Developing Web Interfaces for Scientific Data Archives

Tom Stein, Keith Bennett, Dan Scholes

NASA Planetary Data System
Washington University in St. Louis

Dec 2, 2009

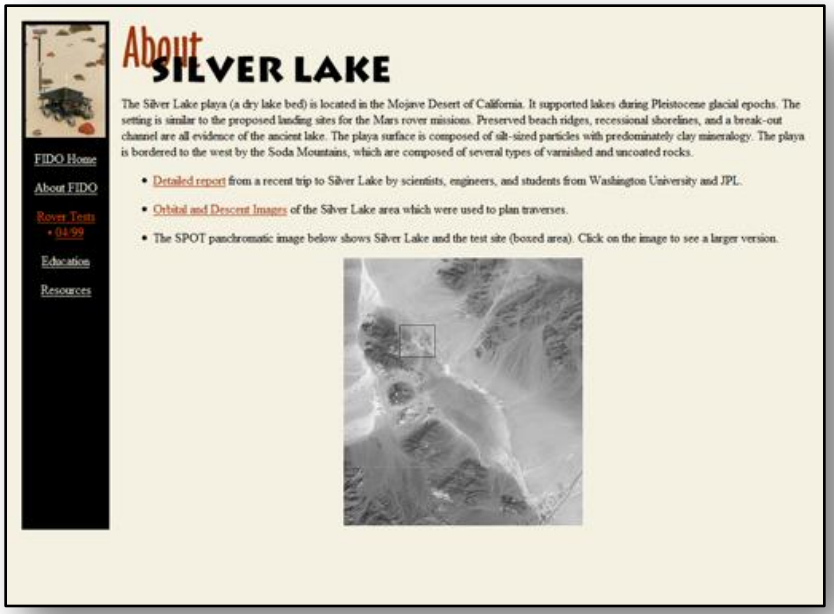
Supported under NASA Grant NNG05GB73G, "Planetary Data System Geosciences Node"

PDS Geosciences Node

- Formed in 1988 to produce, maintain, and distribute archives of geosciences-related mission data for Mars, Venus, Mercury, and planetary moons
- Data sets distributed on optical media until late 1990s
  - Exponential data volume growth
  - Increase in data updates via recalibrations
  - Improved technology supporting electronic search and distribution
  - Change in nature of archives
    - Need for correlating data as missions studied same targets with different instruments
    - Landed missions produced metadata that detailed decision-making process behind data collection

Early interfaces

- Simple design
  - Traverse archive volume directory structures
  - Download individual files
  - View archive documentation
  - Static data product pages



