the claimant should have little trouble in succinctly defining the Technological Advancement attempted and the Technological Uncertainties - thus qualifying the project.

Following a test of a prototype, it may be necessary to build and test a fully featured system intended for productive use before the claimant can demonstrate that he is consistently able to demonstrate the declared Technological Advancement. In this case, the eligibility of the project extends to the point in time when the Technological Advancement is consistently demonstrable, or the time when the Uncertainties have been resolved. In the context of the above, eligible activities include those needed to produce and assemble the software suite and the costs to the claimant of the testing. Some of this testing may be done while the software is in commercial use. Eligible activities include those tasks that were necessary to prove the Technological Advancement and resolve the Technological Uncertainties. The claimant should separate these from the usual concurrent activities that are essentially in support of an early user or of routine product extensions.

The claimant does not have to build prototype software in order to do experiments within the meaning of IC97-1. He can legitimately conduct experiments upon software that is intended for the end user.

As stated in IC97-1, "experimentation or analysis in a situation where there is technological uncertainty", is a strong indication of project eligibility. The claimant should strongly link the experimental work to the Technological Advancement sought and to the Technological Uncertainties declared in the technical description of his project. In this way, the claimant will show clearly that specific experimental work is aimed at resolving the Technological Uncertainty and thus constitutes an **attempt** at providing the Technological Advancement. The onus is on the claimant to provide evidence of experiments. An engineering notebook or any other log with dated entries that briefly state the aims of experiments and list the observations that led to the experimental conclusions is strong evidence. Dated project plans, sequential records, project meeting notes and relevant e-mail are also useful in this regard. Where practical, the claimant should retain executable prototype code and associated test data to provide demonstrations of the project progress, iterations and setbacks. The cost of creating and keeping such records is small compared to the tax benefit claimable.

Issue Title: Fixes, Feature Extensions and Testing

Background: IC 97-1 and IC 86-4R3 both list the three criteria for project eligibility. In Sec 5.1 of IC 97-1 we read that, all activities that flow systematically from the definition of technological requirements to testing and documentation qualify when necessary and sufficient for the attempt to achieve the Technological Advancement and resolve the Technological Uncertainty.

In Sec 5.4 of IC 97-1 we also read that testing qualifies, provided it is necessary for the attempt to reach technological advance and resolve uncertainty; is systematic; and is documented. Furthermore, to qualify as a supporting activity, Section 5.4 says that eligible testing, "..is performed in controlled conditions with feedback from all testers into the development process...". Finally, this section goes on to say, "Testing that deals primarily with user acceptance, suitability, marketability, or competitive assessment does not qualify".

Problem Statement: Software fixes, feature extensions and testing of software that occurs both before and after the commercial release of a new product or before and after the software's first use internally in an operational sense are very much a part of any commercial software development project. On occasion, these activities may also be part of an SR&ED project, both before and after commercial release or operational introduction of the associated product.

The key issue with both fixes and testing is when and to what extent work required to develop particular fixes and to test a software suite is eligible, as it is not always eligible. Feature extensions are typically (but not always) developed following the software's first commercial release or first internal use in the case of software developed for internal use. Here again, the issue is when and to what extent is work required to develop a particular extension eligible, as the activity is not always eligible.

Discussion::This work can be eligible within an eligible project (e.g. an original software development project or a subsequent project associated with the same software suite) in cases where the activity supports efforts to make the Technological Advancements and/or resolve the Technological Uncertainties and is commensurate with that need. The work can also be eligible if the work lies at the heart of the declared Technological Advancement of an eligible project i.e. the work is eligible in its own right. In all other cases, the work is ineligible.

In the first case, eligible work can be associated with fixes, feature extensions and testing whenever they can be shown to be commensurate with the needs of the original SR&ED project. Here, it is helpful to understand two issues:

1. At what point in time does eligibility cease for the original project? Any testing and work on fixes and feature extensions which occurs after this time is ineligible within the original project. [However, the work will be eligible if it can be shown that it is commensurate with the needs of a subsequent eligible project or is an eligible project in its own right.]
2. Under what circumstances is work on "fixes, feature extension and testing" commensurate with the needs of an eligible project?

When Does Eligibility Cease?

Project eligibility for any SR&ED project ceases when:

- the Technological Advancement has been demonstrated to be achieved consistently under the circumstances and constraints declared in the definition of the Advancement within the project's technical description and the associated Technological Uncertainty is thus resolved
  OR

- it is determined that the Advancement sought will not be achieved.
  The first point to understand is that a commercial development and an SR&ED project relative to the same software development may not be identical. In theory, the commercial project and the SR&ED project are fully decoupled in time. In practice, they are often entwined. Hence, the point in time when software is first put into service does not necessarily determine the SR&ED project completion date. Specifically, although a commercial project and an SR&ED project may have identical start dates and the SR&ED project may end with the first commercial use of the project software, this need not be the case. Likewise, the SR&ED project may start before the commercial project is launched and may continue after the first release of the commercial software or its use in operations. At the other extreme, the SR&ED project may both start and finish inside the dates of the commercial project.

  In summary, the cessation of eligibility is not linked to the status of the commercial project.

  Eligibility may cease before, at, or after first commercial use.
  If the Technological Advancement has been consistently demonstrated and the related Uncertainties have been resolved before commercial use, the SR&ED project (and therefore, eligibility) ceases before commercial use.

  Similarly, if the declared Technological Advancement and the Technological Uncertainties are unresolved as of the date of first commercial use, eligibility continues beyond the first commercial use for those activities needed to achieve the Advancement and to resolve the remaining Uncertainties.

Eligibility of Fixes, Feature Extensions and Testing

Eligible within a subsequent project

To be eligible in its own right, work on fixes, and feature extensions must be able to meet all three of the criteria for an eligible project, e.g., the work must be directly associated with obtaining Technological Advancements and resolving Technological Uncertainties that go beyond those associated with the original development project. In other words, one must be able to identify new constraints and technological challenges, and further potential advance in the technology, to qualify a subsequent project.

Eligible as commensurate with the needs

Testing will be eligible when the work can be shown to be an essential part of an eligible project, either of the original project or of a subsequent one. Testing will normally not be eligible on its own (i.e. testing is usually a supporting activity which may be commensurate with the needs of an otherwise eligible project). In the unusual case that the testing is associated with advancing testing methodologies, the work may be an eligible project on its own. In this case, the three criteria for eligibility must be met.

When fixes and feature extensions demanded commercially have to be operational before the software can be tested for its Technological Advancement or before the Technological Uncertainties can be resolved, the work is eligible as commensurate with the needs of the project. In this case, the work can be shown to be a reasonable precursor to proving out (or testing) the architecture or constructs associated with the Technological Advancement. Why this is the case should be explained briefly, but specifically, in the project's technical description.

As well, in most software development projects, the claimant has to provide a test bed for testing (proving out) the commercial product. The commercial product itself may be the most practical environment, from a technological/methodological perspective, for performing the testing required to show that the Technological Advancement is consistently demonstrable and that the Technological Uncertainties have been resolved. In such cases, those of the commercial features which have to be operational in order to provide an adequate test bed for the Advancement are part of the SR&ED project. Specifically, in these cases the activities involved in developing fixes and feature extensions required to provide the test bed are commensurate with the needs of the project. The claimant should clearly explain why the advancement can most practically be proven out through the commercial product when describing the Technological Advancement in the project description. He should state how he will determine that the Advancement has been achieved. It sometimes happens that the work associated with the introduction of fixes and feature extensions is driven by deficiencies discovered in the architecture and constructs. Such work is eligible if it can be shown that it was commensurate with the needs and directly in support of the Advancement.

Ineligible when not part of an eligible project

Once a software architecture or construct has been proven, the claimant will often have difficulty showing eligibility when a fix or new feature is accomplished within this architecture using tools that he has used before. In this case, the work usually involves only routine design challenges and no Technological Advancement.

The Particular Importance of Assignment Records

Work on a commercially defined project often involves a mixture of eligible and ineligible activities. This is almost always the case following the first commercial release but can be the case at any point in the project. We strongly advise the claimant to keep good assignment or time records that will allow for the clear

delineation of eligible and ineligible costs, especially for work that follows first commercial release.

Typically, following the first commercial release, at least some of the efforts of the development team are needed for field support and for the straightforward addition of features (i.e., within the defined architecture and using established tools and test methods) as may be requested by the first user or by prospective users. These efforts do not qualify as they are not commensurate with the activities needed to achieve a Technological Advancement.

Example 1

Technological Advancement: We seek a Technological Advancement in the field of Information Technology. In September of last year our competitor, MedsInc, announced its Framis encoding software capable of encoding a gigabyte in seven seconds on a 400 MHz Pentium III processor. We seek to determine the means of constructing a Framis which will encode a gigabyte in less than five seconds on the same processing platform and which will scale to higher speed platforms more gracefully than does the MedsInc offering.
Comment Under this declaration of Technological Advancement, the project ends when the Framis software is sufficiently debugged and sufficiently featured that it codes a gigabyte in less than five seconds on a 400 MHz Pentium III and it can be shown that it scales to at least a couple platforms with clocks in excess of 400 MHz in a manner which is more "graceful" than was possible with the competitor's algorithm. This point in time may be well beyond the first delivery of Framis to a 400 MHz platform that is able to encode a gigabyte in less than five seconds.

Example 2

Technological Advancement: We seek a technological advance in the field of Information Technology. In September of last year our competitor, MedsInc, announced its Framis encoding software capable of encoding a gigabyte in seven seconds on a 400 Mhz Pentium III processor. We seek to determine the means of constructing a Framis which will encode a gigabyte in less than five seconds on the same processing platform.
Comment Under this declaration of Technological Advancement, the project ends when the Framis software is sufficiently debugged and sufficiently featured that it can be shown that it can encode a gigabyte in less than five seconds on 400 MHz Pentium III processor. This point in time may well be before the first commercial delivery of the software. Furthermore, unless the Technological Uncertainties involved in getting this performance are strong in that they involve at least analysis of a couple alternative approaches, the Advancement or Uncertainties may be judged as routine design and therefore the project could be found to be ineligible.

Example 3

TeethTek Inc is developing software for archiving medical files and images for dentists' laboratories. Tests were done with existing technologies, but none could achieve the design objective. A new approach called "Capture by Wavelets" (to capture data from cameras) was developed, which included the development of a new data compression algorithm. This work was determined to meet the criteria for SR&ED.
Subsequently, in extended trials, the data compression was found to be too slow, and a new technique was developed to speed the compilation. This work, described in terms of its new technological objective, met the three criteria in its own right and was accepted as SR&ED.

Example 4

ABC Company provides an image data capture services to physicians and dentists. ABC bought a third party new product called "Capture by Wavelets". ABC tested the product with different parameters and found that the process was too slow to economically capture data directly from cameras. They recommended some enhancement of the product to the suppliers. A new version of the product was released to ABC. ABC parameterized the setup for two different types of computers. They claim Advancement in discovering and documenting some bugs within the third party product and for adding new functionality through a coupled spreadsheet development.
The work done by ABC was confined to using and parameterizing third party products and the use of existing spreadsheet software in a manner in which its was expected to be used; it is not eligible.

Issue Title: Systems Uncertainty

Background: Scientific or technological uncertainty is described in IC 97-1, Sec. 3.2 as follows: "A scientific or technological uncertainty in software development arises when the solution, or the method of arriving at the solution, is not readily apparent to appropriately skilled and experienced software developers after they have analyzed the problem using generally known software development techniques." Systems uncertainties may arise during the integration process of new or standard, off-the-shelf components due purely to the unexpected interaction of the components when assembled into a system.
IC86-4R3 further identifies "the activities undertaken to resolve technical uncertainties are eligible if the claimant cannot obtain the solutions through commonly available sources of knowledge and experience in the business context of the firm."

Problem Statement: There are situations in which the claimant may make claims for advances made due to technical uncertainties that were discovered during the integration of components, even if these were standard, off-the-shelf components.
The uncertainty here is not in the individual modules or components, but in the modules or components acting as an integrated system. This includes the integration of new with new, new with old, and old with old modules or components, or uncertainties created by the target environment such as sometimes occurs during porting from one environment to another. These uncertainties can lead to eligible advancements. Mundane integration, standard debugging, and uncertainty arising from project management are not eligible except as supporting activities.

Discussion: A **system** is made up of software modules or components as well as firmware and hardware. The system is the tested and documented operational entity made up of the integrated software and hardware components.

The **uncertainty** in system integration may arise from unexpected behaviour in individual components, defects in individual components, inadequate or incomplete information about the components, unexpected behaviour between or amongst the components, uncertainty associated with the integration of the components, and the issue of predictability (can we reach the objective deterministically?). Any mix of one or more of these conditions may lead to uncertainty at a system level on the part of the claimant. However, if the uncertainties are limited to learning from a vendor or the vendor's literature of the applications to be integrated, the uncertainties are routine design challenges rather than Technological Uncertainties. Likewise, the setting of supplier specified parameters involves only routine design challenges.

