
Chapter 13

Traditional Rapid Software Development

Objectives

- To explain the roles of prototyping in the software process

Topics covered

- 13.1 Rapid application development
- 13.2 Software prototyping

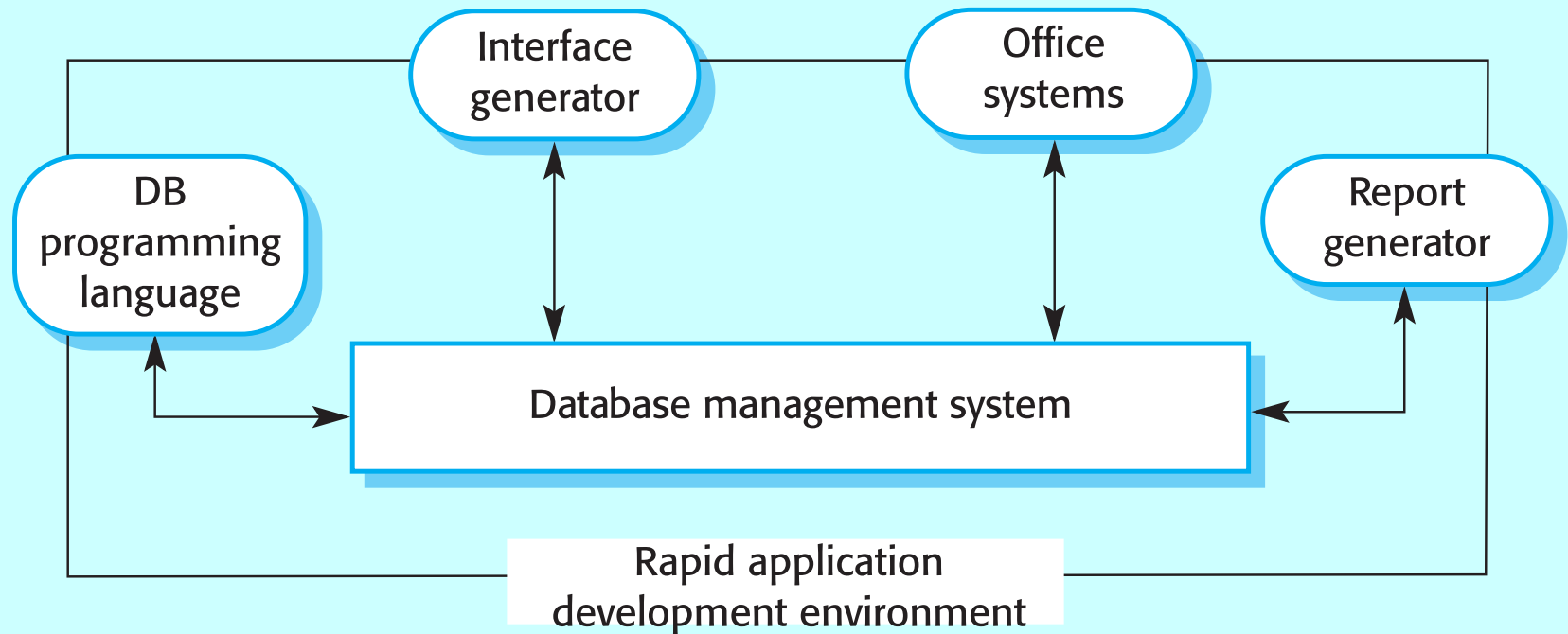
Rapid application development

- Agile methods have received a lot of attention but other approaches to rapid application development have been used for many years.
- These are designed to develop data-intensive business applications and rely on programming and presenting information from a database.

RAD environment tools

- Database programming language
- Interface generator
- Links to office applications
- Report generators

