TECHNICAL REPORT

ISO/IEC TR 14759

First edition
1999-12-01

Software engineering — Mock up and prototype — A categorization of software mock up and prototype models and their use

Ingénierie du logiciel — Maquette et prototype — Un classement des maquettes et prototype logiciels et leur utilisation

Reference number
ISO/IEC TR 14759:1999(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide by a simple majority vote of its participating members to publish a Technical Report. A Technical Report is entirely informative in nature and does not have to be reviewed until the data it provides are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this Technical Report may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 14759 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

Introduction

Mock up and prototypes are commonly confused. This stems from the English word "prototype" which covers the two French terms "maquette" (mock up) and "prototype" (prototype). An examination of the meaning given to the two words mock up and prototype in the aeronautical industry reveals that the first stipulates a version of the plane which cannot be piloted and the second a version which can be piloted. Using this analogy, a software mock up is defined as a provisional product that cannot be piloted by users, is not intended to evolve into a fully operational product and may be thrown away once its objective has been achieved. A software "prototype" is defined as something that can be piloted, is developed as a part of the target product and may evolve into an operational product.

Besides this first distinction between mock up and prototype, two additional distinctions have been introduced to address specific features of software issues:

— "illustrative" (related to realistic graphic representation, as in HCIs) versus "functional" (capable of performing computations) address the objective of the mock up and prototype development.

— "demonstrative" (serving to prove the relevance of a solution) versus "operational" (fit for proper functioning in real conditions of operations) address the level of refinement of the deliverable.

Software engineering — Mock up and prototype — A categorization of software mock up and prototype models and their use

1 Scope

This Technical Report applies to any project which uses evolutionary development.

This Technical Report sets out the concepts associated with the production of mock ups and prototypes by identifying and describing the various deliveries which result.

The Technical Report is written for user representatives, developers, managers, quality assurance practitioners of systems and software products and services.

1.1 Purpose

The purpose of this Technical Report is to define the boundaries of mock up and prototype oriented projects.

The purpose of this Technical Report is to reduce risk in critical projects.

1.2 Field of Application

The purpose of this Technical Report does not provide the reader with a normative definition of a specific software life cycle model.

The Technical Report applies to any project which involves uncertain conditions, e.g. safety critical software, user interfaces, new algorithms.

2 References

This Technical Report includes references to other publications. The latest edition of the publication referred to applies.

ISO/IEC 9126:1991, *Information technology — Software product evaluation — Quality characteristics and guidelines for their use.*

ISO/IEC 12207:1995, *Information technology — Software life cycle processes.*

3 Abbreviation and definitions

3.1 Abbreviation

HCI Human-computer interface.

3.2 Definitions

a) Demonstrative

A demonstrative product is a product which proves the relevance of a solution.

b) Functional

A functional product is a product capable of performing computations.

c) Illustrative

An illustrative product is a non functional product.

d) Mock up

A mock up is a throw-away product. Note that it can be retained e.g. for verification, training and as a record.

e) Operational

An operational product is a product which functions in real conditions of operations.

f) Prototype

A prototype is a preliminary type, form, or instance of a system that serves as a model for later stages or for the final, complete version of the system. A prototype is a usable product.

NOTE Refer to clause 4 and Figure 1 for further information about these definitions.

4 Categorization of mock up and prototype models

Mock up and prototype combine the following possible attributes: illustrative, functional, demonstrative and operational, as shown in Figure 1.