The claimant should identify the evidence used in defining the uncertainty and show that the integration could not be achieved using standard practice. This evidence could include internal documentation, trade literature, vendor documentation, vendor technical support or contact with other qualified specialists via electronic bulletin boards etc. Where the claimant is dealing with the integration of components built by third-party vendors, these vendors should, where possible, be referenced in defining the uncertainties. It is critical that the claimant clearly identify the associated advances in knowledge or technological discoveries that are inherent in the solution.

Many system or component integration activities involve the integration of third party components designed and advertised to function together. Integrating these components or

systems typically should not, in principle, lead to technical system uncertainty and is generally considered a routine development and thus ineligible. However, if the claimant experiences significant problems that are not caused by his lack of knowledge of published documentation on the components or bugs in those components, he may be able to satisfy the Uncertainty criterion.

At other times, it is clear to the claimant that there are significant uncertainties involved in system or component integration activities where these components or systems have not been specifically designed to function together. Even components designed to function together can be sensitive to version or release levels of the other components, the operating system, or the firmware of the equipment. In addition, specifications for legacy systems and old components may be incomplete or unavailable to the extent that experimental work is required. In these cases there may be significant levels of uncertainty requiring experimentation and advances in both software and understanding to be made in order to successfully integrate a system. While the claimant may recognize the uncertainty, and experimentation may be carried out by the claimant, there may be cases where the claimant will need to have the vendor implement the change as he may be the only party with access to the source code. These cases may be eligible if the change made by the third party goes beyond a bug fix i.e. the need for new functionality or structure has been identified. However, if the claimant's role has been only to identify the problem, the claim will not be sustainable. The claimant must provide evidence that his project meets all three criteria for eligibility.

When performing system integration, there may be uncertainties involved both in determining the approach to use to integrate the components or modules, in the determining the specifications of the components and/or in correcting defects in third party components. The activities involved in resolving these uncertainties and evaluating and testing alternatives can be eligible work within the intent of IC 97-1.

Since the onus is on the claimant to show that a system uncertainty existed, the claimant should be able to cite the technological base from which he claims a systems uncertainty. It is essential to make an accurate declaration of the uncertainties as of the start of the project and to define how it will be determined that the uncertainties have been resolved. The technological base described by the claimant may include descriptions of performance, throughput, stability, concurrency, step response, MTBF, and footprint.
If the claimant plans to make claims that extend beyond the time of commercialization, it is important to document the outstanding uncertainties at the point in time of first commercial use.
Quality documentation is key to any successful SR&ED claim. This is particularly true where the claim rests largely on the resolution of systems uncertainty. To develop a robust project technical description for his claim the claimant should show that the work advances our understanding beyond that which could be gained by learning from generally available courses or published material etc. (see Business Environment). The description should identify the technical alternatives investigated, the design, or physical constraints, the limitations in tools and processes, the platforms considered, and the technical design trade-offs, tests performed and test results. The project technical description should identify why the claimant was uncertain of finding a solution or of the nature of the best of alternative solutions.

Example 1

We seek to resolve the uncertainties in integrating a third party software application by ZeroInc with a third party OLTP transaction server by ABM. The ZeroInc application was developed to operate with the same operating system and database as we use, but ZeroInc has no experience with our OLTP transaction software from ABM. We seek to determine the means of developing a module that will connect the ZeroInc application to the OLTP transaction server so as to process at least 5 transactions concurrently.
[For clarity we emphasize that while this is an acceptably declared system uncertainty, for the project to be eligible, the resolution of the system uncertainty must be approached systematically (the project must be shown to have scientific content) and there has to be an uncertainty at the project outset as to how the specific advance might be achieved.]

Example 2

We seek to identify and eliminate the uncertainties involved in integrating three specific software packages developed by third parties in order to achieve a defined functional specification. We are operating the ZeroInc application internally, but wish to expand the user base of the application to the Internet. This means that the transaction server must be able to process multiple transactions concurrently. To accomplish this we must establish secure communications over the Internet using the IP protocol connected to our secure web server and integrated to our ZeroInc based OLTP transaction server. We feel that this can be accomplished at the client-end by using a thin client browser using an encrypted connection to the Internet. At the host site we will need to build a stateless web server and connect it via a transaction mechanism through a firewall to our new OLTP transaction server.
We held discussions with ZeroInc, ABM and Solar, our application and system vendors. Based on these discussions we developed a list of alternative solutions for each part of the system. Our vendors indicated that none of the possible combinations had been integrated before. We needed to analyse and experimentally evaluate performance trade-offs within the integrated system in order to determine their integrated functionality, reliability, security and performance. Starting with a seemingly viable scheme for passing information between the components, we built prototype code, determined throughput, delays and functional limitations imposed by the interactions of our interface with the multiple components and the components with each other. Through three iterations of these experiments we determined the most viable architecture for our interfacing scheme and the optimal combinations of parameters within the components. [Note that to provide strong evidence of eligibility the project technical description should describe the testing and how the test results drove changes in the interfacing architecture.]

Example 3

A company specializes in the development of integrated multi-platform software solutions for various information systems applications (multimedia, imagery, operations management). Four different software packages were integrated to form a new product. A series of compatibility and performance tests were carried out with different operating systems (Windows 95,98, 2000), servers and networks (Windows NT, Novell, SQL server). The company was unable to obtain proprietary information from the vendors that would have facilitated the integration process. The problems had to be resolved through several work-arounds because the claimant did not have access to the vendor's code. System performance was enhanced by using file buffer sharing, record tracers and a time slicing algorithm. The company developed a unique approach that could be used in future integration processes. This project is eligible because technological uncertainties were resolved in pursuit of an advance in technology.

Example 4

Another two commercially available software packages were integrated by the claimant of Example 3 into the product of Example 3. Problems relating to the integration of recent versions of this software with target platforms were encountered. Though some of the problems were different from those encountered in the previous integration process, the methodology that was developed earlier helped to achieve final integration with only minor changes to

the work-arounds. This project is not eligible as the successful integration of the components was achieved through routine engineering, based primarily upon the claimant's previous experience.

Issue Title: Interfaces to other Packages,and Users

Background: IC97-1 includes the following bullets under the heading of "Frequently ineligible":

- Business application software development, customization, and graphical user interface building limited to using commercial off-the-shelf software tools and development environments.
- Routine software upgrading or maintenance, such as many instances of porting software to a new operating system, converting to a new programming language, writing format translators for interfacing to third-party software systems, writing hardware device drivers, code optimization, fine tuning, and debugging.

However, in Section 2.13 of IC86-4R3 we read, "if the 'routine' engineering or 'routine' development activity is carried out in support of an eligible experimental development project, then the activity is eligible."

Problem Statement: Experience has shown that the designation of "frequently ineligible" is not very helpful in determining eligibility because ineligibility is not categorical. Although IC97-1 is accurate in its statement that projects which involve only interfacing software packages to each other (new to new; old to old; new to old) are frequently ineligible, and likewise for the development of user interfaces including graphical interfaces, eligibility cannot be categorically denied. Regulation 2481© specifically mentions incremental improvements in designs as eligible.

Discussion: These activities are frequently ineligible because it is often impossible to show Technological Advancement and Technological Uncertainty i.e. uncertainty beyond routine design challenges.
Eligibility could arise when a claimant finds that software systems which were generally thought to be compatible turn out not to be so in his application. Interfacing design activities are eligible in their own right if it can be shown that the three criteria for eligibility are met. This might be the case if interfacing alternatives had to be explored through serious analysis and/or prototype system construction. The claimant should retain the original analysis as evidence of its seriousness. He should record and retain evidence of testing and testing results on prototype systems. A simple engineering notebook or e-mail that chronicles the nature of compatibility failures and the construction of prototype code can provide good evidence of eligibility if there is eligibility. Eligibility is more likely if multiple prototypes were needed, showing there was Technological Uncertainty, and if there was a resultant knowledge gain or Technological Advancement.

Human-Computer Interface work (HCI) is, like any other work, eligible if it can be shown that the three criteria for eligibility are met. This would generally be for advance in the supporting technology, a new video streaming technique for example. Work which is limited to normal use of existing technologies, such as designing new or enhanced screens and menus, is not Technological Advance, and uncertainty about how best to present information to users is not Technological Uncertainty.

Interfacing activities are eligible as supporting activities if it can be shown that they are essential to resolution of Technological Uncertainties and/or the consistent demonstration of a Technological Advancement claimed for an associated eligible software development project. This might be the case if it were necessary to develop the user interface or the interfaces between software packages before the software that is the principle subject of the

claim could be tested.

Consider a project for which the Technological Advancement is claimed as follows:

Example 1

"In September of last year our competitor, MedsInc, announced its Framis encoding software capable of encoding a gigabyte in seven seconds on a 400 Mhz Pentium III processor. We seek to determine the means of constructing a Framis which will encode a gigabyte in less than five seconds on the same processing platform and which will scale to higher speed platforms more gracefully than does the MedsInc offering."
The claimant showed that the Framis development was itself eligible in that the project embodied Technological Uncertainties and the activities went beyond routine fine tuning. The claimant also showed that the new Framis had to be interfaced to an existing data input system and that a new graphical user interface had to be developed before it could be determined whether or not the new Framis would "scale to higher speed platforms more gracefully". The costs of the supporting activities, although routine in themselves, are eligible.

Example 2

Discussion of the alternative approaches for rendering the graphical output of a simulation package on a screen produced a list of three seemingly reasonable approaches. The central problem was that the graph was expected to evolve on the screen as the simulator ran in compressed time. Since there was no obvious choice among the alternatives, it was agreed that all three approaches would be analyzed, on paper, to estimate their relative loads on the CPU and the architectural complications of being able to change the display parameters while the simulation was running. The two most promising approaches emerging from the analysis would then be coded in prototype and run for live comparison against the same five minute data segment from the simulator.
The project is eligible as described, but the technical description of the project will have to specifically identify the Technological Advancement, the Technological Uncertainty and the Technical Content to provide acceptable evidence of eligibility
If the decision as to which one of the three approaches should be used was successfully made through discussion alone (no serious analysis or experiment), the project would not be eligible because the claimant had decided and evidenced that the only uncertainties were routine design challenges (no Technological Uncertainty).
If that one approach were shown to be inadequate as a result of testing or observations following the software build, the claimant would have demonstrated Technological Uncertainty. If he then initiated a project to solve the problem and could show Technological Advancement and Technological Content within that project, he would have an eligible project.

Issue Title: When is Quality Control Eligible?

Background: In IC86-4R3, Section 2.5 we read that Subsection 2900(1) of the Regulations states that SR&ED does not include, "quality control or routine testing of materials, devices, products, or processes."

Section 5.1 of IC97-1 states that, "Qualifying activities include (9) testing the software that embodies the Technological Advance." Section 5.4 of IC97-1 further clarifies the issue with the statement, "Qualifying testing is the activity in which the software arising through SR&ED is verified against Technological Advancement goals established in the SR&ED project."

Problem Statement: The Quality Control organizations in many software firms are involved in the systematic testing of new software for both features and responsiveness. Because of the negatives associated with "quality control" in some SR&ED Information Circulars, the involvement of quality control personnel in software projects has raised questions as to the nature of their activities within any claimed project. In some cases, claimants have felt that the activities of quality control personnel were unduly scrutinized when their activities were clearly either directly involved or supporting.

Discussion: The intent of the CCRA regulations is to disallow eligibility to the routine analysis and data gathering performed by many Quality Control organizations in industry - and in manufacturing environments in particular. Any testing done to determine whether or not the sought Technological Advancement has been achieved or testing done to reduce a declared Technological Uncertainty qualifies as a direct activity. This is of course true regardless of the claimant's organizational structure and the department in which the testing is undertaken.
The claimant should be prepared to provide evidence that quality control work performed was necessary to being able to consistently demonstrate that the declared Technological Advance had been achieved (partially achieved, or proved infeasible) and/or the declared Technological Uncertainties had been resolved. Evidence might include a description of the tests run, the test results, relevant e-mail and problem logs. It should be clear in the project technical description that the testing claimed was required to show the extent to which the Technological Advancement sought had been achieved and the Technological Uncertainties resolved.

Example 1

In September of last year our competitor, MedsInc, announced its Framis encoding software capable of encoding a gigabyte in seven seconds on a 400 Mhz Pentium III processor. We seek to determine the means of constructing a Framis which will encode a gigabyte in less than five seconds on the same processing platform and which will scale to higher speed platforms more gracefully than does the MedsInc offering.
Comment If this project was judged eligible after examining the factual evidence of the case, then the testing needed to determine whether or not the new encoding algorithm consistently met its target of being able to code a gigabyte in five seconds would be eligible. Eligibility would include the testing of performance against several gigabyte samples and the writing of a report that described performance as a function of the properties of the specific data samples.
On the other hand, if the coding algorithm were later reprogrammed in another language for another platform with reasonable expectation that the same performance specification would be met, rerunning the same tests would be ineligible. In this second case the testing is simply acceptance testing or traditional quality control. Testing has not contributed to a Technological Advance and there was only routine design challenges, in this instance, as to whether the tests would pass following routine debugging to find coding errors and to tune performance.

