



Standard Guide for Rapid Prototyping of Information Systems¹

This standard is issued under the fixed designation E1340; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ϵ) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 This guide covers a rapid prototyping method for developing information systems that is particularly relevant to systems for the healthcare sector. Intended readers of this guide are people who develop information systems, and students and teachers of system development methods.

1.2 Rapid prototyping is an approach to developing information systems which produce a working model more quickly than conventional approaches. Where conventional methods concentrate on preparing Requirements and design documents that describe the needed system, rapid prototyping methods concentrate on preparing a working prototype. Users and developers learn the functional requirements and an appropriate system design by interacting with a series of prototypes, each of which is rapidly produced from a starting framework or from an earlier version. A prototype can evolve into an operational system, it can serve as an exact behavioral specification of an operational system, or it can be used to explore the feasibility of a new idea or design which can be incorporated in a larger system. The method is rapid in preparing each version of the prototype, but the overall time required for system development may be more or less than the time required with conventional methods.

1.3 Rapid prototyping is most appropriate when the Requirements or design for a system are not well understood, or when experimentation is required to explore some aspect of system behavior. It is not appropriate in hazardous settings, or when the requirements are well understood.

1.4 The guide recommends use of prototyping tools, but it is not a standard for the tools themselves. It does not cover executable specification tools. Transforming a prototype that is used to clarify Requirements into an operational system is discussed briefly in Section 8 and in detail in other referenced standards (see 2.1).

1.5 *This standard does not purport to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appro-*

priate safety and health practices and determine the applicability of regulatory limitations prior to use.

2. Referenced Documents

2.1 ANSI Standards:

ANSI/MIL-STD-1815A Ada Programming Language²

ANSI X3.9 Programming Language FORTRAN²

ANSI X3.159 Programming Language C²

ANSI/X11.1 MUMPS Programming Language²

ANSI/IEEE 610.12 Glossary of Software Engineering Terminology²

ANSI/IEEE 770 X3.97 Pascal Programming Language²

ANSI/IEEE 830 Recommended Practice for Software Requirement Specifications³

ANSI/IEEE 1016 Recommended Practice for Software Design Descriptions³

ANSI/IEEE 1058 Standard for Software Project Management Plans³

ANSI/IEEE 1059 Guide for Software Verification and Validation Plans³

ANSI/IEEE 1063 User Documentation for Computer Software³

ANSI/IEEE 1074 Software Life Cycle Processes³

2.2 ISO Standards:

IS 12207 Information Technology-Software Life Cycle Processes

IS 15288 System Life Cycle Processes

IS 15440 Guide for Life Cycle Processes

IS 11756 MUMPS Programming Language

3. Terminology

3.1 *Definitions*—For definitions of terms relating to information systems, refer to ANDIP⁴ and ANSI/IEEE 610.12.

3.1.1 *fourth generation language, n*—a high-level computer language that incorporates data structures and procedures for a specific problem domain.

¹ This guide is under the jurisdiction of ASTM Committee E31 on Healthcare Informatics and is the direct responsibility of Subcommittee E31.25 on Healthcare Data Management, Security, Confidentiality, and Privacy.

Current edition approved May 1, 2005. Published June 2005. Originally approved in 1990. Last previous edition approved in 1996 as E1340 – 96. DOI: 10.1520/E1340-05.

² Available from American National Standards Institute (ANSI), 25 W. 43rd St., 4th Floor, New York, NY 10036.

³ Available from Institute of Electrical and Electronics Engineers, Inc. (IEEE), 445 Hoes Ln., P.O. Box 1331, Piscataway, NJ 08854-1331.

⁴ American National Dictionary for Information Processing Systems, Information Processing Systems Technical Report X3/TR-1-82, Dow Jones-Irwin, Homewood, IL.

3.1.2 *prototype, n*—an original or model from which a system is copied.

3.1.3 *prototype, v*—to create an original or model.

3.1.4 *prototyping, n*—the activities that create an original or model.

3.1.5 *rapid prototyping, n*—an iterative method for developing prototypes of components, subsystems, or complete computerized systems, in which the time between successive versions of the prototype is short.

3.1.6 *RP, n*—rapid prototyping.

3.1.7 *third generation language, n*—a procedural high-level computer language, such as COBOL, FORTRAN, or Pascal.

4. Significance and Use

4.1 Rapid Prototyping (RP) is a specific Life Cycle Model used to develop an information system which produces a working model of the system very quickly. The RP process shown in Fig. 1 has many similarities, and some differences from the conventional system (Waterfall Life Cycle Model) development process shown in Fig. 2. RP replaces the Requirements and Design processes of the conventional method with an iterative process of prototype refinement. Where the phases of the conventional method produce a set of documents that describe the system, RP produces a prototype. The prototype is tested and refined through several iterations, with intense interaction between system users and developers. RP is an

experimental approach to system development which provides a learning device, the prototype, for users and developers. A prototype can be used as a tool for clarifying Requirements for the operational system, as a means of evaluating a design approach, or as a developing series of versions of the operational system. A prototype is sometimes used as an exact behavioral specification for an operational system which replaces it. Quality characteristics are often sacrificed during RP for the sake of rapid development and low cost; robustness, efficiency, generality, portability, and maintainability are commonly ignored but none of these aspects need to be neglected. However, documentation needed to use the system cannot be ignored but none of these aspects need to be neglected. A “Throwaway” prototype is used specifically to define Requirements which are used to implement a final system. An “Evolutionary” prototype is a prototypical system used for ongoing refinement of Requirements while operational versions at specific milestones are used in production settings.

