



# Software Development in the Digital Library Program

Digital Library Brown Bag

Tamara Cameron

David Jiao

Oct. 22, 2004



# Outline

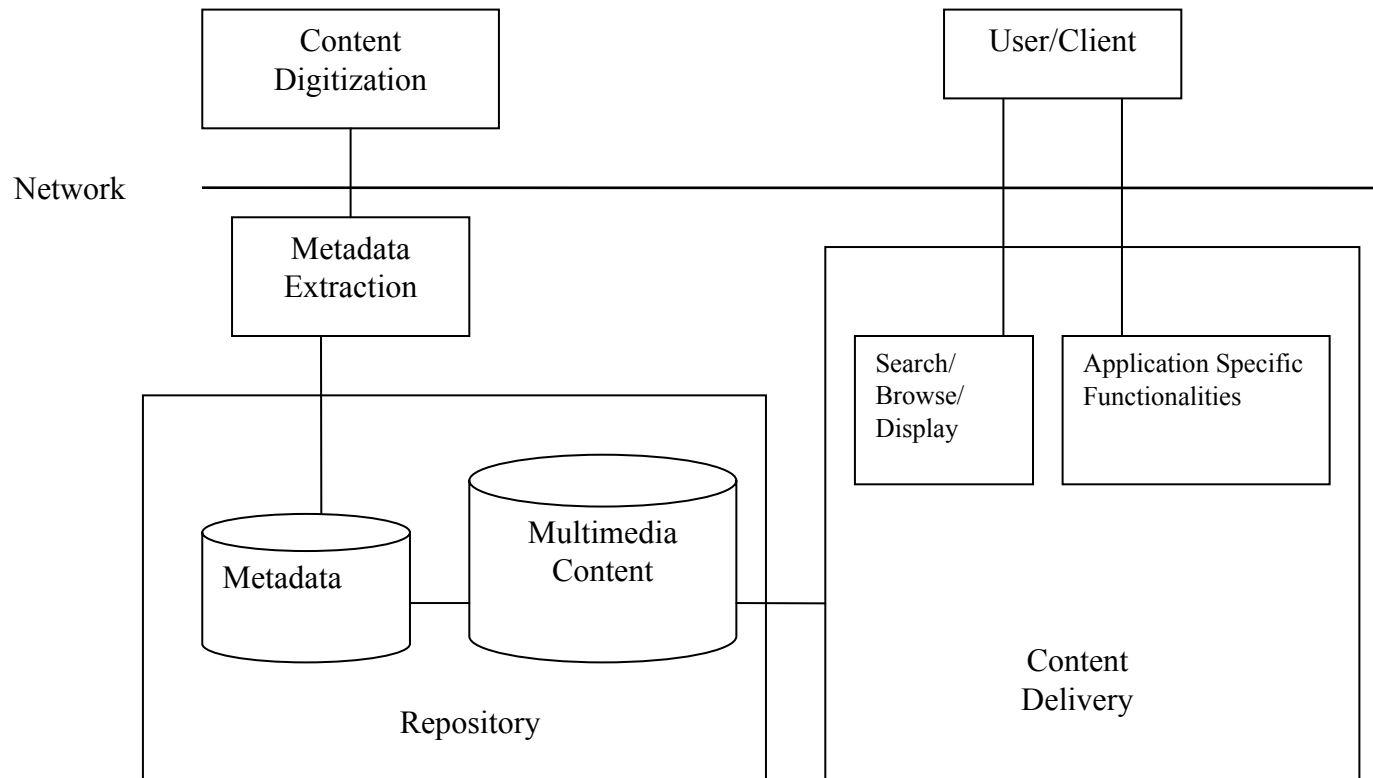
- Custom Development in the DLP
- Overview of Digital Library Program
- Software Development Life Cycle Model
- Case Study – Cushman Project
- Observations
- Recommendations



# Reasons for Custom Development

- Field is relatively young. Early projects were experiments to determine the role, purpose and scope of digital library initiatives within parent organizations.
- Limitations of commercial software in managing large collections of heterogeneous electronic resources

# Digital Library Architecture





# Staffing of Projects

Technical staff are assigned to projects in one of three ways:

- Single, Large Projects (Variations2, EVIADA)
- Single Small Projects (FLI, Newton)
- Multiple Projects (Cushman, Letopis, DLXS)



# Future Directions

DLP are aware and actively working to improve the infrastructure

- Efforts

- Application Architecture: Java/Struts/Tomcat

- Source Code Management: CVS, ANT

- Testing and Maintenance: Bugzilla

- Analysis of the Development Life Cycles of different projects

# Software Development Life Cycle Model (SDLCM)

- SD Activities:
  - Coding/Debugging, Testing, Requirements, Maintenance, Deployment/Installation ...
- A SDLCM is a framework, selected by an organization, on which to map these activities.



# Benefits of Using SDLCM

- Increasing quality
  - Achieve conformance to requirements
  - reduce the number of software defects
  - mitigate risks associated with the software
- Reducing project cost and schedule
  - provides a framework for systematic, incremental software process improvements
  - reduces the cost and schedule of the testing, installation, and maintenance phases
- Improving manageability
  - enhanced accuracy of project planning
  - detailed means of tracking projects
  - early measures of software quality
  - improved repeatability of success stories





# Common SDLCM

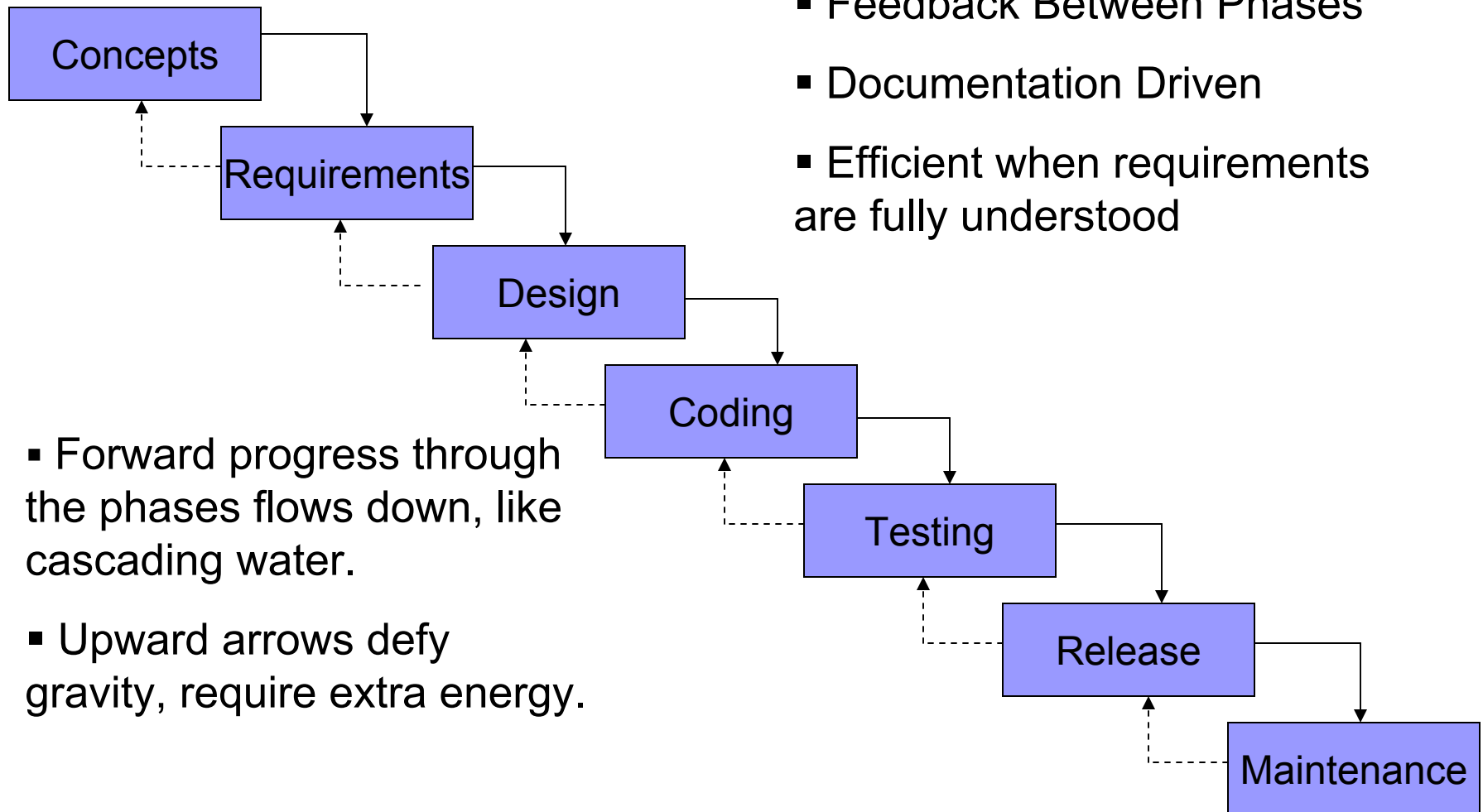
- BuildAndFix
- Waterfall
- Modified Waterfall
  - Prototyping Waterfall
  - Milestone Waterfall
- Spiral