Issue Title: When is Analysis Eligible?

Background: Subsection 2900(1) of the Regulations defines scientific research and experimental development (SR&ED) as requiring a "systematic investigation or search carried out in a field of science or technology through experiment or analysis."
This statement is echoed in Section 1.2 of IC97-1. In Section 5.1 of the same document, we find that "technical analysis and design" is among "Activities that usually qualify within SR&ED projects". Repeated references are made to analysis as an eligible activity throughout IC97-1.

Problem Statement: Although analysis is repeatedly referred to as an eligible activity, analysis is also used during routine design. It is therefore clear that simply showing that analysis was done will not necessarily qualify a project as SR&ED eligible.
What, then, is the dividing line between eligible and ineligible analysis?

Discussion: In the field of Computer Science/Information Technology, analysis is usually closely coupled with system construction and testing within a project. In this case the entire project is reviewed for eligibility. Analytic tasks that are essential to the conduct of an eligible project qualify for treatment as SR&ED, regardless of whether or not the analysis employed only routine methods and whether or not the analytic conclusions, in and of themselves, advanced the claimant's knowledge.
The balance of this discussion covers the (rare) situation where a project consists only of analytic tasks.
If a project is to be made eligible through analysis alone, then the analytic tasks must satisfy the tests for Technological Advancement, Technological Uncertainty and technological content. The latter requirement is satisfied by any substantive and correctly performed analysis. The claimant therefore has to demonstrate with the factual evidence of the project that the criteria of Technological Advancement and Technological Uncertainty were also met.
In this case, determination of eligibility is based upon the objectives and the results of the analysis, rather than the methods and tools used in the process. The use of well-known methods and tools does not, in and of itself, disqualify analytic tasks from eligibility.
A Technological Advancement would come from analysis alone if the analysis was able to show that a proposed solution to a Technological Uncertainty was clearly possible or seemingly impossible. The project would pass the test for Technological Uncertainty if there were at the outset a genuine uncertainty that could not be easily resolved as to whether a viable design alternative existed at all or which was the best of a set of alternatives.
Note that where analysis is used only to select between known alternatives using methods that are commonly used to resolve the alternatives, the analysis tasks, on their own, will fail the test of Technological Advancement. Nevertheless, as previously noted, these same analytic tasks may be eligible as supporting activities within a project that is made eligible for other reasons.
Also note that analysis performed in planning a project when the project itself is not started, will probably fail the test of Technological Advancement.
It is particularly important that the claimant document and retain the original analysis as evidence of eligibility.

Example 1

The project objective was "We seek to show through analysis that the key to both graceful scaling to higher speed platforms and to speed maximization for a specified compression on a Pentium performing Framis coding is the optimal use of the Pentium look ahead. We also want to analytically determine the compression expected as a function of the number of bytes processed in parallel and to thereby determine the speed/compression trade-off." During the project the company used a combination of routine methods to analyze the trade-off between scaling, speed and compression on several computing platforms. It was able to rigorously define the relationships involved without prototyping the software. The definition of the relationship among the variables in question was not available in the public domain, and the determination of the relationships was technologically complex and the result was uncertain at the outset.

Issue Title: Eligibility of work to improve the process of software development

Background: IC86-4R3 Sec. 2.9 states the "Essential tests that must be met before any activity can be considered scientific research and experimental development include the criterion of scientific or technological advancement; the criterion of scientific or technological uncertainty; and the criterion of scientific and technical content."
The software development process continues to undergo change and evolution. When a project is mounted seeking an advancement in the *process* of software development through a new process, new procedures, or activities and is executed in accord with the requirements of IC86-4R3, the project is eligible as SR&ED.

Problem Statement: The claimant needs to cite a problem, show the uncertainty, and identify the alternatives considered. The claimant should then show the state of the process or product before and after the advancement.
These claims may deal in the area of software tools, compilers, utilities, procedures and techniques that contribute new knowledge or capabilities to software practitioners

Discussion: The development of software tools, compilers and utilities can be claimed as projects like any other and as such present no unusual problems. In this issue sheet we address the eligibility of projects aimed at revisions to the development process itself.
The software development process is in a constant state of flux. Changes are being made every day, but many of these are not the result of uncertainties and systematic investigations leading to advancement and for that reason are not eligible under IC86-4R3. IC 97-1 states in Sec. 1.3 "The eligibility of work as SR&ED is evaluated in terms of the process of performing SR&ED for the purpose of scientific or technological advancement." and " Software development can be eligible as SR&ED on the basis that it aims to advance computer science or information technology."
The claimant should clearly state the new functionality or performance required from an identified software development process to be developed or enhanced. This should include a statement of the state of the technology within the industry, at the start of the project. Where the claimant develops a new methodology, it is essential that he describe the systematic process used to resolve uncertainties and gain the advancement in the methodology. Good records of the process help substantiate the claim. An explanation of how the claimant's solution is new or different is essential.

Simply using a new management process or new management tools to develop software does not qualify a project.

Since the onus is on the claimant to show that an uncertainty and an advancement exists, the claimant should be able to cite the technological base from which he claims an uncertainty and an advancement. The claimant should provide a credible reference as evidence of this base. The technological base described by the claimant may include descriptions of performance, throughput, stability, concurrency, step response, MTBF, and footprint.
Part of the development process is adequate testing. When developing tools for use by others it is frequently necessary to have others or 3rd party developers (possibly clients) perform testing to ascertain whether the stated advance in technology has been attained. If this testing is essential to the advancement, has been done in Canada and has been paid for by the

claimant, the costs are eligible.

To develop a robust project technical description the claimant should show that the work advances knowledge. The description should identify the technical alternatives investigated, the design, time, technological, or physical constraints, the limitations in tools and processes, the platforms considered, the systematic process, the testing performed, and the technical design trade-offs. Good record keeping of these activities is essential to good submissions.

Example 1

We seek to investigate alternatives and test the methodologies identified to develop a unique system documentation tool. XYZCo established a set of requirements for a documentation system that included functional and performance objectives. In a review of software tools literature and discussions with likely vendors, XYZCo found that there are no commercially available tools to accomplish this task. Both the vendors and the literature identified significant obstacles to developing such an application.

XYZCo identified and investigated a number of alternative approaches and developed a series of tests to evaluate the alternatives and select the optimum approach. The alternatives and the tests were described briefly in their project description. Despite the fact that industry experts had identified the task as difficult or impossible, XYZCo developed a system documentor. In order to confirm that the stated advancement had been achieved, they tested the new application on their own source listings to see if they could reliably achieve their functional and performance objectives. To further confirm their success they worked with some of their clients to test the documentation tool against a more representative sample of program source code.

Example 2

We seek to investigate alternatives and test these to see if we can develop a unique compiler that meets our performance objectives. ACME Software has determined through a market study that there is a need for a compiler that will compile objects developed in C++ to run on a new embedded controller. They determined through a literature search that objects have never been successfully compiled to embedded controllers, but their computer scientists have some ideas as to how this might be accomplished. They would achieve a software first if they were to accomplish this task, but they must do so with acceptable performance of the embedded controller.

ACME set up a team to investigate and test the alternatives. They eliminated some alternatives through testing and in the process they identified some new algorithms that allowed them to compile workable objects into the footprint allowed by the embedded controller. Testing of the embedded controller showed that they were not able to achieve their performance objectives.

Part 3. Guidance on Issues Associated Developing the T661 with Documentation and Evidence

Guidance to Constructing Effective T661 Software Project Descriptions

1.0 Background Material:

Objective:

The purpose of this document is to assist claimants engaged in development of software based technologies to prepare effective SR&ED T661 project descriptions and identify other evidence in support for said SR&ED claims. The aim is to construct claims which contain sufficient information to allow CCRA to effectively understand at a high level, the nature of the project from a technological perspective, the key or major technological challenges faced; the nature of the key major work elements and their relationships; and the boundaries of the claim.

Background:

The primary interpretation policy for the SR&ED program as embodied within the Income Tax Act (ITA) is Information Circular 86-4R3. IC 86-4R3 establishes the three criteria for eligible SR&ED, i.e. technical or scientific advancement, technical or scientific uncertainty and technical or scientific content. Content includes evidence of a systematic program of investigation through experimentation or analysis However little guidance has been provided on specific documentation and evidentiary requirements. Further, concern has been expressed by claimants, particularly in the area of software development, that the documentation required by CCRA to support an SR&ED claim has required additional resources, as the information that is produced in the course of the work has not been readily recognized by CCRA as evidence of eligible SR&ED .

This document is intended to provide a framework to documentation that meets the requirements of the SR&ED program, as contained in the legislation and interpreted in the Information Circular. It also responds to claimants' needs for practical solutions to the requirement that utilise the contemporaneous documentation or work product that is naturally produced in a software development environment as evidence of SR&ED. The other software issue sheets in this series, are intended to be used in concert with this document. This document is to assist the claimant in constructing effective T661 project descriptions which explain the merits of their project's eligibility, as required within the context of the SR&ED program.

Intended audience:

This document is directed towards qualified software development practitioners, software engineering managers, CCRA technical reviewers, and tax accountants.

Related Documents

This document must be used in conjunction with the following two companion documents;

1. Sufficiency of Supporting SR&ED Evidence for Software Claims.
2. Guidance to Sources of SR&ED Evidence for Software Claims.

In addition the other 1999 issues sheets in this series that have been developed under the auspices of the SR&ED software sector working group also provide valuable insight.

2.0 Overview of the T661 Project Description Requirements:

A T661 Project Description of an SR&ED project, contains the descriptions of the 3 requisite elements of a valid SR&ED claim. These are;

1. Description of the Technological Objectives & Technological Advancements sought or achieved.
2. Description of the Technological Uncertainties anticipated or encountered.
3. Description or explanation of the systematic experimental approach followed in the development process and how it is applied in the context of technological advances being sought or the technological uncertainties that were encounter in the project. This include an explanation or outline of major/key work elements or activities undertaken in the fiscal year to overcome the uncertainties and to achieve the advance.

All work product created by Software Engineering/Computer Science based activities is embodied in the form of documentation. The root or lowest form of such documentation which encapsulates the total solution, the effort, and the process undertaken, is found in the source code that is created in the development process. Additionally, the individual instances of source code revisions that occur during the development process, as well as source code written for supporting functions such as build scripts and test programs, form the underlying foundation of evidence to support the existence of systematic experimental process. The source code represents the elemental evidence for the existence of a given technological advance and the corresponding record of the elimination of a technological uncertainty. However in most cases the evidence is usually represented within the source code in a highly diffused or diluted fashion. As such the level of effort required on the part of both the claimant and CCRA to respectively convey and interpret evidence provided in the form of source code is extremely high. Although source code is an eligible form of supporting documentation, since the onus to demonstrate eligibility is borne by the claimant, **most successful claimants do NOT view source code as an effective medium for conveying eligibility.** Successful claimants rely on more focused (albeit indirect) higher level forms of documentary evidence to support a given SR&ED claim.

In most cases, the ability to recognize and interpret relevant evidence which arises naturally within the context of a software development process is highly dependent on the direction provided within the T661 Project Description. The construction of effective descriptions in the T661 pertaining to the technological advancement sought and technological uncertainties encountered, provides the fundamental framework for interpreting other evidence arising from the documentation and work product created within the project. The T661 project descriptions provide the binding medium which, when combined with the actual work product of the project, allow a knowledgeable software practitioner to make a determination of eligibility for a given SR&ED project. The T661 Project Description is a critical documentary component of the claim, which provides the boundary set and "technological lens" through which other work product (evidence) can be examined and from which eligibility is determined. The T661 is not meant to be an exhaustive treatise on why a project or a component of a project is eligible. It should establish the boundary set of the advancements and objectives which demonstrate the claimant has undertaken R&D which is not of a routine nature. The T661 project description is meant to be a very short, concise narrative illustrating key elements of a project which are pertinent to understanding how the work fits into the domain of the SR&ED program. As such the T661 Project descriptions forms a high level summary of the claimant's expert opinion supporting a given project's eligibility within the SR&ED incentive program. The T661 is not meant to be an exhaustive treatise on why a project or a component of a project is eligible. It should establish the boundary set of the advancements and objectives which demonstrate the claimant has undertaken R&D which is not of a routine nature. The T661 project description **is meant to be a very short, concise narrative illustrating key elements of a project which are pertinent to understanding how the work fits into the domain of** the SR&ED program. **As such the T661 Project descriptions forms a high level summary of the claimant's expert opinion supporting a given project's eligibility within the SR&ED incentive program.**

Thus when constructing a T661 Project Description for a software based R&D effort, the claimant is recommended to:

1. Identify or categorize the research and development context in which the work was undertaken.

2. Write the descriptions with clear English, avoiding or defining jargon as it is used. Construct the descriptions from the technology perspective rather than the business perspective.
3. Maintain a consistent frame of reference with respect to the key aspects of eligibility: advance, uncertainty, and the way in which the activities in the fiscal year act to overcome the uncertainties so that the advance may be achieved.
4. Identify the start and expected or actual duration of the project.
5. Identify the staff working on the project and their respective professional qualifications.
6. Describe the progress made on the project within the fiscal year covered by the claim.
7. Focus on describing the software technology advancements you are trying to achieve rather than products and/or features.
8. Focus on clearly identifying and articulating the constraints which create technological uncertainties.
9. Utilise the T661 descriptions to concisely summarize the essence of both the key technological uncertainties encountered and the key technological advancements sought.
10. Use the T661 descriptions to identify a collection or subset of **key** work products (evidence) or type of work products which provide direct or indirect support of both the eligibility (for example: demonstrate uncertainty) and size of a given SR&ED claim.
11.