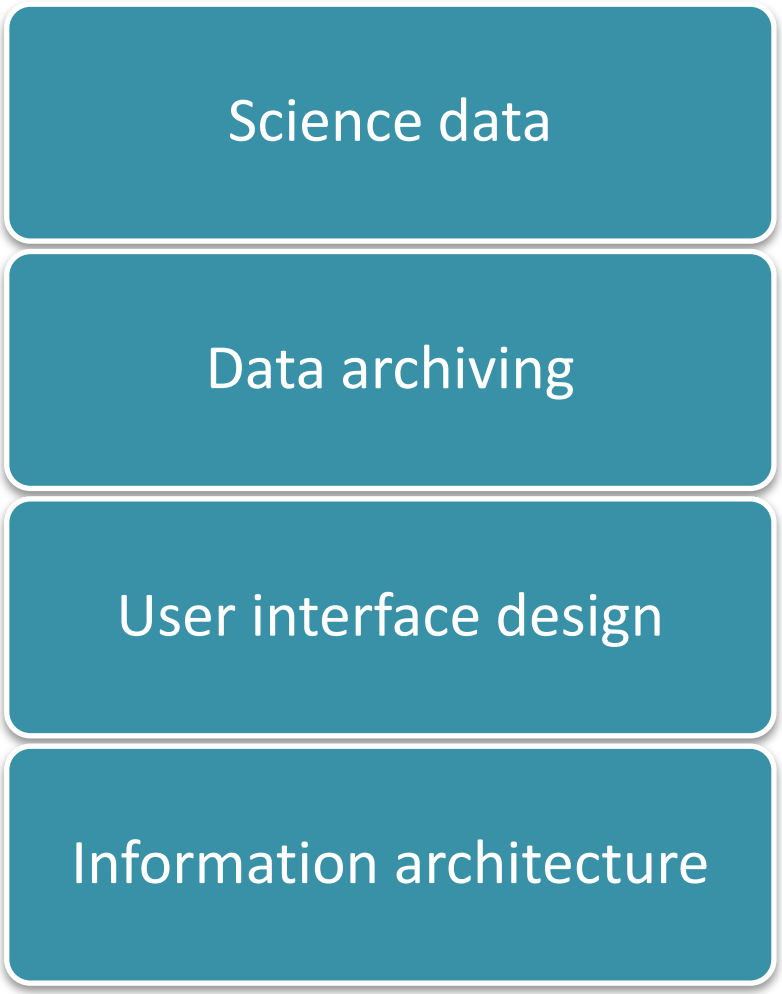

Spectral Library
Opportunity (MERB) Analyst's Notebook
Lunar Orbital Data Explorer
PDS Geosciences Node
Washington University in St. Louis
Mars Orbital Data Explorer
Apollo 16 Analyst's Notebook
Phoenix Analyst's Notebook
Phoenix Mission Historical Overview
Phoenix / Mars Reconnaissance Orbiter (MRO) Coordinated Observations
Sol 21

Developing web interfaces

- Electronic access is primary means for locating and disseminating science data
  - Directory browsing via FTP or HTTP
  - Browser-based interface (simple to complex)
  - Web services that allow client applications connections (e.g., ArcGIS, Google Earth)

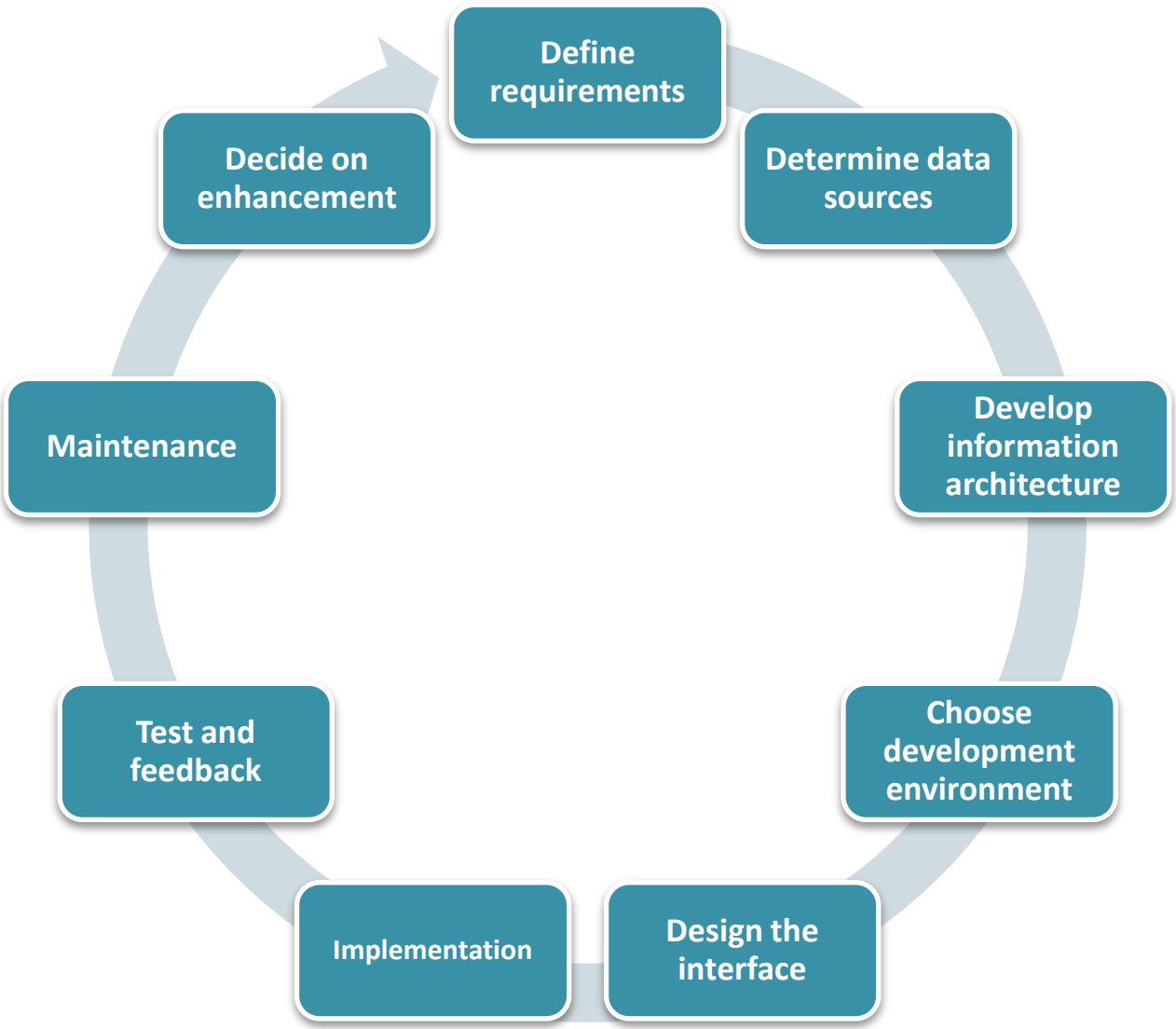
Keys to developing web interfaces	
Specialized knowledge	Iterative development process

