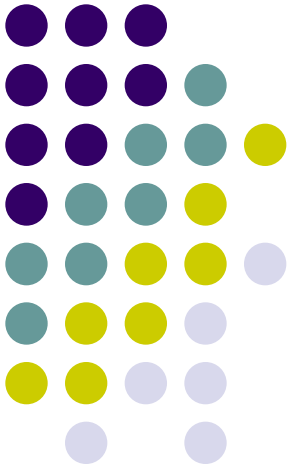
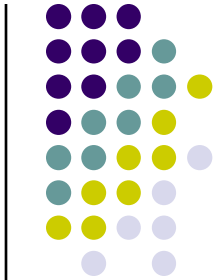
Prototyping

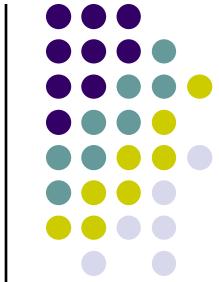


Major Topics



- Prototyping concepts
- Types of prototypes
- Prototyping and the systems development life cycle
- Prototype development guidelines
- Prototype evaluation
- Rapid application development (RAD)

Prototyping

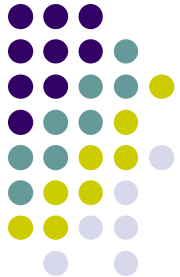


- Prototyping is an information-gathering technique
- Prototypes are useful in seeking user reactions, suggestions, innovations, and revision plans
- Prototyping may be used as an alternative to the systems development life cycle

Initial User Reactions

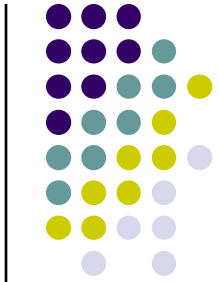
- Reactions must be gathered from users
  - There are three types
    - User suggestions
    - Innovations
    - Revision plans

Four Kinds of Prototypes



- There are four conceptions of prototypes:
  - Patched-up prototype
  - Non-operational scale model
  - First full-scale model
  - Prototype which contain only some of the essential system features

Patched-up Prototype

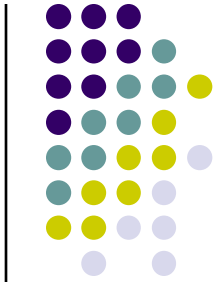


- This is a working model with all the features but is inefficient
- Users can interact with the system
- Storage and retrieval of data may be inefficient
- This type of Prototype Model encourages cooperation of different developers.
- Each developer will work on a specific part of the program. After everyone has done their part, the program will be integrated with each other resulting in a whole new program.
- This type of software development model only needs a strong project manager who can monitor the development of the program.

Non-operational Scale Models

- A non-operational prototype model is used when only a certain part of the program should be updated.
- Although it's not a fully operational program, the specific part of the program will work or could be tested as planned. The main software or prototype is not affected at all as the dummy program is applied with the application.
- This prototype is usually implemented when certain problems in a specific part of the program arises.
- Since the software could be in a prototype mode for a very long time, changing and maintenance of specific parts is very important.

First Full-Scale Models

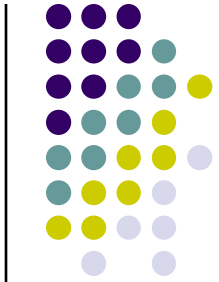


- Create a pilot system
- An operation model
- Useful when many installations of the same information system are planned
- An example is a system to be installed in one location, tested and modified as necessary, and later implemented in other locations.
- Known as a beta version, this Prototype Model could be very efficient if properly launched.

Selected Features Prototype

- An operational model that includes some, but not all, of the final system features
- This is another form of releasing a software in beta version. However, instead of giving the public the full version of the software in beta, only selected features or limited access to some important tools in the program is introduced.
- With the acceptance of these features, later essential features are added
- Some menu items are available
- System is built in modules
- These are part of the actual system

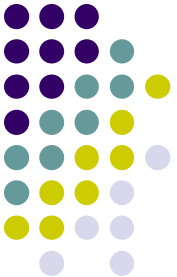
Prototyping As an Alternative to the Systems Life Cycle



- Two main problems with the SDLC
  - Extended time required to go through the development life cycle
  - User requirements change over time

- Prototyping may be used as an alternative

Prototype Development Guidelines



- Guidelines for developing a prototype are
  - Work in manageable modules
  - Build the prototype rapidly
  - Modify the prototype in successive iterations
  - Stress the user interface

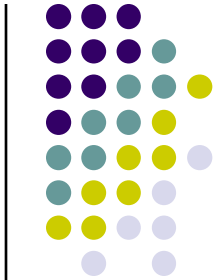
Prototype Advantages

- Potential for changing the system early in its development
- Opportunity to stop development on an unworkable system
- Possibility of developing a system that closely addresses users' needs and expectations
- This way, the actual software could be released in advance.
- The work on prototype models could also be spread to others since there are practically no stages of work in this model. Everyone has to work on the same thing and at the same time, reducing man hours in creating a software.