In the following sections of this document there are a series of questions intended to facilitate the claimant in conveying the essence of their claim framed in the terminology of software development.

3.0 Identifying Technological Advancements:

The following questions have been provided to facilitate the construction of an effective T661 description pertaining to the technological advancements sought by the SR&ED project in question.

Users of this document (claimants, CCRA technical reviewers, and software practitioners) are cautioned that such questions are not to be utilized as a template for compliance either item by item or in aggregate. These types of questions are intended to facilitate the creation of effective project descriptions. The objective here is to outline options for developing sets of questions which may act as catalyst to provide an effective and efficient method of identifying key evidence of eligibility. The resulting T661 content may contain a complete absence or response to the questions proposed , with another set of facts being used to construct the project description.

What is the current state of technology?

1. Is this project a continuation of previous work by your firm. If so in technological terms describe what you technology is currently capable of prior to this SR&ED activity?
2. What is the state of the common body of knowledge with respect to this technology at the start of the project?
3. What did you do to find out about the current state of the technology? Do you have access to the data and know how ? What restrictions to access of technical data were present in determining the state of the art?

What is the advance you are seeking?

1. What are you attempting to achieve technically? Where are you trying to take the technology?
2. What did you do to improve the state of the technology? What process was required to improve the technology? How did you measure the improvement, what metrics were used?
3. How will you define & measure success?
4. How will you identify technical conclusions?
5. What did you learn from this project (or want to learn from this project)? What IP came out of this project or do you expect to arise from this project? What did you technical knowledge did you learn from this project that you do not want your competitors to know about?
6. What would you do differently if you had to do it again?

7. What are you proud of in terms of technical achievement in this project?
8. What do you hope to learn from a technical perspective?

How does the advance you are seeking clearly advance the current state of technology?

1. What technological reason if any has prevented you or someone else from attempting the same thing previous to this development effort?
2. What is out there and how does/will your resulting technology compare/differ to other companies?
3. Did your Technological objectives change over the course of the project if so how & why ? What were you unable to get working?

4.0 Technological Uncertainty:

A second component of the eligibility test which must be met in order to form a valid SR&ED claim is based on the existence of technological uncertainty being present and the activities undertaken to attempt to overcome said uncertainties through research, analysis, or experimental development. Software technologies are by their very nature a collection of abstractions ensconced within a complex collection of rule sets. Over time software technology has evolved to the point whereby these rule sets have now become so large, and the relationships so complex, as to no longer allow a developer or group of developers complete control over their environment. As such the compounding of large complex technology components presented by the modern software environment have created immutable objects which present themselves, and affect environments, in a manner analogous to the complexity of the physical systems found in nature. When examined closely it can be seen that concept of technological uncertainty as described within the ITA and interpreted in circular 86-4R3 is also rooted in the constraints presented by the rule sets dictating the behaviour of physical systems. Thus it follows that identifying the fundamental technological uncertainties associated with a given software based SR&ED project, can be achieved by examining the constraints provided by the software environment within which the technological advance will be sought.

The following questions are intended to facilitate the construction of an effective description for technological uncertainties pertaining to a given SR&ED project;

Users of this document (claimants, CCRA Technical Reviewers, and software practitioners) are cautioned that such questions are not to be utilized as a template for compliance either item by item or in aggregate. These types of questions are intended to facilitate the creation of effective project descriptions. The objective here is to outline options for developing sets of questions which may act as catalyst to provide an effective and efficient method of identifying key evidence of eligibility. The resulting T661 content may contain a complete absence or response to the questions proposed , with another set of facts being used to construct the project description.

1. Identify the limitations/constraints imposed by the technology components being utilized. What technical challenges did these constraints create?
2. Identify the degree of control the claimant has to modify the technology components. What technical challenges did these constraints create?
3. Identify the constraints or uncertainties or paradoxes presented when certain components/objects/technology platforms are operated in conjunction with other software entities. Do you have control over these interactions, can you or the vendors of these components predict the effects of these interactions?
4. Identify any constraints resulting from considerations of;

- Inter-operability
- Conformance to standards
- Performance (step response, throughput)
- Concurrency
- Footprint
- Scale-ability

- Stability
- 3rd party components
- legacy requirements

What technical challenges did these constraints create?

5. Identify any key characteristics of a technology platform you are using to which the manufacturer of the technology component cannot provide a fully deterministic characterization of the platform when utilized in the fashion required by your project.
6. Is the integrated performance of the software components incorporated within the project fully deterministic? I.E. can the behaviour of the components be fully projected both on a stand alone basis as well as when operating within an integrated environment. Can you predict the desired outcome? If not why not ?
7. What technology risks/constraints/problems appeared after the project began?
8. What was or will be hard or technically difficult to do & why ?
9. If you had to do it again what would you do differently?
10. What restrictions are presented by the attributes of objects/components or the API's presented by components on environmental platforms such as operating systems?
11. What technical alternatives did you look at, what did you discard & why?
12. Are you using any of the components in a unique, previously undocumented or unconventional fashion, is the vendor able to confirm the suitability of these components for use in said fashion. Is the vendor capable of providing a deterministic description of the components predicted response when used in this unique fashion?
13. What are the possible technical outcomes other than the results you are seeking ?
14. What if any technical alternatives have you investigated? What are the technical design trade-offs associated with these alternatives?
15. Is the problem commonly recognized or being encountered as a critical challenge by other developers and other companies trying to do something that is similar? or it is commonly being encountered as a constraint when using a specified development tool or common platforms etc. as a starting points?

5.0 Systematic experimental approach and key work elements

In addition to outlining the technological uncertainties encountered and the advancements sought, the T661 provides an overview of the systematic experimental approach or processes used to address these technological issues. Such a summary would provide an outline of the SR&ED eligible work. Depending on the development environment, such an outline may need to be drawn from within the context of a potentially larger development undertaking. In particular such a summary will identify the major effort used to address the technological merits of the claim, as well as identifying likely sources of supporting documentation. A more detailed treatment of potential sources of SR&ED documentation can be found in the companion document entitled "Guidance to sources of SR&ED evidence for software claims." Additional information on required sufficiency of evidence can be found the companion document entitled "Sufficiency of supporting SR&ED evidence for software claims".

Guidance to Sources of SR&ED Evidence for Software Claims

1.0 Background Material
2.0

Obective
The purpose of this document is to;
i) Aid claimants and CCRA Technical reviewers in identifying reasonable and likely sources of evidence (documentation & work product) which are likely to arise naturally from software development processes, and to utilise said documentation and work product as the basis of evidence to support an SR&ED claim.
ii) Identify low overhead processes and practise suggestions which if implemented should increase the focus and effectiveness of naturally arising documentary evidence supporting a given SR&ED claim.

Background:

The primary interpretation of the SR&ED program as embodied with the Income Tax Act (ITA) currently used by CCRA and accepted by the Courts is contained within the CCRA information circular 86-4R3. IC 86-4R3 establishes the three criteria for eligible SR&ED, including technical or scientific advancement, technical or scientific uncertainty and technical or scientific content. Content includes sufficient documentation and evidence of the systematic program of investigation through experimentation or analysis that has been confirmed as an essential element of SR&ED. However little guidance has been provided on specific documentation and evidentiary requirements. Further, concern has been expressed by claimants, particularly in the area of software development, that the documentation required to support an SR&ED claim requires additional resources, as the information that is produced in the course of the work is not recognized by CCRA as evidence of eligible SR&ED activities. This document is intended to provide a framework for documentation that meets the requirements of the SR&ED program, as contained in the legislation and interpreted in the Information Circular and that also responds to claimants' requests for practical solutions to the requirement to produce contemporaneous documentation of SR&ED in a software development environment.

Intended audience:

This document is directed towards qualified software development practitioners, software engineering managers, CCRA technical reviewers, and tax accountants.

Related documents:

This document should be read in conjunction with the following two companion documents;

1. Guidance to Constructing Effective T661 Software Project Description
2. Sufficiency of Supporting SR&ED Evidence for Software Claims

In addition the other 1999 issues sheets in this series that were developed under the auspices of the SR&ED Software Sector Working Group also provide complimentary insight.

3.0 Documentation & Work Product (Evidence):

The work product and the **content** of documentation produced while undertaking an SR&ED eligible software based development project and which is identified within the T661 Project Description, form the foundation of documentary evidence supporting the existence of an eligible claim. In addition the evidence to support the assertion by the claimant that a systematic investigation and experimental development activities were undertaken must also be identified.

Claimants are reminded that it is the content within, not the existence of a given document or the amount of documentation that provides evidence of eligibility. In cases where the documentation arises naturally from software development methodologies applied to a given business project, the claimant will need to identify specific components of the documentation which pertain to the eligibility of the SR&ED claim.

Reviewers are reminded that it is a common and reasonable expectation for the components of eligible SR&ED claims be ensconced within the larger framework of a commercially driven development project. Reviewers should utilise the T661 project description provided as an aid to both focus on, and interpret the relevant documentation which arises naturally from the claimants software development process. Likewise, it is the responsibility of the reviewer to understand how this indirect evidence, when identified and appropriately organized by the claimant, can demonstrate eligibility. As discussed in the companion document entitled "Guidance to constructing effective T661 Software Project descriptions", the nature of the evidence contained within the naturally arising documentation is likely to be of an indirect nature which requires appropriate interpretation from the reviewer using the context ("technological lens")provided by the T661. In this respect the T661 project descriptions serve to direct reviewers to specific aspects of the content or the interpretation of the content with respect to SR&ED eligibility.

The documentation and resulting work product can be likened to the technical/science equivalent of a "receipt" found in financial accounting terminology. Such a "receipt" provides the **evidence** to show that SR&ED conditions existed and the "expenditure" was made in conducting the SR&ED eligible activities in accordance with the requirements of the ITA.
Additionally the scope of the effort and the actual expenditure of funds with respect to material, labour and capital directed towards SR&ED activities needs to be captured. These accounting records form the evidence for the financial component of the SR&ED claim.
As previously discussed work product and documentation kept which most effectively convey eligibility are themselves composed of indirect evidence per se. Additionally it is quite common for indirect work product (evidence) which arises naturally from product development efforts, to be largely constructed from the perspective of managing the project, NOT the technologies, and in that regard often need to be interpreted by a qualified software development practitioner(s) with the assistance of the T661 Project Descriptions to determine SR&ED eligibility.
Users of this document should examine closely the companion issue sheet *"Sufficiency of Supporting evidence for Software claims"* **when reviewing this issue sheet on the sources of documentation. It is not how much documentation you have or the amount of detail but how effectively higher level or forms of documentation are used in the validation of claims that makes for efficient and reasonable processes of validation.**

2.1 Evidence and Development Environment:

The nature of the documentation & work product (evidence) which arises naturally from a given claimant's development environment is predicated on the size, maturity, development methodologies used by the company. At least 4 distinct project environments have been identified with respect to the nature of and composition of evidence likely to arise from a given environment, these are;

1. **Distinct well defined individual SR&ED projects** - These projects have a very specific start and finish with narrowly defined technological objectives and uncertainties. While these are not the norm within the industry they most readily map to the specific requirements of the SR&ED ITC program.
2. **Dedicated R&D environments**-These are environments whereby the focus of the a defined group of personnel is focused on R&D for new products or processes usually to exclusion of all other corporate activities. Examples of these environments would be dedicated research departments of large corporations.
3. **Early stage/ Start-up companies**-Research & Development activities undertaken by early stage/ start-up companies usually prior to initial revenue generation by the product in question.
4. **Large and/or multi-year projects**- These are usually performed by an R&D group in concert with a variety of other non eligible activities required to deliver new or improved products or services within the business context of the claimant. Such projects require the claimant to identify the eligible work within the context of an entire project. Reviewers are advised against de-constructing multi-faceted projects themselves in an attempt to identify eligible work at the activity level; this is not the appropriate level at which to decide eligibility. Individual activities are not eligible or ineligible in themselves; it depends why they were performed. Routine activities may be a legitimate part of the SR&ED project if necessary as support work. It is for the claimant to show which parts of the work were integral to the pursuit of particular technological objectives. claimants extracting SR&ED eligible expenditures from a larger aggregate body of work must be prepared to provide supporting material with respect to extraction, both with respect to technical content as well as financial expenditures.