Specialized knowledge domains



- Must expend resources to gain missing expertise
  - Research/training
  - Adding team members

Interface development lifecycle



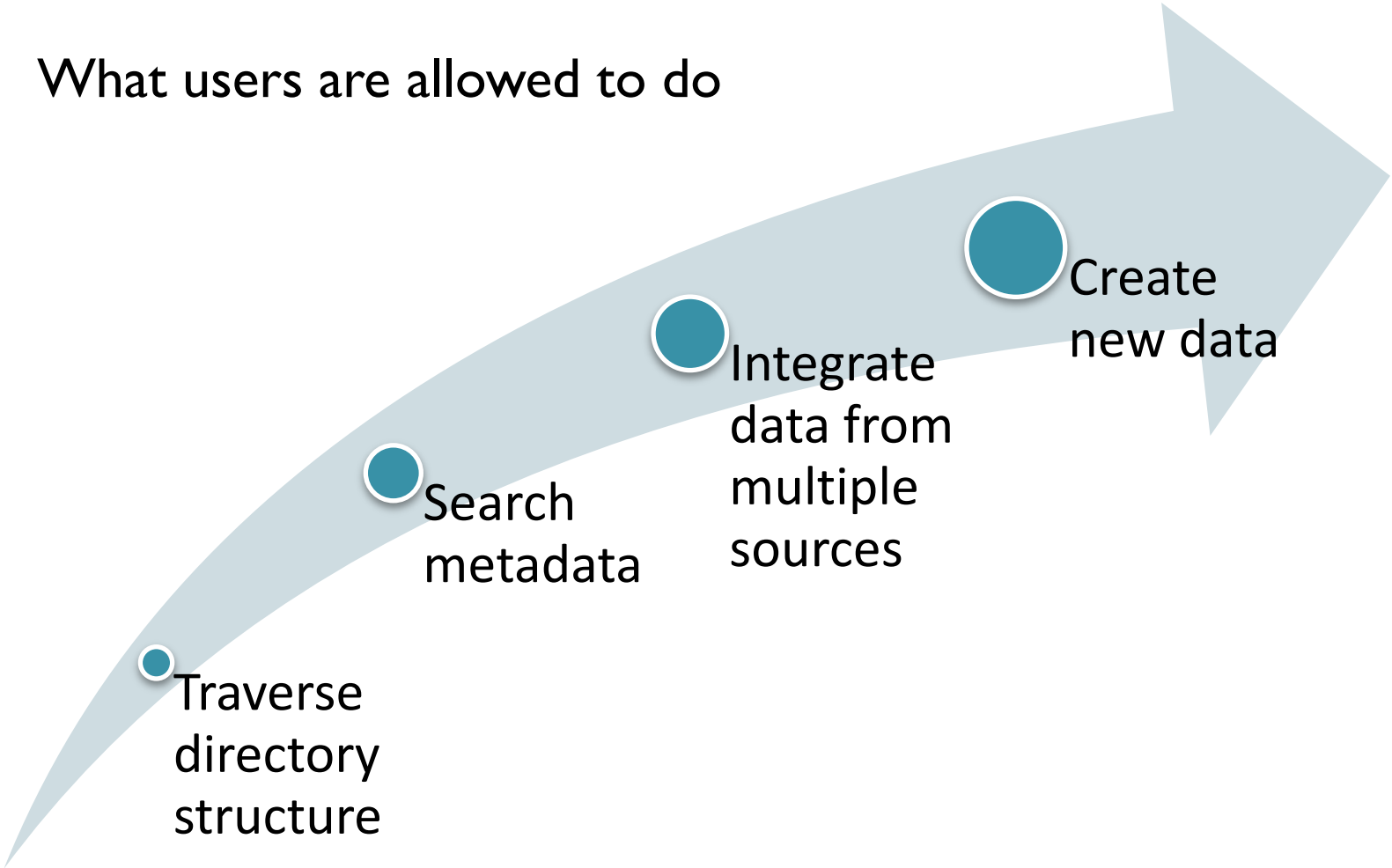
Define requirements

- Understand interface purpose and how it relates to science archive
  - What is intent?
  - What information is to be included?
  - How is information acquired?
  - Who are end users?
  - What are end users' expectations?
- Define
  - Data sources
  - Functionality
  - Deliverables
  - Security
  - Hardware/software limitations
  - Cost
- Create specific, attainable goals
- Incorporate use cases
  - Detail services, tasks, and functions of interface

Primary factors driving requirements	
Money resources	Time resources

Complexity

What users are allowed to do



Determine data sources

- More than just data files—integration of information from a number of sources
  - Data product metadata
  - Documentation
    - Data product documents (data provider)
    - Archive documents (data provider or archiving entity)
    - Additional documents from data collection phase (people directly and indirectly involved)

Example: daily reports from non-deterministic Mars rover and lander missions were captured to preserve knowledge and intent behind the decisions

Data quality

- Data must be well-formed and well-documented
  - Interface developers may have to invest considerable resources

> Example: data from older Mars Global Surveyor's MOC instrument were reprojected into areocentric projection used by newer Mars missions

- Recast data must be clearly labeled and described
  - Change in format
  - Subset into smaller parts
  - Used to create new product

Additional considerations

- Political
- Ethical
- Access rights
- Legal (ITAR – International Traffic in Arms Regulations)