4.1.1 *Rapid* in RP means that the time between successive versions of the prototype is relatively short. It should be short enough that (1) both users and developers can remember how each version relates to the previous one without written notes, (2) user requirements do not change significantly while a version is being developed, (3) the prototyping team will remain in the project through the RP phase, and (4) total time to develop the system is acceptable. (Expected project duration

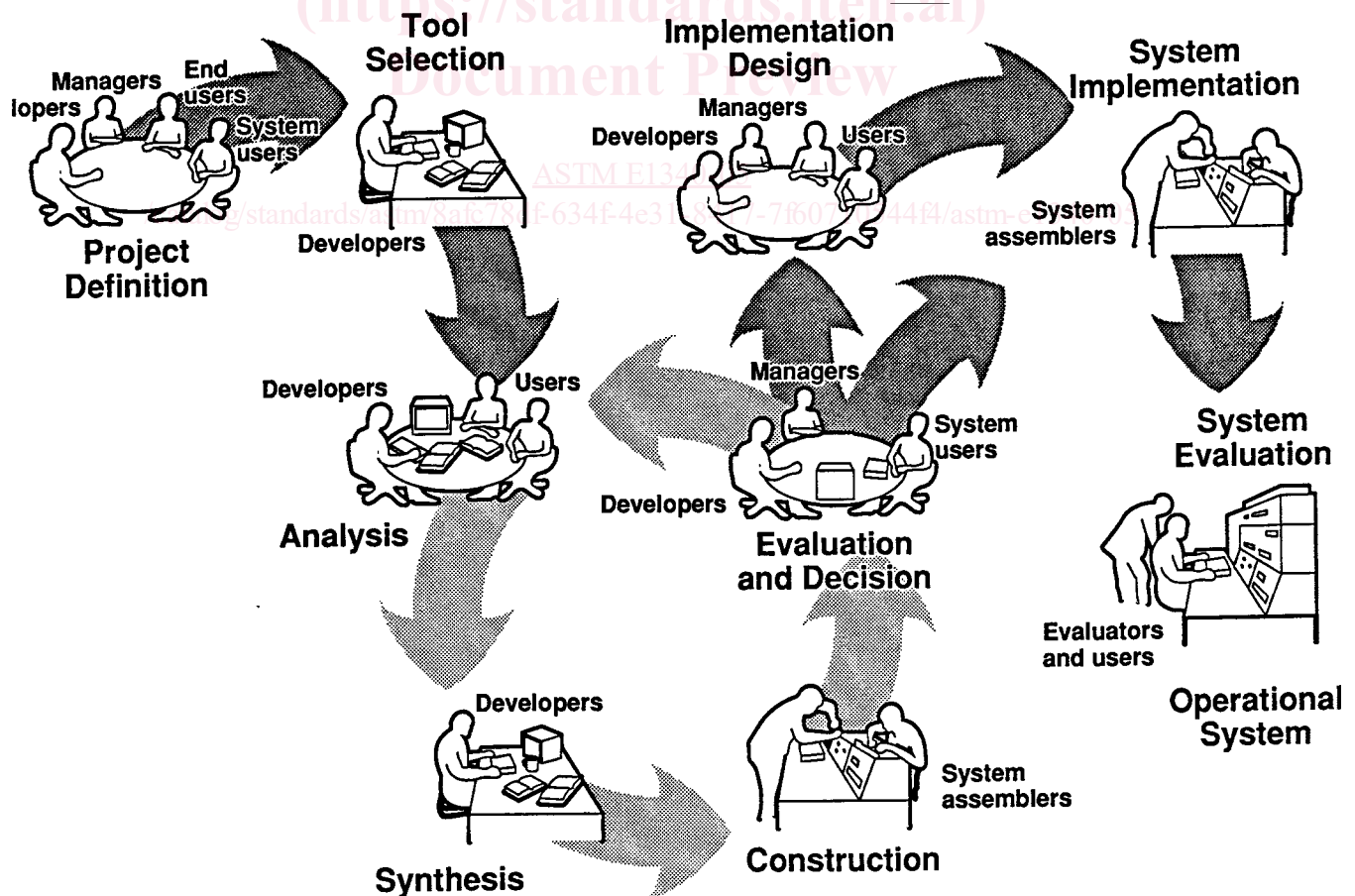


FIG. 1 Rapid Prototyping of An Information System