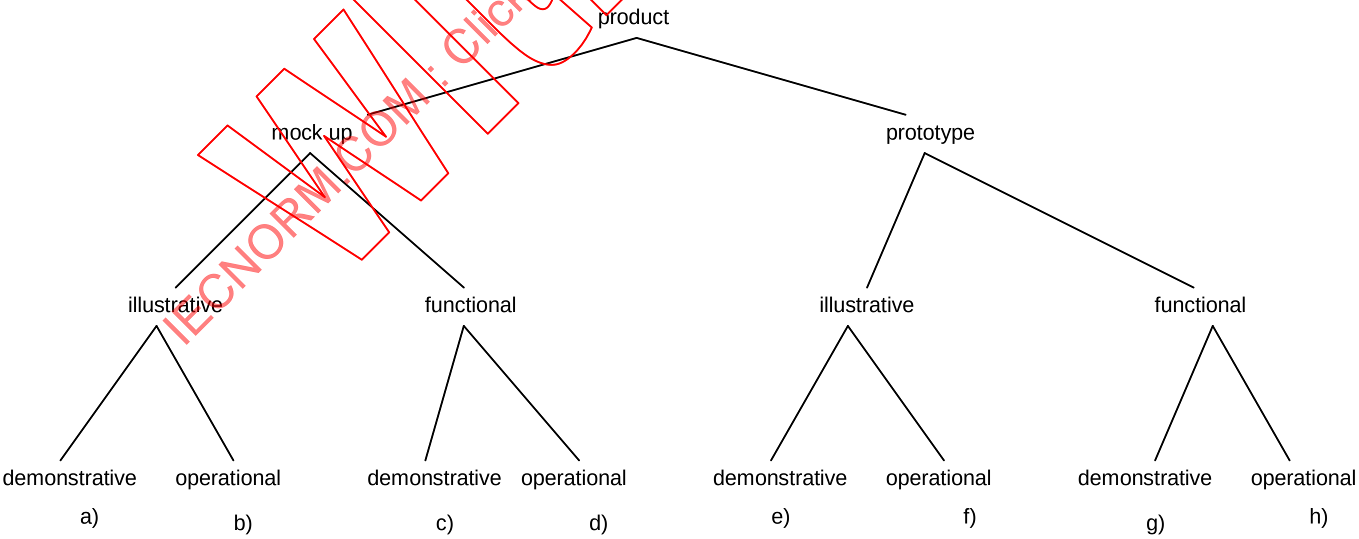

Figure 1 — Three pairs of terms, leading to eight possible types of products

The eight possible types of products identified in Figure 1 are described as follows:

a) A **demonstrative illustrative mock up** is a throw-away product representing the behaviour of the system, running in the development environment.

EXAMPLE A slide-show, usually thrown away, unless it is used for training or marketing the product.

b) An **operational illustrative mock up** is a throw-away product representing the behaviour of the system, running in the operational environment.

EXAMPLE A mock up of a HCI, tested in operational conditions, a situation which is frequently encountered in military contexts. When an operational illustrative mock up is tested, the interface is realistic and usable, but the functional environment is simulated.

c) A **demonstrative functional mock up** is a throw-away product showing the functional organization of the system, running in the development environment.

EXAMPLE The first draft of a real-time algorithm, demonstrating a set of functions.

d) An **operational functional mock up** is a throw-away product showing the functional organization of the system, running in the operational environment.

EXAMPLE The mock up of an algorithm, tested in operational conditions, to validate the reliability of a solution.

e) A **demonstrative illustrative prototype** is a usable product representing the behaviour of the system, running in the development environment.

EXAMPLE A prototype of a HCI, tested in development conditions.

f) An **operational illustrative prototype** is a usable product representing the behaviour of the system, running in the operational environment.

EXAMPLE A prototype of a HCI, tested by target users in operational conditions.

g) A **demonstrative functional prototype** is a usable product showing the functional organization of the system, running in the development environment.

EXAMPLE The prototype of real-time software, still tested in development conditions.

h) An **operational functional prototype** is a usable product showing the functional organization of the system, running in the operational environment.

EXAMPLE The prototype of real-time software, tested by target users in operational conditions.

5 Use of mock up and prototype models

Both mock up and prototype are helpful to analyse user's software quality requirements and to investigate possible risks on software quality characteristics, although they have differences and should be distinguished.

These software characteristics are, for example, interoperability of functionality, fault tolerant of reliability, operability of usability or time behaviour or efficiency and so on, which may be found in ISO/IEC 9126.

5.1 Mock up versus prototype

The mock up does not provide a real version but a realistic version of the target product. The purpose of a mock up is to give users a representative image of the target deliverable, enabling them to detect and correct deficiencies at a time when the cost of such modifications is still quite low. Later, the software supporting the mock up is discarded; the design of the mock up is terminated but its specification may be reused in any further development.

The software supporting the mock up may also be kept along with the design as a quality record and for use in subsequent verification and validation.

The mock up is characterized by the following statements:

a) Not all features of the target product need be represented;

b) The development environment (machine, language, and tools) need not be that of the target product but it must be representative of the target product;

c) The operating environment technical components need not be those of the target product but they must be representative of the target product;

d) It is acceptable to use a subset of the documentation normally used to document the full development life cycle.

The prototype provides a simplified but usable version of the target product. The purpose of a prototype is to assess the interactions between the target operating environment and the target deliverable. Later, the software supporting the prototype should be improved and should remain as a part of the target product.

The prototype is characterized by the following statements:

a) Not all features of the target product need be represented;

b) The development environment (machine, language, and tools) is that of the target product;

c) The operating environment technical components are those of the target product;

d) Document the full development life cycle as if it were the final product.