Develop information architecture

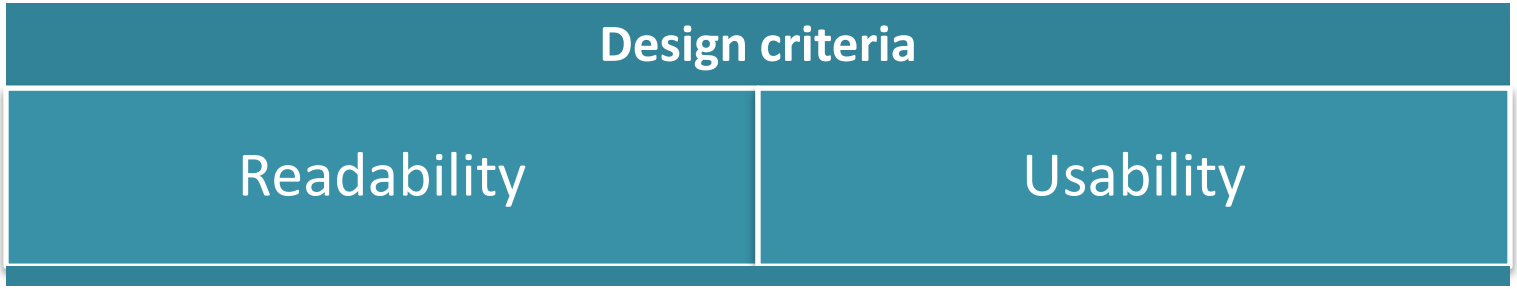
- Information architecture models system in which interface is developed and made operational
- Stakeholders interact with system in different ways
  - Data providers
  - End users
  - Developers
- End users will be affected by requirements placed on them
  - Is user required to download client software?
  - Is user required to have an account for access?
- Longevity and availability are important drivers

Choose development environment

- Factors in selection of development environment
  - Information architecture
  - Size and expertise of development team
  - Technologies required to support planned interface
- Development environment maturity
  - Increase long term stability of interface
  - Minimize effort required to maintain functionality

Design the interface

- Web interface is initial entry to an archive
- Bad design can impede users, even if there is useful functionality underneath
- Users' initial perception is significant factor in success of interface



Readability

- Readability elements may be considered hygiene factors

Color	Images	Other
• Color theory (principles of correct use) • Physical factors (color blindness) • Cultural color associations • Standard practices	• Icons provide subconscious cues • Too many images damage users' perceptions of professionalism	• White space • Navigation aids • Text • Font choice • Layout

Usability

"The capability of the software product to be understood, learned, used and attractive to the user…" *[ISO 9126-1]*

Learnability	Understandability	Operability
• First time users • Consistency and intuitiveness	• Expected user domain knowledge • Common terminology • Clear instructions	• How easily can a user find and retrieve what he or she is looking for?