# BuildAndFix

- Vague planning and analysis, if any
- The working program is the only work product
- Appropriate for small programs written by one person
- Understandability and maintainability decrease rapidly with increasing program size

# Waterfall



- Sequential Steps (Phases)
- Feedback Between Phases
- Documentation Driven
- Efficient when requirements are fully understood

- Forward progress through the phases flows down, like cascading water.
- Upward arrows defy gravity, require extra energy.



# Waterfall (cont.)

## ■ Advantages

- Clearly defined phases, good documentation
- Easier Maintenance

## ■ Disadvantages

- Lack of customer involvement
- Sequential and complete execution of phases often not desirable
- Product available very late in process (high risk of wrong product)



# Modified Waterfall

- Prototyping Waterfall

- Build an example system to help elicit requirements
- Perfection of Prototype can take too much time

- Milestone Waterfall

- Milestone builds, dedicated to addition of key functionality
- Excellent for proving concepts when attempting a totally new technology
- Reduce risks by incorporate highest risk functionality into the earliest builds



# Spiral

- Combination of prototyping with iterative development
- Seek feedback from customer
- Appropriate for big projects
- Not appropriate for database driven projects

# Case Study – Cushman Project

- Funded with an Institute of Museum & Library Services (IMLS) grant
- ~ 14,500 color slides, 1938-1969
- Site launched 10/2003 and 03/2004
- Functionalities
  - Search, Browse and Display
  - Controlled Vocabulary, Thesaurus
- <http://webapp1.dlib.indiana.edu/cushman/>