The following is an example of usage of mock up and prototype throughout the software life cycle.

Table 1 — Mock up or prototype cross-reference to ISO/IEC 12207[1]

	Mock up	**Prototype**
Development Process		
environment	operational or demonstrative	operational or demonstrative
machine	**mock up machine**	**target**
language, tools	**mock up language and tools**	**target**
operating env. tech. components	**mock up components**	**target**
operating env. constraints	operational or demonstrative	operational or demonstrative
functions	illustrative or functional	illustrative or functional
data	operational or demonstrative	operational or demonstrative
Quality assurance Process		
output	operational or demonstrative	operational or demonstrative
reliability	operational or demonstrative	operational or demonstrative
Validation Process		
users	operational or demonstrative	operational or demonstrative
conditions of use	operational or demonstrative	operational or demonstrative
Documentation Process		
phase management doc.	**specification only**	**normal**
technical production doc.	**reduced**	**normal**
manuals	**reduced**	**normal**

5.2 Illustrative versus functional

The aim of an illustrative deliverable is to assess the components of the software which are related to representation. An illustrative deliverable provides a realistic preview of the target behaviour of the system.

The illustrative deliverable is characterized by its ability to illustrate HCIs without providing the functionality of these HCIs. For example, the screens are provided and can be activated, but no output is computed. Tools used for these representations can produce computer output responses to verify user interaction with the system.

The aim of a functional deliverable is to assess relevance of the solution. The functional deliverable is characterized by its ability to implement all functions which are supposed to be within the scope of the prototype.

The following is an example of usage of mock up and prototype throughout the software life cycle from illustrative and functional viewpoints.

[1] The features used in Table 1 are:

target - present in the target environment;
reduced - simplified from the target environment; and
normal - conforming with the standards.

Table 2 — Life cycle processes cross reference to illustrative and functional deliverables[2)]

	Illustrative	**Functional**
Development Process		
environment	demonstrative or operational	demonstrative or operational
machine	mock up or prototype	mock up or prototype
language, tools	mock up or prototype	mock up or prototype
operating env. tech. components	mock up or prototype	mock up or prototype
operating env. constraints	demonstrative or operational	demonstrative or operational
functions	**represented**	**normal**
data	demonstrative or operational	demonstrative or operational
Quality assurance Process		
output	demonstrative or operational	demonstrative or operational
reliability	demonstrative or operational	demonstrative or operational
Validation Process		
users	demonstrative or operational	demonstrative or operational
conditions of use	demonstrative or operational	demonstrative or operational
Documentation Process		
phase management doc.	mock up or prototype	mock up or prototype
technical production doc.	mock up or prototype	mock up or prototype
manuals	mock up or prototype	mock up or prototype

5.3 Demonstrative versus operational

This distinction is sequential in that the early phase of a project may be demonstrative, while the later versions are always operational.

Demonstrative deliverable

a) Reduction (in terms of quantity) of functions implemented, and data processed;

b) Simplification of operating environment constraints;

c) Reduction in validation conditions:

— Predetermined scenarios;

— Carried out on development environment;

— By the developer.

Operational deliverable

a) Full range of functions implemented and data processed;

b) Respect of nominal operating constraints;

c) Validation in operational conditions:

— Open scenarios;

— Carried out on target environment;

— By the target users.

[2)] The features used in Table 2 are:

represented - described by signs or symbols;
normal - conforming with the standards.

Design is cumulative from demonstration to operational use.

The following is an example of usage of mock up and prototype throughout the software life cycle from demonstrative and operational viewpoints.

Table 3 — Life cycle processes cross reference to demonstrative or operational deliverables[3)]

	Demonstrative	Operational
Development Process		
environment	**demonstrative**	**target**
machine	mock up or prototype	mock up or prototype
language, tools	mock up or prototype	mock up or prototype
operating env. tech. components	mock up or prototype	mock up or prototype
operating env. constraints	**demonstrative**	**target**
functions	illustrative or functional	illustrative or functional
data	**demonstrative**	**target**
Quality assurance Process		
output	**demonstrative**	**target**
reliability	**demonstrative**	**target**
Validation Process		
users	**demonstrative**	**target**
conditions of use	**demonstrative**	**target**
Documentation Process		
phase management doc.	mock up or prototype	mock up or prototype
technical production doc.	mock up or prototype	mock up or prototype
manuals	mock up or prototype	mock up or prototype

[3)] The feature used in Table 3 is:

target - present in the target environment.