Implementation, testing, and feedback

"The most well-defined interface can suffer from the reality of implementation" *[Merlyn, 1991]*

- New technology
- Schedule pressures
- Personnel changes
- Requirement "scope creep"

Testing	Feedback

• Ongoing process during design phase
• Begin early in implementation phase and occur regularly
• Use appropriate test group
• Include stress test
	• Seek from testers prior to release
	• Seek from users after release
	  • Online (e-mail, forum, survey)
	  • In person (science conferences)

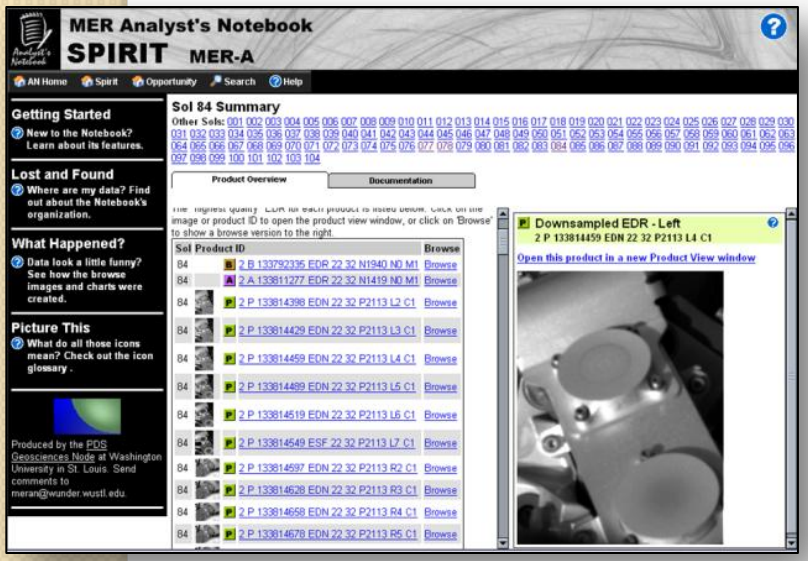
Maintenance

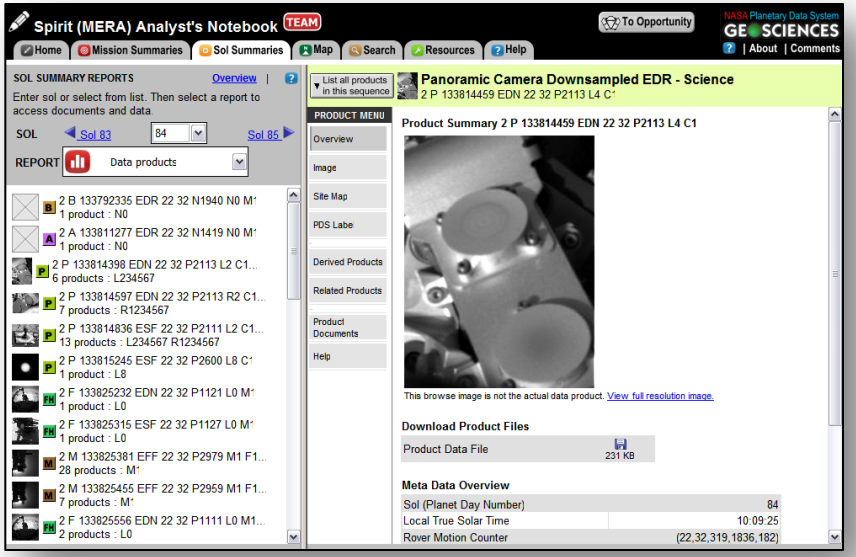
- Plan for longevity
  - Design interface to be viable for many years
  - Keep documentation and unit testing procedures up to date
  - Test interfaces regularly for broken links and loss of functionality
- Prepare for “progress”
  - Development platform and server hardware/operating system may become obsolete
  - Client web browser standards may change

Eventually, there will be a time when the interface must be updated, replaced, or retired.