A RAD environment



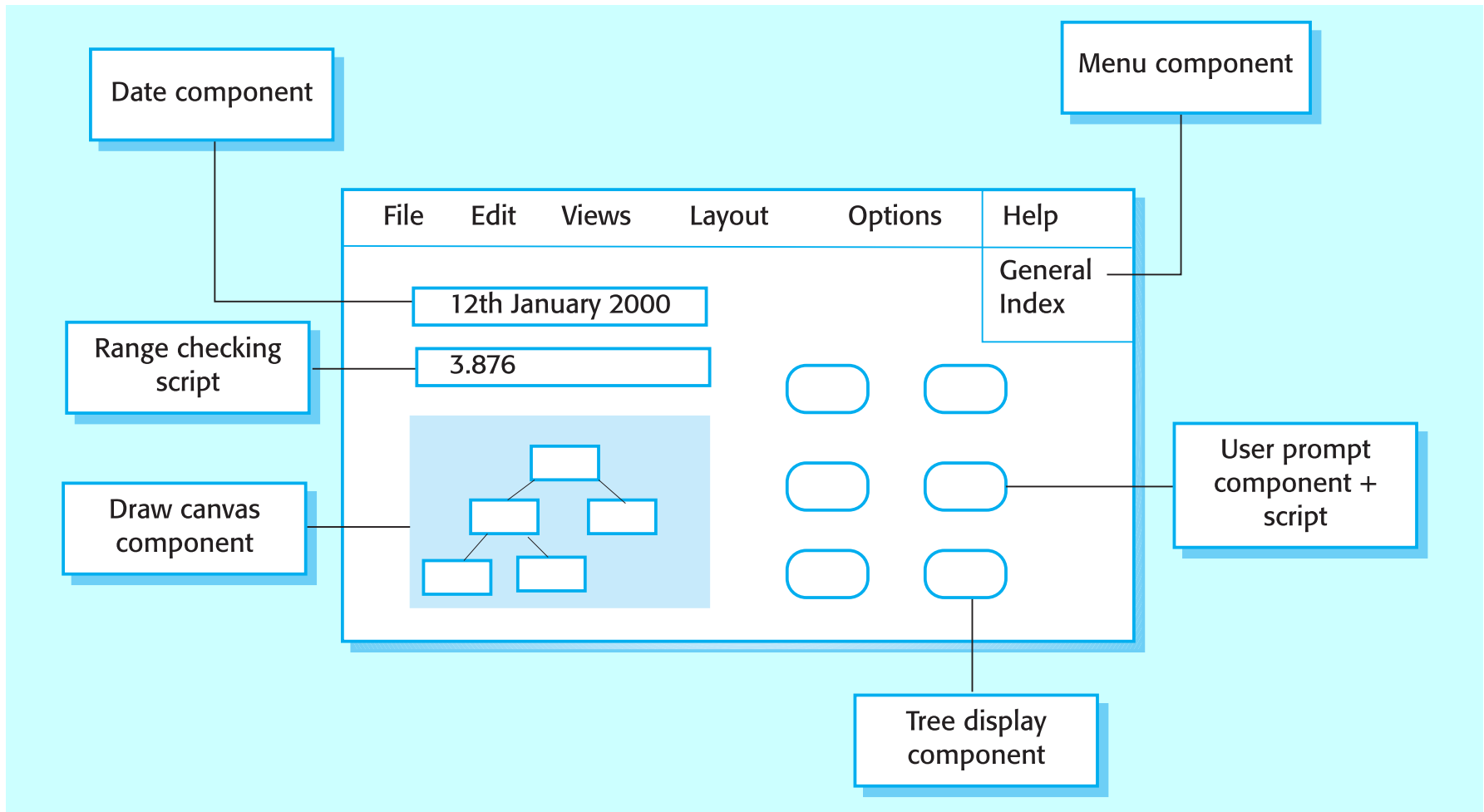
Interface generation

- Many applications are based around complex forms and developing these forms manually is a time-consuming activity.
- RAD environments include support for screen generation including:
 - Interactive form definition using drag and drop techniques;
 - Form linking where the sequence of forms to be presented is specified;
 - Form verification where allowed ranges in form fields is defined.

Visual programming

- Scripting languages such as Visual Basic support visual programming where the prototype is developed by creating a user interface from standard items and associating components with these items
- A large library of components exists to support this type of development
- These may be tailored to suit the specific application requirements

Visual programming with reuse



Problems with visual development

- Difficult to coordinate team-based development.
- No explicit system architecture.
- Complex dependencies between parts of the program can cause maintainability problems.