2.2 Documentary Matrix:

The following is a mapping of some of the common potential sources of naturally arising work product which provide evidence that can be examined, and with the assistance of the T661 project descriptions may be used to assist in determining the eligibility of a claim:

Users of this document (claimants, CCRA Technical Reviewers, and software practitioners) are cautioned that the list of work product identified here should NOT be viewed as a template required for compliance either item by item or in aggregate. The purpose is to identify options for potential sources of evidence not mandatory sources. And how such evidence supports the 3 basic criteria required for SR&ED eligibility not the quantity of evidence. Such evidence is often of an indirect form. Users are directed to the other 2 companion documents for further discussion of what is sufficient and the role of T661 project description.

The objective should be to identify the best highest level kinds of evidence and ways to retain it that is consistent with claimants development methodology.

The table below provides a mapping of typical sources of documentation likely to be available as a function of corporate size and development environment;

	Small Direct	**Dedicated Large**	**Start-up/early stage**	**Aggregate Large/ Multi- Project**
Technological Advancements	3rd Party documentation Test Results Source Code	3rd Party Documentation Test Results Test Plan Source Code	3rd party Documentation Test Results Source Code	3rd party Documentation Test Results Test Plans Source Code
Technological Uncertainties	Email 3rd party Documentation Architectural Design Docs Vendor Correspondence Personnel Qualifications	Email 3rd party Documentation Defect Tracking Logs Development Plans Architectural Design Docs Vendor Correspondence Personnel Qualifications	Email 3rd party Documentation Vendor Correspondence Personnel Qualifications	Email 3rd party Documentation Defect Tracking Logs Vendor Correspondence Architectural Design Docs Personnel Qualifications
Systematic & Experimental Development	Email Project Plan Project Schedules Source code rev control logs	Email Software Development-Methodology Standards Project plans Project Schedules Design Reviews Source code rev control logs Configuration Management	Email Project Schedules Source code rev control logs	Email Software Development - Methodology Standards Project plans Project Schedules Design Reviews Source code rev control logs Configuration Management
Expenditure Size	Project Plan Project Schedules Resource Allocation	Project Plan Project Schedules Project Based Time sheets Resource Allocation	Project Schedules Resource Allocation	Project Plan Project Schedules Project Based Time sheets Resource Allocation

2.3 Sources of Documentation & Work Product

The following list of potential sources of documentation and work product which may arise naturally from a variety of development methodologies, and which may provide evidence supporting SR&ED eligibility. To assist the claimant an effort has been made to list the descriptions of the various forms of documentation in an approximate order of effectiveness and cost efficiency (from most to least). **The ranking of documentation here is not meant to be definitive but rather to provide a generic guide. No attempt should be made by either the claimant or the CCRA to infer an absolute order of precedence with respect to the relative strength or the efficiency of evidence sources identified.**

2.3.1 Developer's Journals

Lab books, personal day timers, electronic journal entries, or any other form of short form individual notation by technical staff can be used as supporting documentation. Entries are usually made in "real time" on a periodic (daily or weekly) or an event driven basis (when something important has occurred). The entries are usually very brief often consisting of just a sentence or two per entry. When utilized this type of documentation can provide an efficient and highly cost effective method of generating supporting evidence for SR&ED claims, especially valuable if some entries are dated Depending on the manner in which these records are entered and kept, they may or may not require a fair amount of effort to sort through the information and present the appropriate records to the reviewer. [**Evidence potential** - technological uncertainties, technological advances sought, systematic experimental development]

2.3.2 Architectural Design documents, High Level Design documents

Documents of this nature typically provide a "structural overview" of the project's objectives from a purely technological perspective. As such these types of documents are usually quite efficient at providing evidence pertaining to the technological advances being sought and the associated technological uncertainties. Most architectural design documents provide some level of analysis with respect to the alternatives considered, as well as providing a foundation or rational with respect to the constraints within which a solution is being sought. The resulting architecture (or architecture sought) may be the advance in and of itself. [**Evidence potential** - technological advances sought, constraints & technological uncertainties present.]

2.3.3 Design Review Minutes

Design reviews are often conducted to examine technically critical or risky aspects of a given software development effort. As such the minutes recorded from these reviews often provide rich technically based commentary from which to extract supporting evidence of technological uncertainties present or technological advances being sought. Such information may also contain commentary with respect to design iteration, resulting from analysis of failures and successes arising from test activities. Design reviews can readily provide a large amount of quality supporting documentation if they are constructed with a view to capturing the SR&ED component . Often it is easy to add to the design review document, quick discussions of advance, uncertainties remaining and solved, and activities complete and planned. If these are created regularly throughout the project and are dated documents they may be sufficient for the technical review. Note that non-traditional formats, eg. photographic or digital records of white board information, are acceptable. [**Evidence potential** - technological advances (not) achieved, technological uncertainties present, systematic experimental development.]

2.3.4 Written Correspondence

Correspondence of all forms (Letters, Email, Fax, memos) between team members as well as external software technology vendors can provide a wealth of evidence. In particular correspondence with vendors can often contain evidence of state of the art, or confirmed technological uncertainty. Team members' correspondence can provide evidence of a systematic experimental process, as well encapsulate technological objectives and advances sought. [**Evidence potential** - state of the art, technological advances (not) achieved, technological uncertainties present, systematic experimental development.]

2.3.5 Performance Requirement Specifications

These specifications may be presented as stand alone or as part of a Product Requirement Specifications. The performance requirements can in certain projects be useful in identifying a series of key technological objectives and advancements sought, as well as illustrating the constraints which cause technological uncertainties to be present within the project. [**Evidence Potential** - identification of technological uncertainties, technological advances sought]

2.3.6 Test results

Work product arising from the execution of appropriate test activities can provide both direct and indirect evidence of pertaining to eligibility. Test results may include output from 3rd party performance and test tools, as well as output from customized test programs. In order to provide direct evidence the tests undertaken would need to be specifically designed to measure the technological advancement sought. Depending on the nature of the development environment, it is just as likely that the results being measured would be feature or performance based. The resulting evidence provided by the test would be of an indirect nature. In such cases it may be helpful to refer to the T661 project description to provide the framework when determining the applicability of the test results as evidence for technological advancement. A series of applicable test results would provide evidence of a systematic experimental approach. Test results can also provide both direct and indirect evidence of the occurrence of unforeseen technological uncertainties. [**Evidence potential** - technological advances (not) achieved, technological uncertainties present, systematic experimental development.]

2.3.7 3rd Party Documentation

This category relates specifically to technical documentation which provides both summary and detailed information with respect to software technology components being purchased and utilized by the claimant within a given software development project. This can be the results of a review of competing technologies or theoretical solutions to technical problems. In the context of modern software environments, this documentation is often the basis for evidence in determining both the state of the art and accordingly the basis for the technological advances being sought. In addition it is quite common for this documentation to provide evidence of the constraints present which form the basis of technological uncertainties. This documentation often contains API specifications standards for the software components in question as well detailed descriptions of the pertinent internal architecture and mechanisms embodied by the technology. [**Evidence potential** - state of the art, technological advances sought, constraints & technological uncertainties present.]

2.3.8 Defect tracking records

Defect tracking systems are most commonly deployed in more mature development environments. When fully integrated defect tracking systems are in use, the data stored during a project can provide summary evidence of technological uncertainties encountered, failure to

achieve a particular technological advancements, evidence of successful advancement as well as evidence of systematic experimental development. [**Evidence potential** - technological advances (not) achieved, technological uncertainties present, systematic experimental development.]

2.3.9 Test plans

Test plans can provide evidence which identifies or relates to technological objectives and corresponding advances sought. Test plans can also provide evidence of a systematic experimental development process. In cases where test plans are focused on feature based metrics, the plans may when examined in conjunction with the T661 descriptions continue to provide relevant information with respect to SR&ED eligibility. [**Evidence potential** - technological advances (not) achieved, systematic experimental development.]

2.3.10 Project Schedules & Resource allocations

Project scheduling and resource allocation tools typically provide a concise indication of the overall development process undertaken. Depending on the development environment these types of documents may encompass all of the activities (both eligible and ineligible) undertaken with respect to a given development initiative. In such cases these documents can be used in concert with other documentary evidence (such as revision control logs and test plans) to establish that a systematic & experimental process was undertaken with respect to the SR&ED objectives. In instances where the scheduling tool was actively used to track actual versus planned progress, the documents are useful in identifying or confirming the occurrence of unforeseen technological uncertainty. Resource allocation information recorded within the project schedule, and associated with specific tasks can also provide useful evidence in identifying the size and expense of the work undertaken. Resource allocation may be clearly outlined in dated organizational charts that clearly delineate resources by project, or development work from routine work such as routine software maintenance or routine screen customization. For environments where development methodology standards or project plans are the norm, Project schedules can commonly be found as a sub components of those master documents. For other environments these documents are usually maintained as independent entities. [**Evidence potential** - existence of systemic, experimental process, commencement & completion of SR&ED activities, unexpected technological uncertainties encountered, expenditure size]

2.3.11 Product Requirement Specifications

While the Product Requirement Specifications usually address the market or functional requirements, examination of these documents from a technological perspective may be useful in extrapolating the technological objectives embodied within the project which are required to attain the functionality described. This document usually requires the context provided by the T661 project descriptions in order make an examination of the document useful in identifying corroborating information with respect to determining technological advances being sought. [**Evidence Potential** - identification of technological advances sought, or technological uncertainties present]

2.3.12 Source code

As previously discussed while source code provides the root evidence of technological advances achieved, it is generally not considered an effective medium for conveying eligibility. Within the context of an eligible SR&ED project and with significant effort on behalf of the claimant and the CCRA technical reviewer, source code can provide evidence of technological advances being sought in the presence technological uncertainty. When relying primarily on source code for evidence, the presence of a complete revision history and

incremental archiving of intermediate revisions of source code is very useful to the establishment of technological advance through the use of experimental development. [**Evidence potential** - systematic experimental development, technological advances sought, attempts to overcome technological uncertainty]

2.3.13 Logs & comments from source code revision control systems

These documents generally provide evidence of the use of a systematic and experimental development process. In addition an examination of the entries accompanying a given revision usually provides a synopsis of the incremental improvements or changes which have been made since the last version. This information can in some cases provide evidence with respect to attempts made to overcome specific technological uncertainties, or to verify specific technological advances being sought. [**Evidence potential** - systematic experimental development, technological advances sought, attempts to overcome technological uncertainty, expenditure size]

2.3.14 Project Plans

These are often found in development environments which have not yet adopted a global development methodology. These plans typically outline the process upon which the software development effort will be based. These types of documents are also prevalent in more mature environments where a conscious decision to deviate from global standard has been made. As with their global counterparts such documents usually provide a summary of the planned process and resulting work product documentation which is likely to be available. Deviations midway through a project from a planned process due to reasons of a technical nature can be an indicator of the emergence of an unforeseen technological uncertainty. [**Evidence potential** - existence of a systematic, experimental process, identification of documents which contain potential evidence of technological advances sought, and unforeseen technological uncertainty, expenditure size]

2.3.15 Configuration Management Documentation, Build plans & Build files

These documents generally provide evidence of the use of a systematic and experimental development process. In addition they may also provide additional supporting evidence with respect to the nature of advancements being embodied within a particular build or release. [**Evidence potential** - systematic experimental development, technological advances sought]

2.3.16 Software Development Methodology Standards

These are documents which formally outline the methodology or process upon which any given software development project within the claimant's operation will be guided. These types of standards usually dictate a series of other documents and process milestones to be recorded. Identification of this type of document provides significant evidence of a mature scientific research or experimental development processes being utilized. These documents usually provide a road map to the majority of other key work product entities generated within the development process. [**Evidence Potential** - existence of a systematic & experimental process]

3.0 Documentary Evidence Case Study:

This is an example of how the evidence necessary to support the claim typically arises naturally from the standard documentation and work product of a given software development effort;

A large container rental company is developing a custom, geographically distributed, transaction based, enterprise wide, operations, reservations, billing, and inventory yield management system. The new system will replace an ageing and simple UNIX terminal based main frame reservation and contract recording system.

Prior to undertaking the project the claimant retained an independent consulting firm to find a 3rd party vendor from which they could purchase a system with the required functionality. Ultimately the consultants made the recommendation to develop rather than purchase a system. [**Evidence** - pertaining to the state of the art is the report examining the solutions available and the recommendation to make versus buy.]

The claimant did not have the internal development expertise to necessary to design and implement the new system, and consequently subcontracted a respected Canadian software development firm to undertake the project. [**Evidence** - retention of such a development firm provides evidence of access to qualified personnel which in turn relates directly to the validity of advancements sought and uncertainties encountered.]

The new system architecture was implemented utilizing object oriented software technology components in an N Tier thin client configuration. The functional requirements with respect to transactional, reporting, and yield management processes for the system resulted in the requirement to support very complex transactions. This in turn required the implementation of a very large and complex database schema.

The nature of the problem appeared in the later half of the project as result of unexpected interactions between the transaction server component technology and the SQL database technology. Within the given architecture the two components combined to constrain the manner and mechanisms related to the level of granularity at which the SQL database could undertake record locking within a given table(s). The end result was an unexpected and severe impairment with respect to both the concurrency and throughput as it pertained to the processing of transactions.

