Smalltalk Language Concepts

- Everything is an Object
- Every object is an instance of some Class
- Processing occurs by sending messages to objects
- A Class defines the messages to which its instances respond
- Messages common to similar classes are defined in superclasses

Bytecode Interpreter

- The Smalltalk "CPU"
- Executes the Smalltalk instruction set
  Push Pop Branch Send-a-message
- Uses the Object Memory
- Implements "primitive methods"
  - Integer +-*/
  - I/O
  - Array access

Xerox Provides

Describes the language
Specification for the
"Smalltalk-80 Virtual Machine"
Design of an interpreter
which implements the V.M.

Virtual Image

A memory image for the
Smalltalk-80 Virtual Machine.
≈500K Bytes of compiled
Smalltalk-80 code

WHY IS XEROX RELEASING SMALLTALK-80?

- LOTS OF CURIOSITY OUTSIDE OF XEROX
- EGO GRATIFICATION
- CHANGE USER INTERFACES
- CHANGE COMPUTER SCIENCE
- INCREASE THE NUMBER OF SMALLTALK PROGRAMMERS

D.A.D. Implementation Goals

- Demonstrate that Smalltalk can be implemented at Tek
- Learn how Smalltalk works
- Minimize initial implementation time

D.A.D. Implementation Strategy

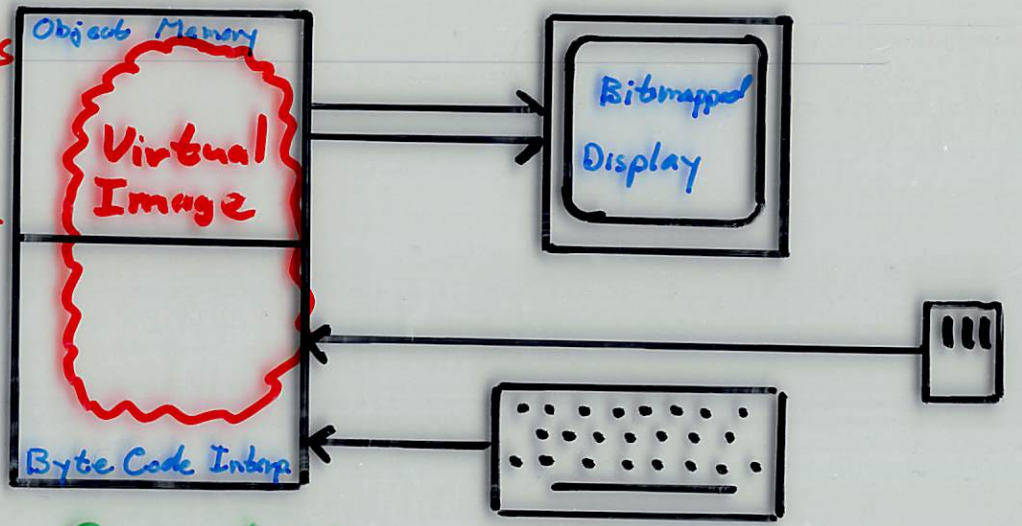
Use a standard microprocessor
- Motorola 68000
- 68000 Board Buckets

Implement interpreter using a high-level language
- GCS 68000 Pascal

Build minimum hardware
- 512×512 Bitmap Display
- Floppy disk, mouse interface

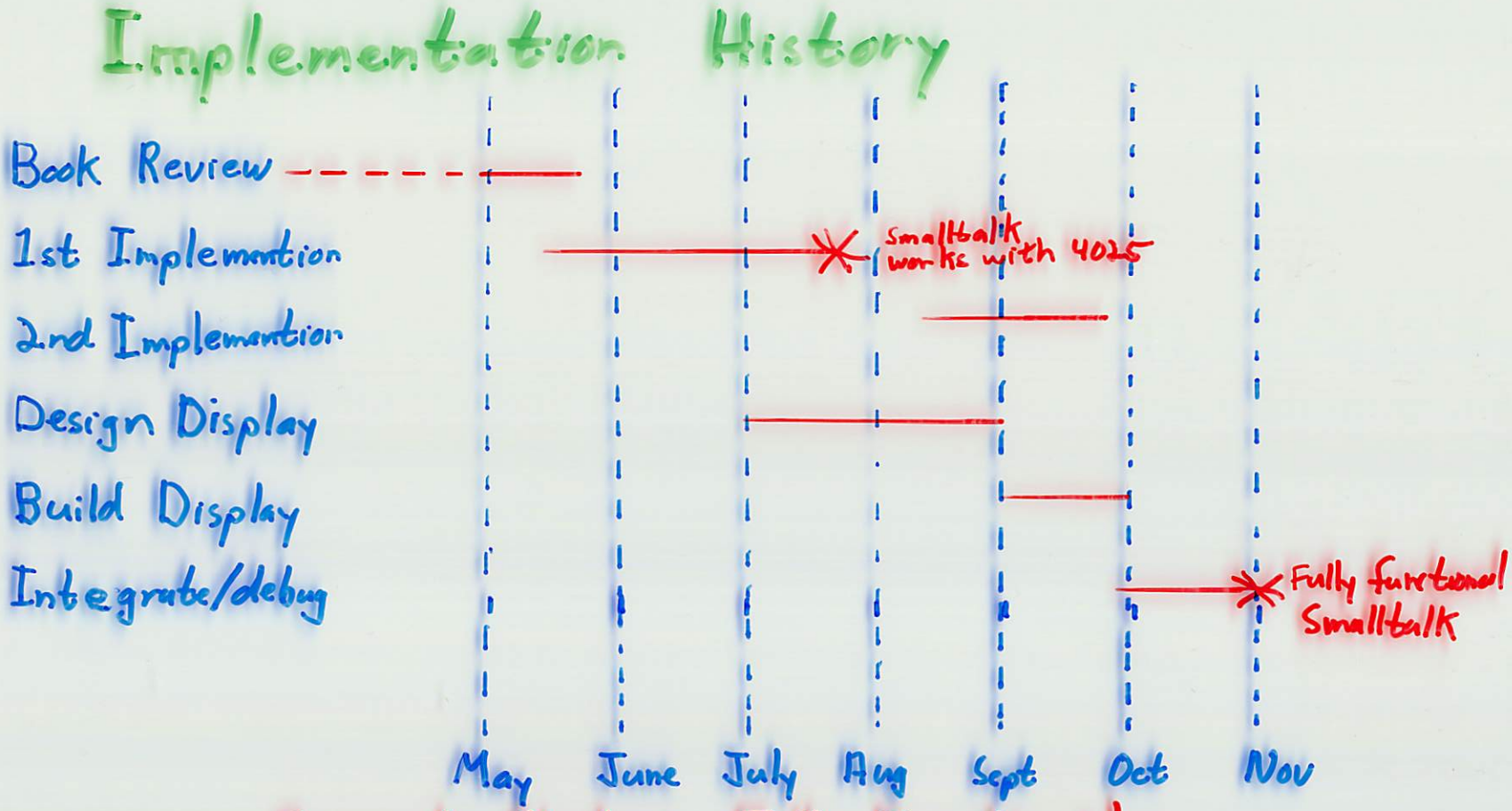
Implementation Leverage

Xerox Provides
≈ 500K
≈ 80 Person-years



Tek Provides:
32k-96k (3000-4000 lines)
3-6 Person-months

Implementation History



Current Status: Fully Functional!
Slow (1000 inst/sec.)

Future Implementation Issues

- Virtual Memory
- Other Display Technologies
- Bit-slice Implementation
- Special purpose hardware support for Smalltalk