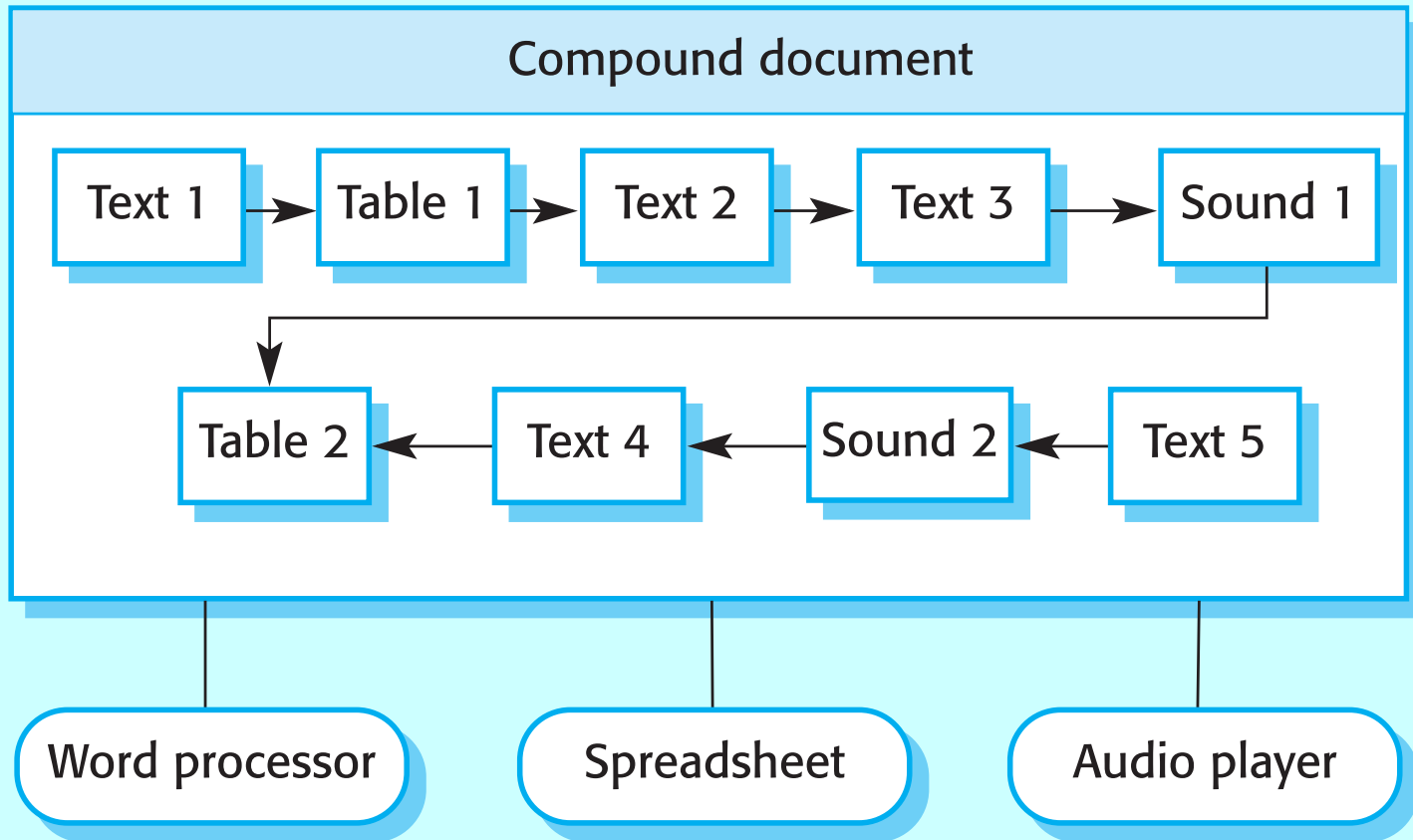
COTS reuse

- An effective approach to rapid development is to configure and link existing off the shelf systems.
- For example, a requirements management system could be built by using:
 - A database to store requirements;
 - A word processor to capture requirements and format reports;
 - A spreadsheet for traceability management;

Compound documents

- For some applications, a prototype can be created by developing a compound document.
- This is a document with active elements (such as a spreadsheet) that allow user computations.
- Each active element has an associated application which is invoked when that element is selected.
- The document itself is the integrator for the different applications.