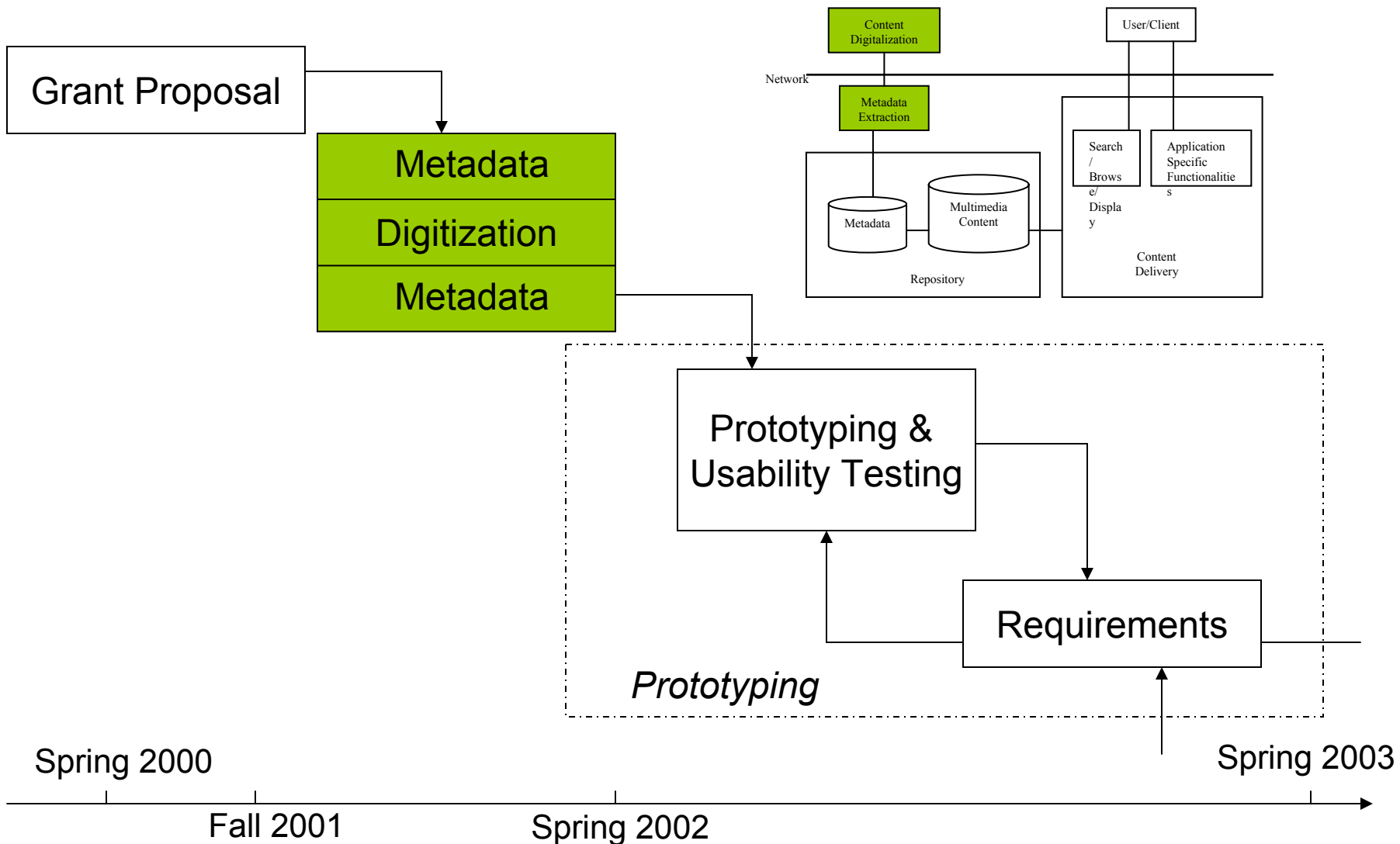


# Success of Cushman Project

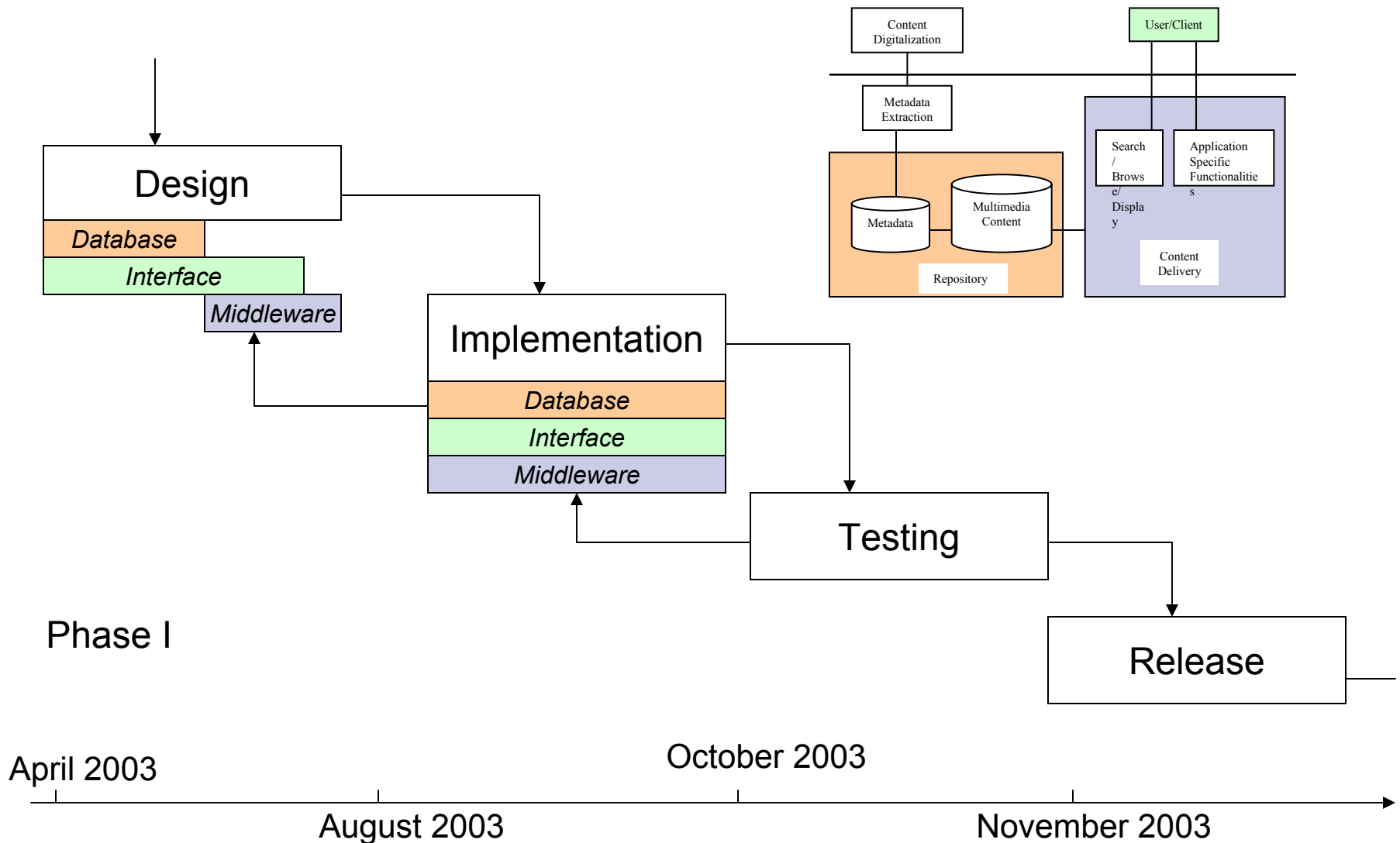
- Completeness of functionality
- High quality, few bugs
- Close to schedule
- Possible reusable code
- Experiences earned are precious



