

What is the Frascati Manual and why is it relevant in determining eligibility for software companies under the R&D Tax Incentive?

The Frascati Manual is widely recognised as a cornerstone of internationally accepted definitions of R&D and classifications of its component activities. The Manual contributes to intergovernmental discussions on “best practices” for science and technology policies, and its pertinence to the Australian policy framework was reaffirmed in last year’s Review of the R&D Tax Incentive (4 April 2016) authored by Mr Bill Ferris AC, Chair, Innovation Australia, Dr Alan Finkel AO, Chief Scientist and Mr John Fraser, Secretary to the Treasury. In the Review, the panel found that:

“...the definition of [eligible R&D] mirrors the principles in the OECD Frascati Manual which is regarded internationally as setting the benchmark for identifying R&D activities.”

Below we set out excerpts taken from the Manual, which may assist R&D software companies better understand when software development qualifies as R&D.

• Routine software development

77. Software-related activities of a routine nature are not considered to be R&D. Such activities include work on system-specific or programme-specific advances which were publicly available prior to the commencement of the work. Technical problems that have been overcome in previous projects on the same operating systems and computer architecture are also excluded. Routine computer maintenance is not included in R&D (see Section 2.4.1 for a more

detailed discussion of borderline problems between software development and R&D).

• Section 2.4.1. Identifying R&D in software development

135. For a software development project to be classified as R&D, its completion must be dependent on a scientific and/or technological advance, and the aim of the project must be the systematic resolution of a scientific and/or technological uncertainty.

136. In addition to the software that is part of an overall R&D project, the R&D associated with software as an end product should also be classified as R&D.

137. The nature of software development is such as to make identifying its R&D component, if any, difficult. Software development is an integral part of many projects which in themselves have no element of R&D. The software development component of such projects, however, may be classified as R&D if it leads to an advance in the area of computer software. Such advances are generally incremental rather than revolutionary. Therefore, an upgrade, addition or change to an existing programme or system may be classified as R&D if it embodies scientific and/or technological advances that result in an increase in the stock of knowledge. Use of software for a new application or purpose, however, does not by itself constitute an advance.



2 BASIC DEFINITIONS AND CONVENTIONS

138. A scientific and/or technological advance in software may be achieved even if a project is not completed, because a failure can increase knowledge of the technology of computer software by showing, for example, that a particular approach will not succeed.

139. Advances in other fields resulting from a software project do not determine whether an advance in computer software has occurred.

140. The following examples illustrate the concept of R&D in software.

Should be included in R&D:

- R&D producing new theorems and algorithms in the field of theoretical computer science.
- Development of information technology at the level of operating systems, programming languages, data management, communications software and software development tools.
- Development of Internet technology.
- Research into methods of designing,

What is the Frascati Manual and why is it relevant in determining eligibility for software companies under the R&D Tax Incentive? *cont.*

developing, deploying or maintaining software.

- Software development that produces advances in generic approaches for capturing, transmitting, storing, retrieving, manipulating or displaying information.
- Experimental development aimed at filling technology knowledge gaps as necessary to develop a software programme or system.
- R&D on software tools or technologies in specialised areas of computing (image processing, geographic data presentation, character recognition, artificial intelligence and other areas).

141. Software-related activities of a routine nature which do not involve scientific and/or technological advances or resolution of technological uncertainties are not to be included in R&D. Examples are:

- Business application software and information system development using known methods and existing software tools.
- Support for existing systems.
- Converting and/or translating computer languages.
- Adding user functionality to application programmes.
- Debugging of systems.
- Adaptation of existing software.
- Preparation of user documentation.

142. In the systems software area, individual projects may not be

considered as R&D but their aggregation into a larger project may qualify for inclusion. For example, changes in file structure and user interfaces in a fourth-generation language processor may be made necessary by the introduction of relational technology. The individual changes may not be considered R&D if viewed in their own right, but the entire modification project may result in the resolution of scientific and/or technological uncertainty and thus be classified as R&D.

• Examples illustrating differences between basic, applied and experimental research

256. Examples from software development:

- Search for alternative methods of computation, such as quantum computation and quantum information theory, is **basic research**.
- Investigation into the application of information processing in new fields or in new ways (e.g. developing a new programming language, new operating systems, programme generators, etc.) and investigation into the application of information processing to develop tools such as geographical information and expert systems are **applied research**.
- Development of new applications software, substantial improvements to operating systems and application programmes, etc., are **experimental development**.



If you are undertaking R&D activities please call (as above) or email:

Gerry Frittmann	gerry@tcf.net.au
David Tonkin	david@tcf.net.au
Dr Raja Mohan	raja@tcf.net.au
Rick Eardley	rick@tcf.net.au
Steven Danaskos	steven@tcf.net.au
Ian Leal	ian@tcf.net.au
Dr Robert Judd	rob@tcf.net.au
Jenny Cammell	jenny@tcf.net.au
Carlo Ramondetta	carlo@tcf.net.au
Dr Terry Freund	terry@tcf.net.au
Michael Valentine (MELB)	michael@tcf.net.au
Lee-Ming Au (MELB)	lee-ming@tcf.net.au