Application linking



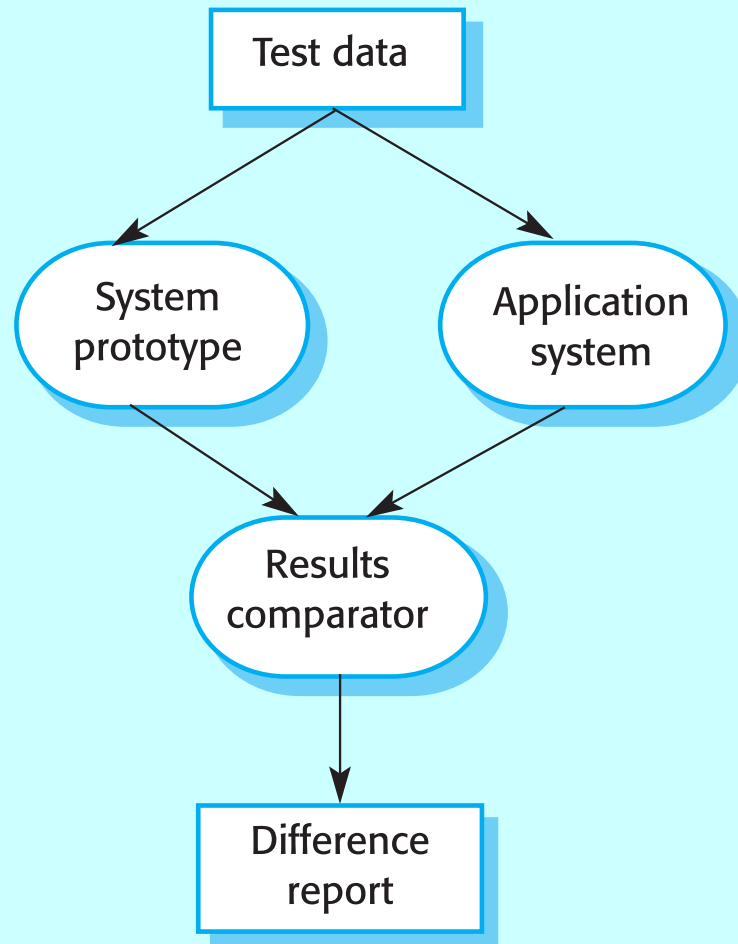
Software prototyping

- A prototype is an initial version of a system used to demonstrate concepts and try out design options.
 - A prototype can be used in:
 - The requirements engineering process to help with requirements elicitation and validation;
 - In design processes to explore options and develop a UI design;
 - In the testing process to run back-to-back tests.
-

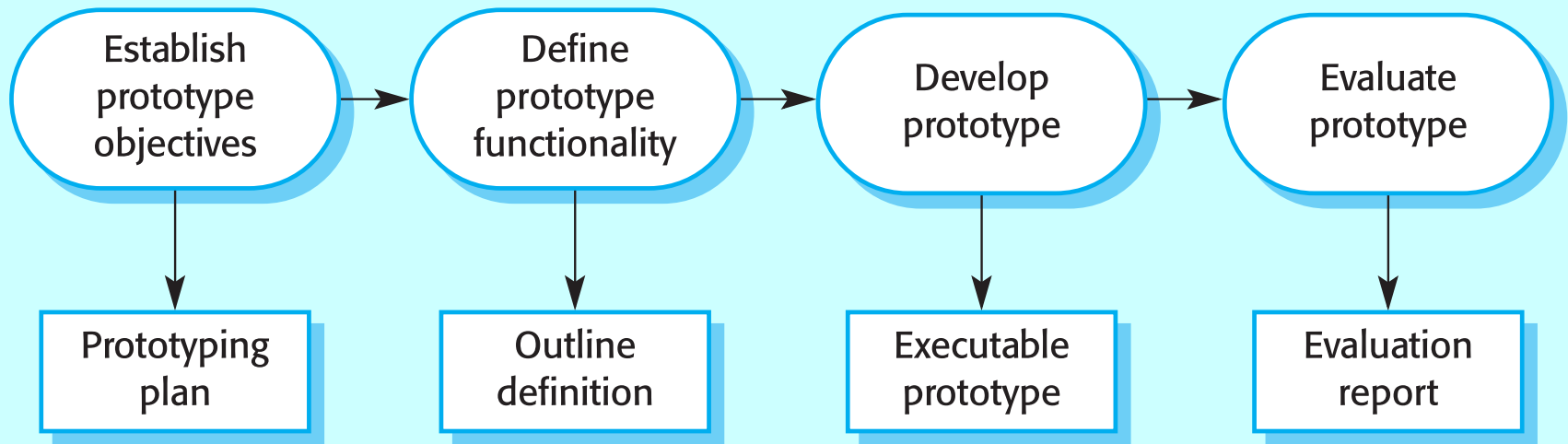
Benefits of prototyping

- Improved system usability.
 - A closer match to users' real needs.
 - Improved design quality.
 - Improved maintainability.
 - Reduced development effort.
-

Back to back testing



The prototyping process



Throw-away prototypes

- Prototypes should be discarded after development as they are not a good basis for a production system:
 - ❑ It may be impossible to tune the system to meet non-functional requirements;
 - ❑ Prototypes are normally undocumented;
 - ❑ The prototype structure is usually degraded through rapid change;
 - ❑ The prototype probably will not meet normal organisational quality standards.