Annex A

Examples of life cycle models for mock up and prototype production

This Technical Report presents three examples of mock up and prototype usage.

The aims of these cycles are as follows:

a) To constitute a frame of reference:

- Applicable to projects of any size; and
- Able to accept methods and their tools used for the development of the mock up or prototype;

b) To structure the software life cycle;

c) To manage interactivity by formalizing the meeting points between developers and users.

Example A — Evolutionary development

Mock up and prototype in their various illustrative or functional, demonstrative or operational forms, provide evolutionary steps to the life cycle.

In contrast to conventional life cycles which are put into place progressively and in which the contents of the phases are determined by those of the target product, this life cycle is guided by the forms of deliveries. According to the definitions given above, the deliveries are the following:

a) Demonstrative (illustrative / functional) mock up;

b) Operational (illustrative / functional) mock up;

c) Demonstrative (illustrative / functional) prototype;

d) Operational (illustrative / functional) prototype.

From the mock up to the prototype, emphasis shifts from specification to design, as shown in Figure A.1.

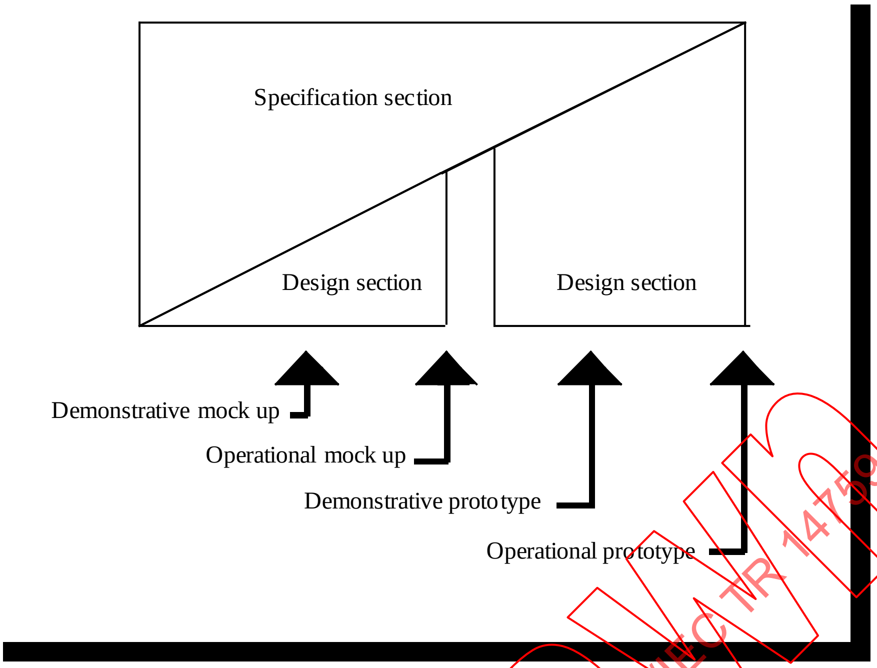

Figure A.1 — Evolutionary specification and design

The specification is cumulative from the mock up to the prototype. It should be seamlessly improved from the mock up to the prototype. With regard to the specification, development from the mock up to the prototype is always evolutionary. At each step of development, the scope of the specification is increasing.

The specification choices, which have a major effect in determining the target cost of the project, are therefore made and validated at the point when they are least expensive.

Design is not cumulative from mock up to prototype. With regard to design, the mock up itself always represents a “throw-away” product. The design effort which was required by the mock up and which was directly linked to the software environment is terminated, and the next step of development should involve a new design effort. The conceptual part of the design, once validated by the mock up, may be accepted or modified in the prototype.

The additional cost with a throw-away design approach is actually the cost of the mock up design phase.

For these reasons, the following guidelines are recommended:

a) Choice of powerful specification components (upper part of Figure A.1) combined with design components chosen for their compatibility with specification mock up and used in an increasingly simplified form;

b) Choice of design components (lower part of Figure A.1) promoting recurrent use of specification components.

Figure A.2 provides several alternative life cycles:

a) The left-side is dedicated to mock up development, the right-side to prototyping;

b) Each of these life cycles may combine illustrative or functional developments;

c) Each of these life cycles put demonstrative and operational developments in sequence;

d) The output of the mock up can be an input for the prototyping.

The original feature of delivery planning is the functional organization of the deliveries. Certain deliveries can be considered as being optional. For instance a demonstrative mock up can jump to a demonstrative prototype, if the product is sufficient to fix the objectives (for example clarify requirements).

