Peopleware

An example of how several seemingly divergent areas can be merged with multimedia technology.
Multimedia Development
A Team Effort

- Project Manager
- Project Designer
- Writer(s)
- Technical Specialist(s)
- Media Specialist(s)
- Authoring Technicians

Peopleware

- MIS, IS, IT Directors
  - Manage people, projects, purchase/maintain resources
- System Analysts
  - Acquire the requirements then design the systems
- Programmers
  - Code the design of the system
- Technical Writers
  - Document changes, write the manuals, catch inconsistencies, document code, write Scripts

Peopleware

- Testers
  - Create test data, test software and hardware.
- Users
  - Should be involved in All Levels
- Database Managers
  - Maintain data structures, data stores, backups, reports
- Data Warehouse Managers
  - Create statistics, clean, group, categorize, filter data

Peopleware

- Quality Assessment Managers
- Installation Managers
- Service/Help Desk Managers
- Communication Specialists
  - Connect nodes (bounded/unbounded), modems
- Network Administrators
  - Set up IP addresses, user access, maintain system

The Systems Development Life Cycle

Investigation → Analysis → Design → Development → Implementation → Maintenance → Retirement

The graphical "waterfall" model of the SDLC shows a basic sequential flow from identifying the "right things to do" to making sure that "things are done right."
From Idea to Algorithm: Control Structures

- Three basic control structures:
  - **Sequence** - a group of instructions followed in order from first to last
  - **Selection** - to choose between alternative courses of action depending on certain conditions.
  - **Repetition** - allows a group of steps to be repeated several times, usually until some condition is satisfied

Programming a robot Karel

Karel’s World

Primitive Commands

- **TURNON**: Karel must be turned on at beginning of each task.
- **TURNOFF**: Karel must be turned off after last instruction.
- **MOVE**: Karel moves forward one block. He turns himself off if he sees a wall and is told to MOVE. This is an error shutoff.
- **TURNLEFT**: Karel pivots 90 degrees to left.
- **PICKBEEPER**: Karel picks up only one beeper and puts it in his sound proof bag. If no beeper exists at a corner, then an error shutoff occurs.
- **PUTBEEPER**: Karel takes only one beeper out of his bag and puts on the current corner. If there are no beepers in his bag, then an error shutoff occurs.

Task: Pick up all the beepers

There are many routes that could solve the problem

Task: Pick up all the beepers

Which is the most efficient?

Task: Pick up all the beepers

TurnOn
Task: Pick up all the beepers

- TurnOn

Task: Jump walls to beeper

- TurnOn
- Must make decisions
- Test for a wall (True or False)
  - FRONT_IS_CLEAR
  - LEFT_IS_CLEAR
  - RIGHT_IS_CLEAR
- Selection
  - If FRONT_IS_CLEAR Then MOVE Else TURNLEFT
- Repetition
  - While FRONT_IS_CLEAR Do MOVE

Task: Jump walls to beeper

- TurnOn

What if the height of the walls was not known to Karl

Going up the wall use
While right_is_blocked
  Move

Would you use the same code going down the other side?
While front_is_Clear
  Move