The development team undertook a series of testing and corrective actions but was unable to isolate the root cause for the combined interaction behaviour of the database and transaction server technology components. The development team contacted the vendor of the components (which in this case was common to the Operating system, SQL database, Transaction server and Internet server software technology components) and requested assistance with the problem. The vendor investigated the problem and made several recommendations in an attempt to solve the problem, but was unable to direct the company to a solution to the problem. None of the directives from the vendor were able to correct the system performance. In fact during the course of the investigation the vendor was unable to accurately predict the resulting system performance with respect to several of the suggestions they made.[**Evidence of the technological uncertainty** - test plans, test logs, test programs, defect tracking records, email correspondence with vendor(s) relating to the problem]
The development teams continued to utilise a series of prototypes and experimentation to empirically characterize the behaviour of the system in order to gain further insight into the problem. Subsequently 3 experimental solutions were prototyped. Each of the potential solutions was then implemented and tested. The solution which showed the most improvement was then further refined using another series of experiments. [**Evidence of systematic experimentation** - the test prototype programs, test plans, test results, emails, & defect tracking entries.]

The final solution resulted in the utilization of a combination of a series of unorthodox connection pooling and directed record locking techniques. **[Evidence - of the Technological**

Advance - final source code, test programs, test results, emails to/from vendor.]
This case study is an example where only a portion of the development activities for the entire project would be eligible under the SR&ED program. Specifically the project eligibility commenced at the time when the concurrency and performance issues were first identified. Since the project contained a large amount of routine software engineering tasks, only those tasks directly related to (including the routine tasks performed directly in support of) the analysis, experimental prototyping, coding, testing and validation of the solutions associated with the performance and concurrency issues would be deemed eligible. Eligible activities ceased upon successfully overcoming the performance and concurrency problem. In this case the claimant would be required to identify or extract the expenditure size of the SR&ED component from the balance of the business project. For both the technical and financial components of the project, the claimant bears the burden of providing the documentary evidence to identify and substantiate the relative size of the SR&ED component of the project as compared to the overall project costs.

4.0 Development Process and Practise Options:

This section of the document attempts to identify options for simple low overhead practices which could be deployed within most of the software development methodologies currently in use within industry, with a view to improving the clarity of the evidence available to support a given SR&ED claim.

4.1 Development Process

1) **Birth certificate** - Upon commencement of a project, write a “birth certificate” for the project outlining the technological advancements which will be needed and the technological uncertainties faced (I.E. the basic components of the T661; a draft T661 project description filled out and dated). Often the perspective required to clearly articulate the issues with respect to technological uncertainties faced at the outset of a project are eroded by the passage of time.

2) **Status Report** - Throughout the project, on a regular basis (weekly, monthly, quarterly, or at development “gates“ depending on the size of the project) prepare a status report update. This may take several forms:
   - a regular updated draft of the T661
   - design reviews that cover the technological criteria for eligibility
   - a status report designed to meet both your business requirements and the requirements for SR&ED eligibility

   Each of these forms may be customized to cover some of the costing support needs as well as technical eligibility including: resource usage and resource planning. Prepared at the beginning of the project, and throughout the project, these may be sufficient documentation for supporting your SR&ED claim as well as providing key management information for your business.

3) **SR&ED Tagging** - Within your software development process introduce the notion of SR&ED tagging. Tagging is the simple act of having team members simply identify in real time information they receive or create which MAY have significance to the SR&ED program or the components of a given program. The emphasis here is on identifying existing documentation or components thereof rather than the creation of additional

documentation. Create a separate common index file to hold the names of all documents which team members have created and which contain specific information pertaining to SR&ED eligibility. As team members create or edit project documents related to SR&ED they can update the index.

4) **Defect tracking records** -When constructing the classification descriptors for a defect tracking system database incorporate an attribute allowing the issues entered to be "tagged" as SR&ED. This will allowed the SR&ED related defects to be identified and extracted.

5) **Email** - For each project which is SR&ED eligible or which has SR&ED eligible components have a common email account created in which correspondence with SR&ED significance can be "CC'd". When team members receive or send email which has information relevant to a given SR&ED issue the member simply forwards or CC's a copy to this email address. The result is a single repository of all SR&ED related email for a given project.
6) **Prototype Archiving** - where possible archive a fully working copy of each significant key prototype, complete with execution environment and test data. Particular emphasis should be placed on saving specific software revisions which address the major issues associated with the SR&ED components of a given development project.

4.2 Expenditure Tracking

A significant component of the documentation for a given SR&ED project is that which substantiates the size of the expenditures claimed. In general the evidence sought to substantiate given SR&ED claim from a financial perspective can usually be separated into two broad categories. The first deals with the correlation of the size of expenditures claimed in comparison to the technological objectives and scope of the work undertaken within a given SR&ED project. Specifically is the size of the expenditures claimed consistent with the scope and complexity described by the technical components of the SR&ED project. The second component of the claim is the verification of the actual expenditures themselves. It is the former rather than the latter category which has proven historically to be the most difficult for the program. The establishment of the appropriate size for given claim is most problematic in development environments where projects and project accounting are organized along business projects rather than focused or isolated SR&ED projects.
The following is a list of some potential options of methodologies which could be utilized to provide documentary evidence which reasonably establishes the size and the allocation of costs for a given SR&ED project. Each of the options identified below usually provide sufficient evidence on their own individual merit with respect to capturing the size of the labour expenditures for a given eligible claim.

1) **Resource Allocation Plans** - Utilise the resource planning element of a project schedule plan to identify the resources associated with specific activities, where the activities can be delineated on the basis of ITC eligibility. Use these planning records to frame an estimate of the total time spent on R&ED activities. For example in dedicated R&D environments or emerging start-up companies where the activities are obviously dedicated to R&D it is often sufficient to identify individuals who by virtue of the current corporate environment and job function, are clearly undertaking eligible work contained within an SR&ED eligible project job function in order to determine costs associated with a given project.

2) **Time Sheets** - Individual time sheets which differentiate time spent on SR&ED eligible

from that spent on routine engineering are quite effective. This particular technique however can prove to be quite problematic in that most conventional commercial time tracking systems used by a particular software developer are oriented towards business projects rather than individual SR&ED projects or activities.

3) **Supervisors Summaries** - Periodic (weekly/,monthly) summaries constructed by technical supervisors in "real time" which estimate the amount of SR&ED work undertaken for the given period and identifies both the significant SR&ED effort as well as and any major ineligible activities during the period in question.
Section 2 of this document outlined a collection of potential sources of documentary evidence which may be used to support the various eligibility criteria for a given SR&ED the project. Such documentation can also play a significant role in validating the size of a given claim.

Sufficiency of Supporting SR&ED Evidence for Software Claims

1.0 Background Material:

Objective:
The purpose of this document is to examine the issues surrounding the interpretation of what is sufficient evidence with which to support claims in respect to software development projects for the SR&ED tax credits;

Background:
The primary interpretation policy for the SR&ED program as embodied with the Income Tax Act (ITA), that is currently used by CCRA and accepted by the Courts is information circular 86-4R3. IC 86-4R3 establishes the three criteria for eligible SR&ED, including technical or scientific advancement, technical or scientific uncertainty and technical or scientific content. The content criterion includes the expectation that claimants will have sufficient documentation and evidence of a systematic program of investigation through experimentation or analysis for any developments claimed as an essential element of an SR&ED project. However, little guidance has been provided on specific documentation and evidentiary requirements. Concern has been expressed by claimants, that the documentation required to support an SR&ED claim requires additional resources to create. Additionally it has been unclear what information that is produced in the course of the developmental effort will be recognized as evidence of eligible SR&ED activities.

Intended audience:

This document is directed towards qualified software development practitioners, software engineering managers, CCRA technical reviewers, and tax accountants.

Related documents:
This document must be used in conjunction with the following two companion documents;

1. Guidance to Constructing Effective T661 Software Project Descriptions
2. Guidance to Sources of SR&ED Evidence for Software Claims

In addition the other 1999 issues sheets in this series that have been developed under the auspices of the SR&ED software sector working group also provide valuable insight.

2.0 Sufficiency of Evidence:

It is the premise of this guidance that useful and sufficient evidence pertaining to the eligibility of a particular software project, is normally contained within the naturally arising work product resulting from a given software development project. In order to achieve optimal usage of the evidence for the purposes of supporting an SR&ED claim, such evidence should be archived as it is produced.
The nature of this evidence and how it relates to the three criterion for eligibility are outlined in other issue sheets in this series. Additional information pertaining to maintaining documentation, and identifying potential sources of documentation can be found in the initial guidance provided in this series of issue sheets, i.e. the "*general guidance*" and "*Guidance to sources of SR&ED evidence for software claims*"

This document focuses on what is required to make a reasonable determination of eligibility. Such a determination is the shared responsibility of the claimant and the Agency's reviewers, and the following are the key considerations;

Throughout this issue sheet, it is assumed that the process of documentation by a claimant and its interpretation with respect to software based SR&ED claims will be conducted by

experienced practitioners of the art (e.g. software engineers and computer scientists) who have a current and in-depth understanding of the software technologies being worked with as well as an understanding of the SR&ED incentive program.

As discussed within the "*Guidance to constructing effective T661 Software Project Descriptions*", the documentation and work product (evidence) is not required to be of a tutorial nature. However the claimant ultimately bears the burden to provide reasonable evidence and explanation of eligibility, i.e. that the three criteria are met. As well, the claimant must be able to show that costs of non SR&ED related actives, capitol etc, are appropriately segregated from the claim.

A key complication is that the context in which much of the evidence for eligibility is constructed and stored may be the result of a product oriented development paradigm not an SR&ED focused paradigm. As such, the evidence often resides in documentary forms that do not directly address the specific nature of the eligible components of a claim, (i.e. the three criteria). Nevertheless, the usefulness of specific work product resulting from eligible work (evidence) derived within a product development paradigm, must be apparent to an experienced practitioner of the art, when examined using the guidance and assertions provided by the T661 Project Descriptions.

Ultimately, it is the responsibility of the claimant to demonstrate how records arising during the course of product development can be interpretated as supporting the assertions in the claim, and of CCRA to understand how to effectively interpret the normal range of natural work product commonly produced in the development of software. Such interpretation should normally seek to use the highest level or form of documentation available, which is able to support the assertions of the claimants claim. The claimant should consider, while the work is in process, whether records are being generated which will support this interpretation, and if not should put in place an adequate record-keeping protocol. CCRA will advise on the adequacy of the proposed protocol, in addition the CCRA is responsible for understanding the range of legitimate methodologies which allow claimants to identify and reasonably allocate appropriate costs to eligible SR&ED efforts.

As with all claims the essence of the claimants technical assertions for eligibility will or should be found within the T661 Project Descriptions (users are referred to "Guidance to constructing effective T661 software project descriptions"). Science advisers and claimants alike are commonly faced with the challenge of attempting to correctly ascertain the merits of a given software SR&ED claim, using documentation which has been generated within the context of a larger business oriented development program. The four basic questions that an experienced software practitioner will ask when examining the assertions contained in an SR&ED software claim are as follows;

1. Based on the descriptions and assertions contained within the T661, which documents arising from the development work flow are likely to support the assertions made with respect to technological uncertainty?
2. Based on the descriptions and assertions contained within the T661, which documents arising the development work flow are likely to support the assertions made with respect to technological advancement?
3. Which documents will support the assertions that a systematic or iterative processes of experimentation, testing and or analyses occurred?
4. What is the highest level or most general type of information that will provide evidence?

In recent years it has become a not uncommon occurrence that in the effort to reach a finer granularity with respect to SR&ED evidence, the claim review process began to collect

information at a level of granularity which was too fine to allow recognition of the broader constraints which created the technological uncertainties. **Thus, in each case, science advisers and claimants should measure the merits of the claim from the highest level of or most general level of abstraction possible.**

Irrespective of the methodology used to develop the software, the root issues associated with technological advancement, technological uncertainty must manifest themselves within the work being undertaken and the problems being encountered. Invariably, this leads to an examination of the root issues or challenges which lead to technological uncertainty associated with a given project.

Constraint(s) is/are intrinsically required at a fundamental level in order to form a technological uncertainty. In the complete absence of constraint(s) there is a complete absence of technological uncertainty. The resolution of technological uncertainty created as a result of these constraints, or understanding that they can not be resolved inherently produces technological advancements.

Therefore an eligible SR&ED project will at a minimum always contain evidence of identifiable constraint(s) which provide root cause for the existence of a technological uncertainty or uncertainties. (see section 4.0 of “Guidance to constructing effective T661 software project descriptions”)

Competing constraints of even a generic form which are not technologically founded (i.e. cost etc.) often provide the root conditions which create true technological challenges to the development effort and hence, technological uncertainties. It is important to make the distinction that the presence of such constraints in and of themselves do not constitute a technological uncertainty, but rather the constraints provide boundaries or barriers which create the environment in which a technological uncertainty exists from the perspective of trying to overcome said constraints.

Thus documentation or evidence which addresses or speaks to the central constraints or challenges which the software developer had to address in developing his product will often provide the “markers” needed to identify evidence (either of a direct or indirect nature) for both the technological uncertainty and the advancement.