A jump from a demonstrative mock up toward an operational prototype would represent a maximum amount of risk. Very short projects may accept short-cuts from demonstrative mock up to operational prototype.

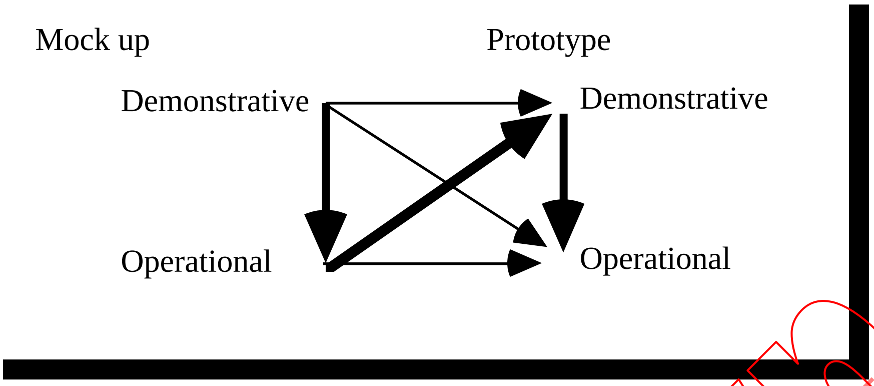

Figure A.2 — Alternative evolutionary life cycles

Example B — Common use of the mock up or prototype to overcome problems

The objective of the mock up or prototype is to fix the problems of definition which are encountered in the main stream of the project. A better understanding of the problem should be the added value of the mock up or prototype approach.

The mock up or prototype as a single product, is incorporated in the software life cycle, to fix some lack of information. As shown in Figure A.3, when the cycle of the mock up or prototype begins, the activity (specification or design) of the main project (grey shade) which are concerned by the mock up or the prototype cannot progress. When the cycle of the mock up or prototype comes to an end, the cycle of the main project resumes at this point.

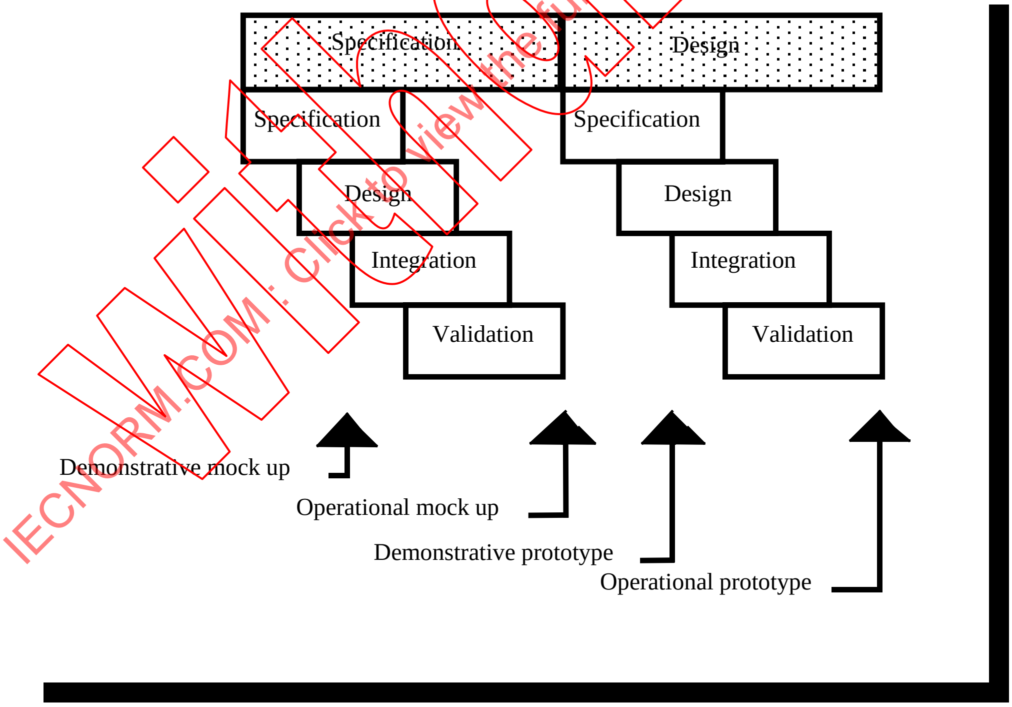

Figure A.3 — Mock up and prototype as single products for the main project (requirements analysis and design phases[4])

[4] To stay within the limits of the guide, these processes will not be addressed exhaustively.