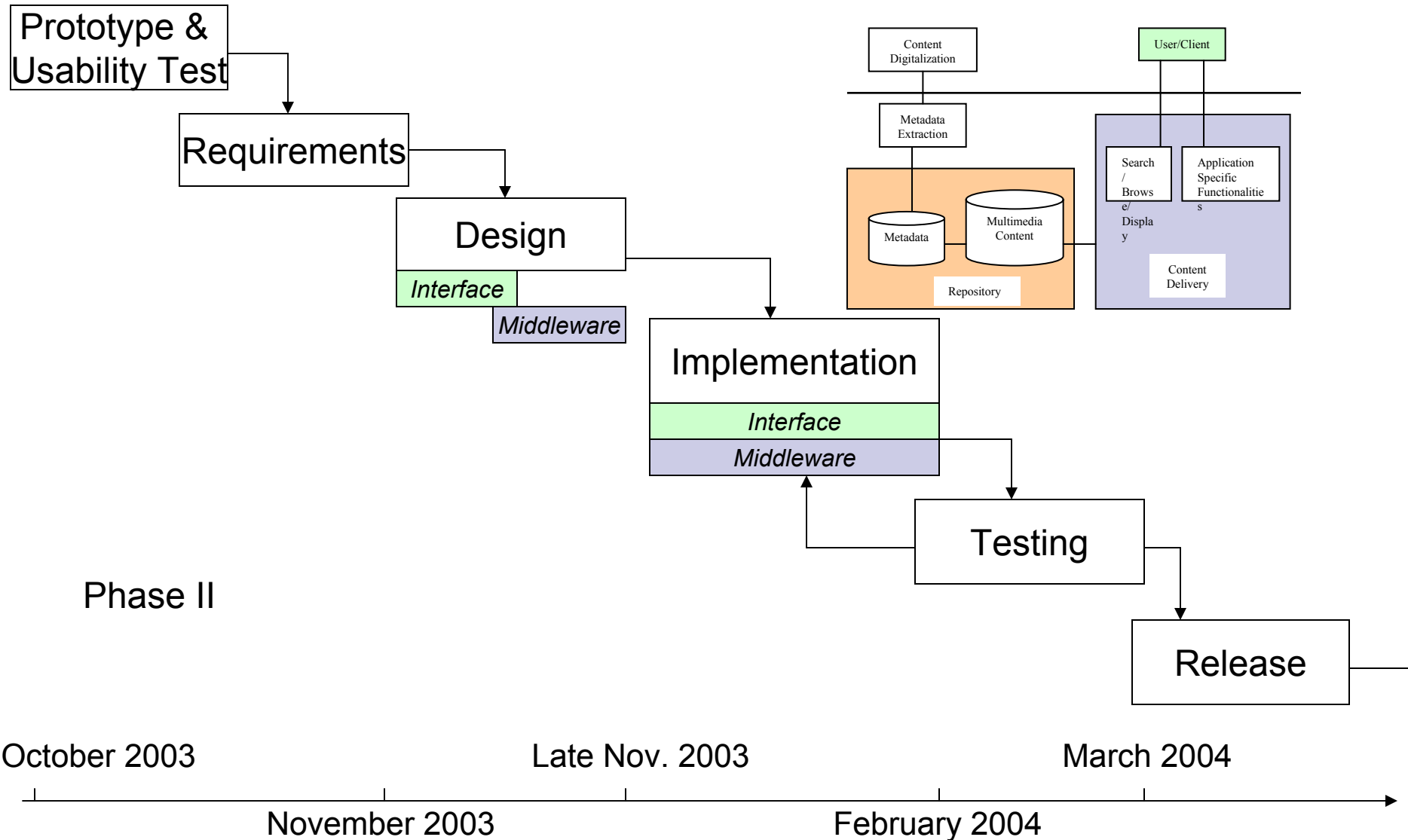
# Cushman Development Life Cycle



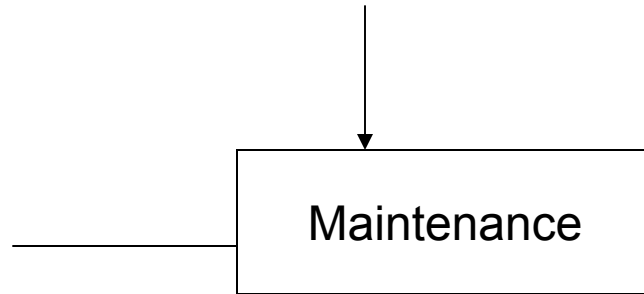
# Cushman Development Life Cycle



# Cushman Development Life Cycle



# Cushman Development Life Cycle





# Summarization of Cushman Development Life Cycle

- Factors contributing to Success
  - Effective Resource Management
  - Commitment to Documentation
  - Clear phases of Development
  - Staged Release
  - Choice of Technology
- Lessons Learned
  - Lack of documentation templates
  - Implementation Documentation
  - Time management



# Observations

- Time and Resource Constraints
  - Not unique to DLP
- Models can be identified but are not planned
  - Growing Pains
- Pros and Cons for programmers
  - Pros: Organic Process, Creative, Lots of freedom
  - Cons: Organic Process, Creative, Too much freedom



# Recommendations

- Continue to work to formalize process
  - Prototyping-Milestone-Waterfall recommended
- Look to Cushman for ideas
  - developing project plans
  - provide structure for current projects
  - identify areas of common functionalities



# Q & A



# References

- Milstead, J., & Feldman, S. (1999). Metadata: cataloging by any other name .... Online (January 1999). Available at: <http://www.onlinemag.net/OL1999/milstead1.html>
- Pasquinelli, A. (2002). The SUN Whitepaper on Digital Library Trends. Retrieved October, 2004 from: [http://www.sun.com/products-n-solutions/edu/whitepapers/pdf/digital\\_library\\_trends.pdf](http://www.sun.com/products-n-solutions/edu/whitepapers/pdf/digital_library_trends.pdf)
- Schmidt, M.E.C. (2000). Implementing the IEEE Software Engineering Standards  
Sams © 2000
- Software Engineering (1997). Dorfman, M. & Thayer, R. (Eds.). IEEE Computer Society Press @ 1997