Thus evidence of constraints which cause a technological uncertainty, coupled with identification of the technological advancements being pursued, in conjunction with evidence that a systematic experimental approach was undertaken to overcome the uncertainty, provides a minimum but sufficient level of evidence for the existence of SR&ED eligibility. The evidence of the existence of such central constraints can and often is contained within high-level documentation.
To justify the size of a claim, experienced practitioners should be able to recognize that the effort required to work on central constraint (s) or uncertainties were commensurate with the need to resolve the constraint(s). Obviously, at a minimum there must be evidence of the nature of the major work which constituted all of or portions of the effort that was needed to resolve the constraints(s) or uncertainties identified.

Recognizing that eligible work will be undertaken within the broader context of a larger business development program, it is reasonable that costing information will reside in various forms of documentation. Such information may or may not resolve to individual activities. Useful and sufficient information for establishing the costs of a project may and does reside in alternate higher level forms than the activities. Such information provides reasoned accounting or reasoned estimates of costs incurred in performing the SR&ED

effort. When such information is used it is often the credibility of the process that is important in establishing sufficiency. The reader is directed to the companion document entitled **"Guidance to sources of SR&ED evidence for software claims"** for examples of possible documentation sources pertaining to project costing.

The test of sufficiency of evidence is met when examination of work product provides sufficient direct or indirect information as to provide a small group of independent, knowledgeable practitioners of the art, (I.E. experienced software developers who are actively working with the technologies in question) with sufficient information to form a reasoned opinion. And that upon forming such an opinion a majority of the practitioners agree that the evidence set before them is sufficient to sustain their expert opinion of eligibility.

Ultimately the onus is upon the claimant to identify and bring forth the evidence to support the claim, and for the CCRA to have the ability to effectively interpret the evidence. By convention and practise the preferred embodiment of such evidence is of a summarized and indirect form.

Part 4. Background Information on this Project

A PROPOSAL FOR A SOFTWARE ACTION PLAN TO ASSIST THOSE CLAIMING SR&ED TAX CREDITS IN THE SECTOR

Prepared by the *Ad Hoc* Software Working Group
for Ron George

November 20, 1998

CONTEXT

Ron George, Vice-Chair of Revenue Canada's "Action Plan Steering Committee" for the renewal of the SR&ED program, has asked that the software industry prepare a proposal for him identifying the key issues impacting on SR&ED claims for software. In addition, he asked that the plan set out processes for addressing the issues and the associated resource requirements.

This proposal for a "Software Action Plan" has been developed through the Canadian Advanced Technology alliance (CATA*Alliance*) in association with the Information Technology Association of Canada (ITAC) and with the assistance of regional associations such as the Information Technology Association of Nova Scotia (ITANS), the B.C. Technology Association of Canada (TIA) and the Vancouver Island Advanced Technology Centre (VIATeC). At this stage, the Canadian Information Processing Society (CIPS) wishes to participate as an observer. CIPS has proposed that its members participate fully in the various activities outlined in this action plan as the work progresses.

The issues outlined in this Software Action Plan represent an expansion of the problems identified in industry's earlier submission to Minister Dhaliwal in May 1998, the Vancouver Conference on the renewal of the program in June 1998, and through subsequent discussions that we have had with company managers, and Revenue Canada officials over the last several months.

We have also reviewed the recently released Action Plan of the Department in developing this paper on what is needed for the software sector. **We would like to note that while we are encouraged by the plan and wish to strongly support it, we do believe that truly substantive movement will occur only when the restructuring of the program gives clear leadership to the renewal. This includes the hiring of a Director General. We would also like to stress that evidence that the department is making progress in dealing with backlogs and achieving the 120 day turnaround will be looked upon on as a major achievement**.
More specifically, from the perspective of the Software sector, we have organized the issues into three separate groups of issues or Actions. Our proposed Actions are as follows:

ACTION 1

Recommendations for the implementation of a series of short term remedial actions by Revenue. They are aimed at encouraging increased greater reasonableness, fairness and consistency in the Department's audit practices. We believe that this action is needed to win back the confidence of the community concerning the Department's commitment to the program. The recommended lead is **Revenue Canada**. Our proposed action will contribute to the actions in Revenue's Plan #s 4, 10 & 11.

ACTION 2

Recommendations for the transformation and revision of some or all aspects Software

Guideline IC 97-1 into a series of clarifications which would be put out as issue sheets or issue specific papers. The intent would be to produce guidance for the software sector that is consistent with the legislation and clearly an extension of Information Circular IC 86-4R3. The issue sheets should be carefully tied back to the primary guideline for the program, Information Circular IC 86-4R3. The recommended prime lead in this case, as recommended at the Vancouver Conference, is **INDUSTRY**. Our proposed action will contribute to the actions in Revenue's Plan #s 3, 6, 7, 8, & 9.

ACTION 3

Recommendations for the development of a separate guide or issue sheet and illustrative examples or models which show how a company can effectively develop the information and records needed to support an SR&ED claim. The objective will be to outline documentation processes that can become an integral part of a company's activities to manage its software development projects. This is in contrast to the stand-alone or add-on approaches being used by most businesses today.. The recommended prime lead is **INDUSTRY**. Our proposed action will contribute to the actions in Revenue's Plan #s 12, as well as 3 & 6.

We feel that it is important for Revenue to have the prime lead on the first action and for industry to have the prime lead on the other two to maximize the acceptance of the products in the community. **In this respect, it is also important to note that the assignment of the prime lead to industry for the development of a consensus on interpretative issues, Action 2, gives greater authority to the product for the courts appear to be looking more favourably on interpretive policy on what's eligible SR&ED when it reflect a true community consensus on where the line should be drawn and is consistent with Information Circular IC 86-4R3 . In all three cases, we emphasize the need for both parties to work closely and be committed to the projects**.

ACTION 1

Comment:

In the review of the program submitted by industry to Minister Dhaliwal in May 1998 and at the Vancouver Conference, a significant number of highly contentious audits were reported. In these audits, the approach being used for verification appeared to be exacerbating the process. In many of these audits of software claims where this problem was reported, the companies did not see major changes in their eligibility once the audit was completed. Some of the problem practices include:

- positional-challenging, non-informative, and show me approaches,
- the refusal to explain what type of information would help support a claim or the basis of a decision,
- what appeared to be the arbitrary raising of requirements for documentation, project detail, project break-out (dis-aggregation), transactional records, and for proof in general over practices that had been acceptable and established in previous audits.

It is understood that, in part, these practices arose because of Revenue's frustration with the poor records being kept by companies in the software sector and the failure of many companies to improve their documentation. **On the other hand, we do not believe that these practices are helping the program to function as an incentive or, for that matter, helping to solve the problem. They are not consistent with the client oriented approaches being promoted by Department officials**. We believe that there are more productive and constructive ways to address the verification of claims. As well, the better use of risk management selection techniques appeared to offer an opportunity to avoid unnecessarily protracted audits and to better focus on establishing accountability rather than revenue recovery .

Detailed Recommendation:

Senior head office staff should issue interim instructions or guidelines for transparent, client oriented

prospective practices commensurate with the administration of an incentive program and the systematic management of the risk. As well, it is recommended that senior head office officials lead regional educational seminars and training sessions which emphasize these techniques, risk management and client service. Where compliance is not compromised, the instructions should be available to the community and training sessions should be opened up to the community. This would let everyone know what is expected. These instructions should be developed after consulting on the problems with accounting managers from the business community, public accounting officials and Revenue's senior field officers. They should be reinforced by the support of senior National and Regional officials.

The interim instructions should cover:

- that the SR&ED program is an incentive program and that audits should reflect a more flexible approach which emphasizes risk management, such as:
  - the use of prior year audits as the base to avoid re-inventing the wheel unless the operations have changed significantly, and
  -better focus on material items;
- that before auditing any details, a short business process review should be undertaken to understand the organization and how it operates to determine the level of detail and focus needed to provide effective reviews targeted on truly material issues;
- approaches to prospective auditing of SR&ED claims which emphasize client assistance, finding out if and why a claim is eligible or has eligible elements, assisting clients to understand what evidence demonstrates eligibility and what records are useful in establishing accountability and the importance of establishing long term agreements on allocation issues and record keeping. Instructions should emphasize that keeping claimants informed of outstanding items throughout the audit is important, as well as the importance of not surprising them at the end;
- the provision of effective, timely, objective and transparent redress (problem solving/conflict resolution), **locally for administrative problems**, and when appropriate, **nationally for policy or interpretative issues**;
- reasonable approaches to project and activity break-out, as well as record keeping;
- the identification, selection and management of consultants so that they provide truly neutral, objective, and reasoned opinions which maximize the acceptance of the consultants' work by the companies;.
- that the list of supporting documentation now identified in IC 97-1 is not to be held up as a hard requirement but is put forward only as an example of what types of information are useful.

Time Frame: within one to two months.
Prime Lead: Revenue Canada.

ACTION 2

Comment:
The software guideline, **IC 97-1**, was seen by those in the software business who helped develop it to be functional and useful **only** if applied reasonably. This caution was explicitly included in **IC 97-1** at the insistence of business before they would sign off on it. **It is now apparent that the guideline must be revisited to assist both Revenue and the software industry to more easily identify the line between what is eligible and what is not.** To do this will require the development of clearer tests and explanation which can help identify the line. The practice of setting out the more rigid position and then explaining that there could be exceptions does not provide sufficient guidance to help **either the auditor or the claimant** understand how to draw a reasonable line nor does the tempering of a relatively restrictive or rigid requirement with a call for its reasonable application. **Specifically, guidance must be developed that is an extension of IC 86-4R3 and the legislation that**:

- better delineates, for both parties, the distinction between eligible projects and activities, and those that are not eligible;
- better delineates the hierarchy of primary SR&ED projects, sub-projects and activities; and how to reasonably aggregate them into a well structured claim which does not artificially force one to create

projects or declare projects finished when they are not. The present definition of a project in IC 97-1 is seen in practice to promote the artificial restructuring of complex SR&ED projects into a series of disjointed projects which can actually hide the reasons for eligibility in the first place;

- better highlights how in business settings, the actual environment in which a technology is to be used and how the product or process will be used, create constraints on the use of known technologies or on the nature of new or improved technologies that in turn create technological uncertainties requiring experimental development. This is severely underestimated in IC 97-1 and, at times, by Revenue's science auditors and consultants;
- better delineates between what is standard practice, debugging, routine fixes and what is eligible - the concept of **"experimental"** as put forward in IC 97-1 as a delineator simply does not work as both are experimentally based in practice;
- better delineates how work on user interfaces and user requirements can be eligible when it is needed to establish user constraints on the technologies that impact on the design of the technologies, and technical specifications. It should be explained that such work is not marketing even when done in a market setting or in conjunction with work on defining the market;
- better delineates how the integration of technologies, both old with old and new with old or new with new, creates major uncertainties requiring substantial advancements, including system uncertainty; and how to separate these situations from the more mundane ones where problems arise from other factors, including project management considerations. This is an area in which Revenue had reported some problems with abuse. The pendulum appears to have shifted since the issuance of IC 97-1 to the point where projects are being challenged where eligibility clearly exists. Some careful thinking on how to separate out the routine is urgently required;
- more clearly spells out that when software products are sold before the software platform is stable or proven out in a technological sense, that commercial sales do not serve as a good indication that a project is complete. It should be the state of the technology that matters - not the sales. A concept of **"technology maturity"** should be explored, i.e., has a technology reached a point where it is stable and become a known entity to experienced users.

Detailed Recommendation:

That the interested members of the working group developing this **Software Action Plan**, augmented by technical mangers from both the software and telecom sectors, and several senior software specialist from the department, form a small new **team** to provide the leadership to clarify IC 97-1 through a series of issue sheets. **These sheets should expand on what is working in IC 97-1 and is consistent with Information Circular IC 86-4R3 and develop new analyses of the methodologies for separating what is eligible in the areas that are causing confusion**.

The following process is suggested:

Step 1- that the team hold **a one day workshop** to cross check the issues, to determine why the problems are arising, to set priorities and to identify potential solutions, and to obtain Revenue's take on the problems - where they are coming from, their constraints, and Finance's take on the policy constraints. Some attention should be given in this workshop to determine if there are particular expenditure issues that should be similarly addressed in a separate process.
The output of the workshop would be a set of priorities for issue sheets, a first cut on potential directions for finding solutions and the establishment of a small leadership team of private sector software developers to work to oversee the development of a draft series of issue sheets.
Step 2- **development of a first-cut at draft issue sheets** through a senior technical manager with broad experience in the industry and with the tax credits (consultant(s) and consensus builder), and a technical writer with the input and oversight of interested developers working with the team.
Step 3- meetings with the full work team especially to review the concepts and issue sheets with Revenue's senior software specialists to obtain their comment, to determine if their constraints have been respected, to see how they see the issue sheets being misinterpreted and to find solutions. **Ultimately, where there are**

fundamentally different positions, it may be necessary for both positions to be brought forward to the Steering Committee or for both positions to be explained in the issue sheet. The objective should first be to find a consensus.
Step 4- finalization of Issue Sheets to the Steering Committee's expectations after focus group testing with technical managers and accountants, as well as Revenue's senior software specialists.
Time Frame: 4 to 6 months.
Prime Lead: Industry
Resource Requirements: Funds for the hiring of authoritative consultant(s)/ technologist(s) and technical writer with demonstrated leadership abilities in consensus building and appropriate support for workshop and focus groups, as well as for the travel of those without adequate resources.
Next Steps: Identification of a consultant(s) and acceptance of the consultant's proposal and plans.