Prototype Disadvantages

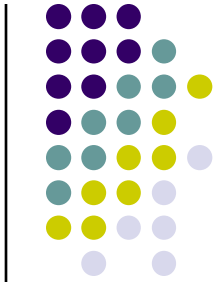
- Managing the prototyping process is difficult because of its rapid, iterative nature
- Since its being built out of concept, most of the models presented in the early stage are not complete. Usually they lack flaws that developers still need to work on them again and again. Requires feedback on the prototype
- Since the prototype changes from time to time, it's a nightmare to create a document for this software. There are many things that are removed, changed and added in a single update of the prototype and documenting each of them has been proven difficult.
- Incomplete prototypes may be regarded as complete systems

Prototype Evaluation



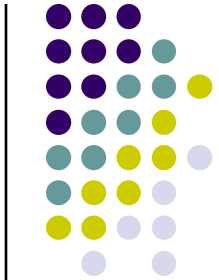
- Systems analysts must work systematically to extract and evaluate users' reactions to the prototype
- Three ways the user is involved
  - Experimenting with the prototype
  - Giving open reactions to the prototype
    - Use a prototype evaluation form
  - Suggesting additions to and/or deletions from the prototype

Prototyping on the Web



- Prototyping on the Web can help to facilitate the prototyping process by
  - Allowing users at a distance review the prototype and send comments
  - Allowing users to review the prototype when they have time, and on any machine that has Internet capabilities
  - The analyst does not have to install the software on the user's computer

Rapid Application Development (RAD)



- RAD, or rapid application development, is an object-oriented approach to systems development that includes a method of development as well as software tools

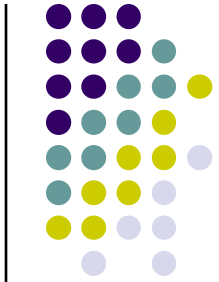
RAD Phases

- There are three broad phases to RAD:
  - Requirements planning
  - RAD design workshop
  - Implementation

Requirements Planning Phase

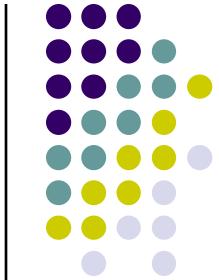
- Users and analysts meet to identify objectives of the application or system
- Oriented toward solving business problems

RAD Design Workshop



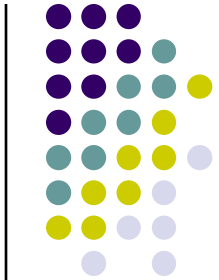
- Design and refine phase
- Use group decision support systems to help users agree on designs
- Programmers and analysts can build and show visual representations of the designs and workflow to users
- Users respond to actual working prototypes
- Analysts refine designed modules based on user responses

Implementation Phase



- As the systems are built and refined, the new systems or partial systems are tested and introduced to the organization
- When creating new systems, there is no need to run old systems in parallel

RAD and the SDLC

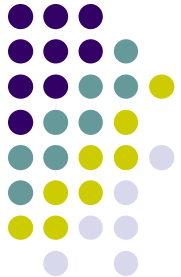


- RAD tools are used to generate screens and exhibit the overall flow of the application
- Users approve the design and sign off on the visual model
- Implementation is less stressful since users helped to design the business aspects of the system

When to Use RAD

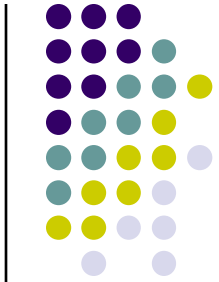
- RAD is used when
  - The team includes programmers and analysts who are experienced with it
  - There are pressing reasons for speeding up application development
  - The project involves a novel ecommerce application and needs quick results
  - Users are sophisticated and highly engaged with the goals of the company

Using RAD Within the SDLC



- RAD is very powerful when used within the SDLC
- It can be used as a tool to update, improve, or innovate selected portions of the system

Disadvantages of RAD



- May try and hurry the project too much
- Loosely documented
- May not address pressing business problems
- Potentially steep learning curve for programmers inexperienced with RAD tools