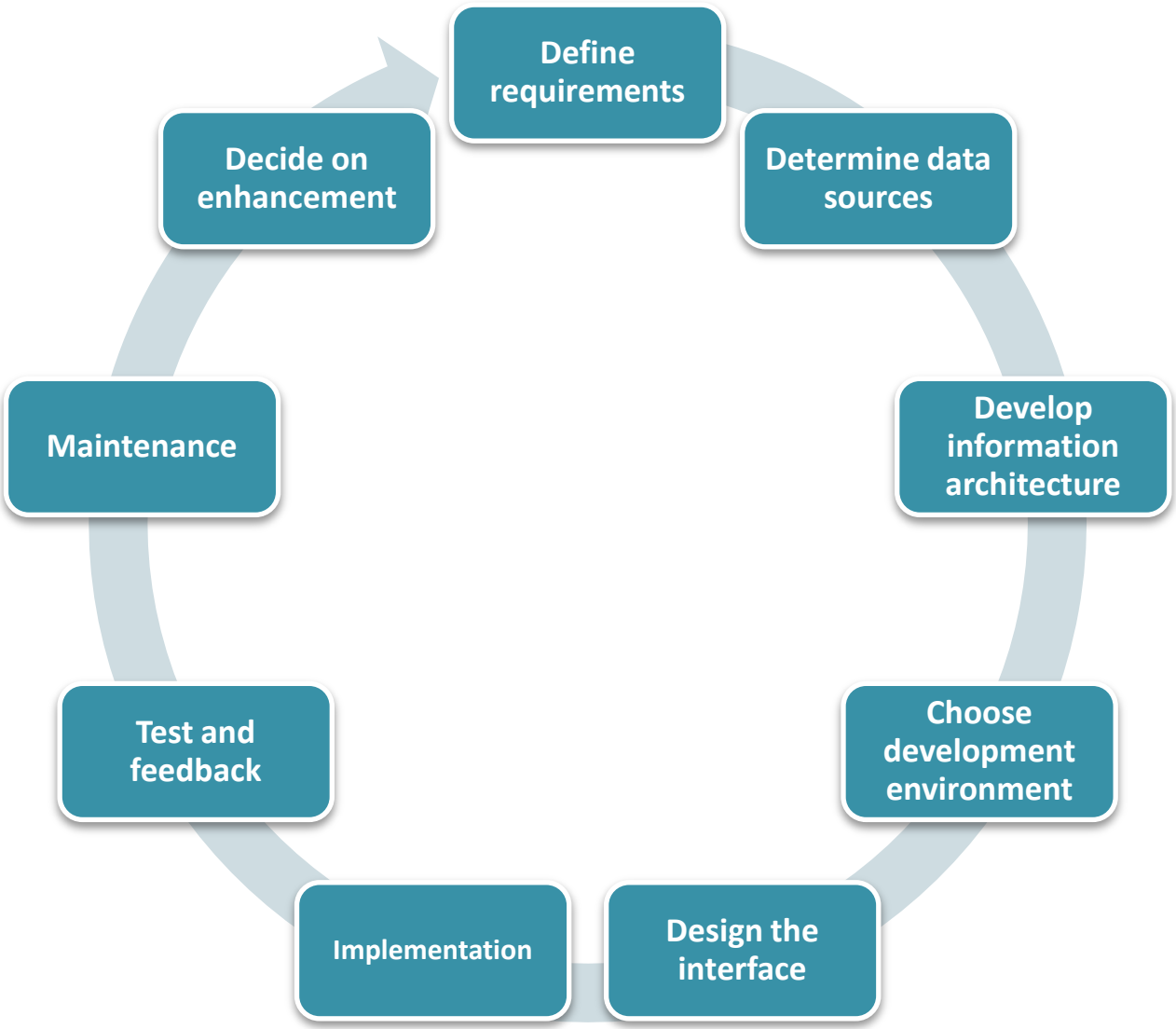
Enhancements

- Brought about by
  - Increased user abilities and expectations
  - Updates to technology supporting the interface
- Types of enhancements
  - New data from data providers
  - Improved versions of existing data
  - Interface enhancements based on user experiences and requests





Interface development lifecycle



Conclusion

- Require four knowledge domains
  - science data
  - data archiving
  - information architecture
  - user interface design
- Consider requirements in terms of resources
- Intentionally plan for longevity
- Involve all system stakeholders early in the process
- Maintain open communications throughout

QUESTIONS?