ACTION 3

Comment:
While IC 97-1 does outline the type of information that can support a claim, many users of the program are still having major problems in providing documentation or maintaining documentation. It is also reported that IC 97-1 is being interpreted by some auditors as setting a very inflexible, high standard of documentation which is far in excess of what many companies normally maintain.
The approach outlined in this recommendation would lead to the establishment of a new guide or issue sheet which focuses on showing that the way companies manage their software development creates opportunities to efficiently generate supporting documentation for the key eligibility tests on an ongoing basis, rather than when a claim is filed. It would also focus on identifying the minimum standard of evidence for the various tests, including costs required to support a claim and how this evidence naturally arises in the product development process.

It would emphasize the development of flexible, illustrative example or models of workable approaches to documentation, the type of information that should be maintained, and best practices for a spectrum of business settings common to the sector.

It will be particularly important to emphasize the need to tailor requirements and standards to the needs and capabilities of both large companies and smaller firms to minimize the documentation overhead while still maintaining an appropriate level of accountability. The requirements of companies that contract out SR&ED should be addressed.

Detailed Recommendation:

That the Working Group preparing this ***Software Action Plan*** set up a **small team** of those interested members of our Working Group, of software developers/managers and of senior Revenue software specialists and/or Revenue's consultants to take responsibility for the development of the guide through consultants working on their behalf. Other officials from Revenue should be consulted throughout the processes on documentation standards, and **the consultant should seek a consensus of all parties on what is reasonable**. **Specifically, the consultants would be expected to come up with guidelines, as well as examples and models of *effective, low overhead* approaches to documentation in the various development environments encountered in the sector and a guide to their use** by working with the team.

Specifically, the **examples and models and the guide should be designed to illustrate how a developer can, on an on-going basis, utilize their project management systems and associated documentation to collect supporting information, and what, *if any,* additional information is needed.** It should be detailed enough to show how a developer can establish a simple project tracking system that would actually assist them to develop their products and more effectively develop evidence for or against eligibility.

The guide would be expected to provide a **minimum and consistent standard, but be flexible** about the

documentation that is deemed reasonable to keep as a function of business practice, capability and size. **This material should be focus group tested** for functionality across a full spectrum of businesses and practices common to the sector. The consultant(s) would seek a consensus on what the standard should be and Revenue's senior software specialists should participate in the evaluation. **The private sector team leading the project would seek advice of the Steering Committee, should a consensus not emerge.**
Time Frame: 4 to 6 months.
Prime Lead: Industry.
Resource Requirements: Funds for the hiring of authoritative consultant(s)/technologist(s) and technical writer. Demonstrated leadership abilities in consensus building, and experience with the management of the documentation and claiming process for the tax credits in the software sector are essential. Appropriate support for workshop(s) and focus groups, and travel of participants with limited resources is also required.
Next Steps: Identification of a consultant and acceptance of the consultant's proposal and plans.

Roles of the Software Sectoral Working Group, the Technical Committee and its Stakeholder Group and the SR&ED Steering Committee

The work of CCRA to improve the effectiveness of their administration of the SR&ED Tax Credit program calls for the extensive involvement of the business community.

The SR&ED Steering Committee provides a national perspective and point for this input. The work of the Steering Committee feeds into individual sectors through the Sectoral Working Groups, which have been established on behalf of the Steering Committee. In the software sector, the Sectoral Working Group, which authored the Action Plan for the sector, has set up a Technical Committee to give leadership to this project aimed at clarifying the guidance available on developing claims for software development. More specifically:

The SR&ED Steering Committee provides advice on and oversight of CCRA's efforts to renew the program. Its members represent a broad spectrum of firms that use the SR&ED Tax Credit program. The Software Action Plan was developed in response to a request by the SR&ED Steering Committee last fall.

Software Sectoral Working Group (the Working Group) was set up on behalf of the Steering Committee to provide leadership in the software sector and input on the progress being made on the renewal from a sectoral perspective. As noted , the Working Group developed the Software Action Plan. As well, it has been a strong advocate for the development of new approaches to the SR&ED audits and for the use of consultants that better reflect the unique characteristics of an incentive program. These issues have become a priority for the Steering Committee.

As well, the Working Group called for the clarification of the interpretive guidelines for eligibility and the documentation of software projects in their Action Plan. The provision of support of this project is the response of CCRA to the priority that the Working Group gives to these problems. The Working Group has set up the Technical Committee to conduct this project.

The Technical Committee is a panel of individuals from businesses across Canada who have extensive experience with the technical aspects of the SR&ED Tax Credit program and with the management of software development projects. The Technical Committee will provide the primary direction for the project. The Technical Committee includes representation from with individuals with experience in both large and small businesses.

Not everyone who expressed interest in this project could be included in the Technical Committee. The interest of these individuals in the project has been accommodated by their inclusion in the Workshop on the 16th of October and in a **Stakeholder Group.** The comments, advice and ideas of the members of the **Stakeholder Group** will be formally sought on the various drafts of material as it evolves during the project. This will be achieved through e-mail distribution of material and collection of comment. As well, follow up conference calls and web posting of the product will be used to collect comment as it is developed.

Partnership between CCRA and the Software Community

The following provides a brief background on the partnership between Revenue Canada and the software community and is important to understand the context around this project for the clarification of Guidelines for SR&ED software claims.

Background

The Scientific Research and Experimental Development (SR&ED) program of CCRA has joined with industry and industry associations to build a partnership.
A conference in June 1998 entitled "Building Partnerships" brought together over 70 representatives from industry associations and individual companies to work with government officials to improve the SR&ED program, and to ensure continued industry participation in its administration. The Department responded to these recommendations by way of an action plan; an essential step in this plan was the establishment of an industry steering committee to oversee its implementation, "the SR&ED Steering Committee". Both industry, through the SR&ED Steering Committee, and CCRA took responsibility to ensure that the program is administered within its legislative framework, and that it is also administered in a way that reflects how R&D is performed in industry.
The SR&ED Steering Committee made up of industry representatives ensures cross-industry consistency and the overall fairness of industry-produced SR&ED guidelines. CCRA, in collaboration with the SR&ED Steering Committee, is providing proper governance and monitoring of the overall performance of the SR&ED tax incentive program. Since the Action Plan is based on industry's recommendations and is driven by industry's steering committee, industry support of the initiative is increased. A marked increase in mutual understanding between government and industry has resulted.

Software sector renewal

Industry claims about 1.4 billion dollars annually of support from the SR&ED program and one of the major beneficiaries of these credits are Canadian software developers who account for approximately one third of the total amount. Two key elements of CCRA's renewal efforts for the software industry are the development through the private sector of:

- Additional guidance on what types of software development work are eligible for the credits and what are not (i.e., clarifications); and
- Additional guidance on how to properly prepare project descriptions and what documentation needs to be maintained to support a claim.

This work arises as a result of an Action Plan developed by the Software sector last fall. Specifically, the work responds to "Action Items 2 & 3" of this Action Plan (below).

ACTION 2

Recommendations for the transformation and revision of some or all aspects Software Guideline IC 97-1 into a series of clarifications which would be put out as issue sheets or issue specific papers. The intent would be to produce guidance for the software sector that is consistent with the legislation and clearly an extension of Information Circular IC 86-4R3. The issue sheets should be carefully tied back to the primary guideline for the program. Information Circular IC 86-4R3. The recommended prime lead in this case, as recommended at the Vancouver Conference, is INDUSTRY. Our proposed action will contribute to the actions in CCRA's Plan #s 3, 6, 7, 8, & 9.

ACTION 3

Recommendations for the development of a separate guide or issue sheet and illustrative examples or models which show how a company can effectively develop the information and records needed to support an SR&ED

claim. The objective will be to outline documentation processes that can become an integral part of a company's activities to manage its software development projects. This is in contrast to the stand-alone or add-on approaches being used by most businesses today.. The recommended prime lead is INDUSTRY. Our proposed action will contribute to the actions in CCRA's Plan #s 12, as well as 3 & 6.

Partners' role

The lead for this work is with the business community who have taken on the task of developing a series of issue sheets and proposed clarifications. CCRA is providing support for this effort in the form of funding and advice on the legislative and administrative constraints, which set the boundaries for the clarifications. By administrative constraints we mean the development of "clarifications" that themselves can be interpreted in ways that result in confusion or in the abuse of the system, i.e., unintended results of proposed clarifications. Specifically, CCRA's inputs will be coordinated through the National Technology Sector Specialists for Software in collaboration with the Manager Client Liaison and will include the following:

- Provide advice on legislative boundaries - i.e. what the law allows for and what is clearly out or beyond the documented policy intent of Finance.
- Provide advice on administrative constraints.
- Field experience - i.e. CCRA's experience in terms of examples of eligible and non-eligible projects, well prepared claims, good documentation processes that work, those that don't work, etc…
- Potential problems - eg. imprecisions in definitions and practices
- Implementation process - i.e. what is feasible (or not) in terms of changes to the program, and the time frame it may take

The work is being managed through the "Software Sectoral Working Group" (the Working Group) which has been set up by CCRA's SR&ED Steering Committee. In turn the SWG has set up a Technical Committee to carry out and oversee the project.

Consultants

1. Denis Hall- Tech Team Management
2. Carl Köhn- CEK Associates
3. Keith Martin- Tantalus Systems Corp.
4. Russ Roberts- CATA*Alliance*/RSA

Technical Committee Attendees

SMEs'

1. Bill Currie-W. Currie & Associates (Ottawa)
2. Bob Horwood -Temple Horwood Consultants Inc (Toronto)
3. Chandra Ahooja - Omicron (Pointe Claire, QC)
4. Chris Prendergast-netMedia Technology Inc (Toronto)
5. Don Kane- NMD Manufacturing Limited (Halifax)
6. Edgardo Gonzalez- The TPI Group (Calgary)
7. Eric Germain- Groupe Conseil Sygertech (Montreal)
8. Lance Gutteridge- Thinking Works Inc (Vancouver)

Large Companies

1. Bill Small & Dave Chen- IBM (one voice) (Toronto)
2. Alan Bernardi - Bell (Montreal)
3. Ian Gordon- Mitel (Ottawa)
4. Roger Fortin- Ericsson Canada (Montreal)
5. Tony Yee and Shelley Cameron (one vote)-Pratt & Whitney (Montreal)

Technical Committee Advisors

1. Ken Murray- Deloitte & Touche (Toronto)
2. Paul Gibney- Thorsteinssons (Toronto)

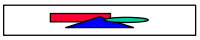
Other

1. Heather Speer-CATAAlliance (Ottawa)
2. Russ Roberts-CATAAlliance (Ottawa)
3. Jean-Claude Gavrel-Revenue Canada (Ottawa)
4. Gerry Goodchild-Revenue Canada (Ottawa)

Stakeholder Group

1. John Leppik- (SME) Don Mills
2. Mark Krebs- Sigscan Systems Corp.(SME) Toronto
3. Neil Law- NCR .(L) Waterloo
4. Allan Hendrickson- Simba Technologies Incorporated (SME) Vancouver
5. Norbert Winklareth- Omnimark (SME) Ottawa
6. Terry Curtis- Ogma Consulting (SME) Victoria
7. Susan Bishop- Ernst & Young (L) Toronto
8. Winnifred Brown- Ernst & Young (L) Toronto
9. Joanne Haushe- Deloitte & Touche (L) Vancouver
10. Rainer von Konigslow- Cookson Walker Chartered Accountants (SME) Toronto
11. Arun Gatha-CGI (SME) Calgary
12. Alan Viau- Xerox Canada Ltd (L) Toronto
13. Ben Dulley- Beta Systems of Canada (L) (Toronto?)
14. Glenn Lott- Price & Lott (SME) Toronto
15. Earl Viner- KPMG (L) (Toronto)
16. Sam Tharani-Arlington Software Corp. (SME)(Quebec)
17. Weston Waldron- -Pratt & Whitney (L) (Montreal)

18. David Sabina- Bateman Mackay Chartered Accountants-(SME) (Burlington)
19. Karen Wensley- Ernst & Young (L) Toronto
20. Fannie Schirmer - CAE Electronics Ltd. (Montreal)
21. Dean Morrison- Price Waterhouse Coopers (L) Toronto
1. Harvey Cantor-Harvey Cantor CA (SME) Toronto
2. Paul Senechal- AIXIA Computer Technics (SME) Toronto



in association with ITAC, CIPS, TIA, ITANS and VIATeC.
(613)236-6550
For comments/feedback hspeer@cata.ca
1999