Migrating a Large Scale Legacy IT System to a System with Current Technology

Bill Wood, Mike Gagliardi, Phil Bianco

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0001020

Background

Agency wanted to Migrate two paired and tightly coupled systems

- Both are 24/7/365 and have disaster recovery at multiple sites
- Both have many classes of users

1. Service Based System - 15% functionality (going to 40%)
   - Java, Relational DB, COTS tools, Service-based, J2EE, SAP, Informatica
2. Legacy System - 85% of functionality (going to 60%)
   - COBOL, Hierarchical DB, mainframe

Migrate to a well defined target reference architecture (TRA) as a basis for a common platform infrastructure (CPI) – developmental, operational, test

- Briefing is focused on the Legacy Migration

Its Complicated

Understanding the Legacy System Architecture

- Infrastructure Tools
- Application Component Relationships (Data and Code)
- Business Process Threads

Understanding the Target System

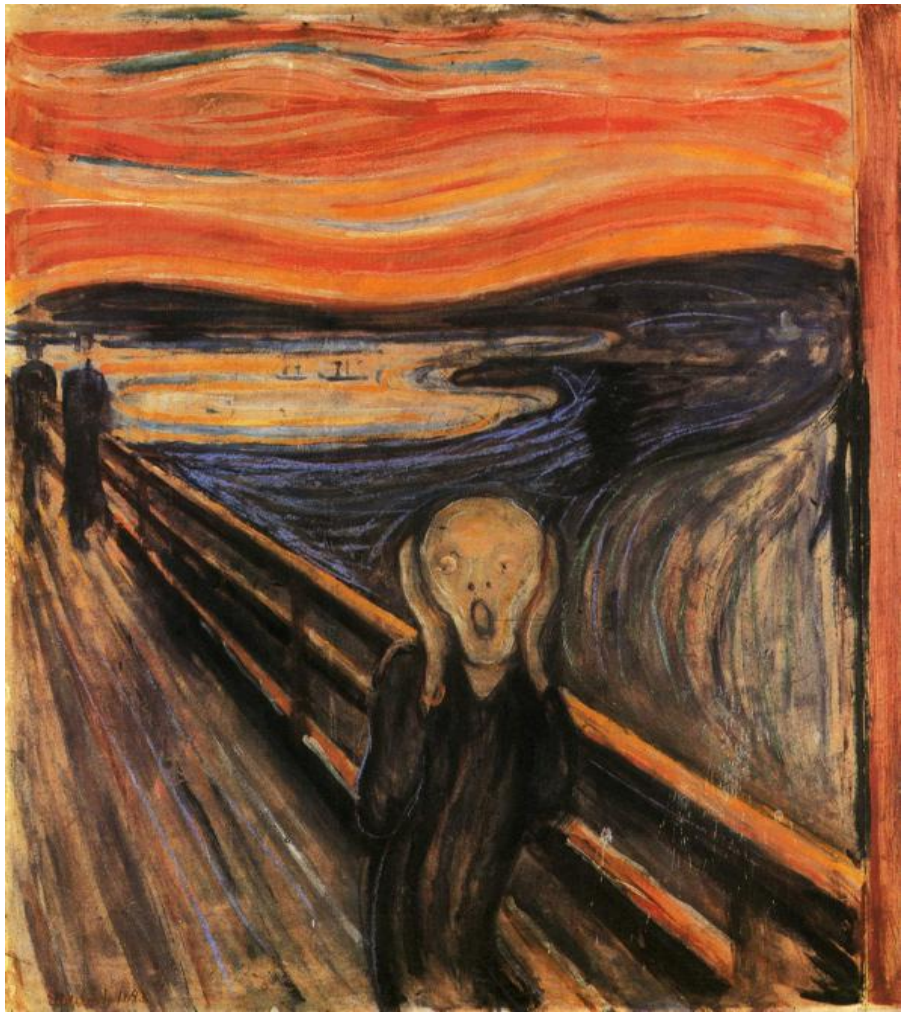
- Target Reference Architecture (TRA)
  - SOA; Layered Infrastructure
- Designing Services on top of TRA
- Business Process threads

Mapping between Legacy and Target in Phases

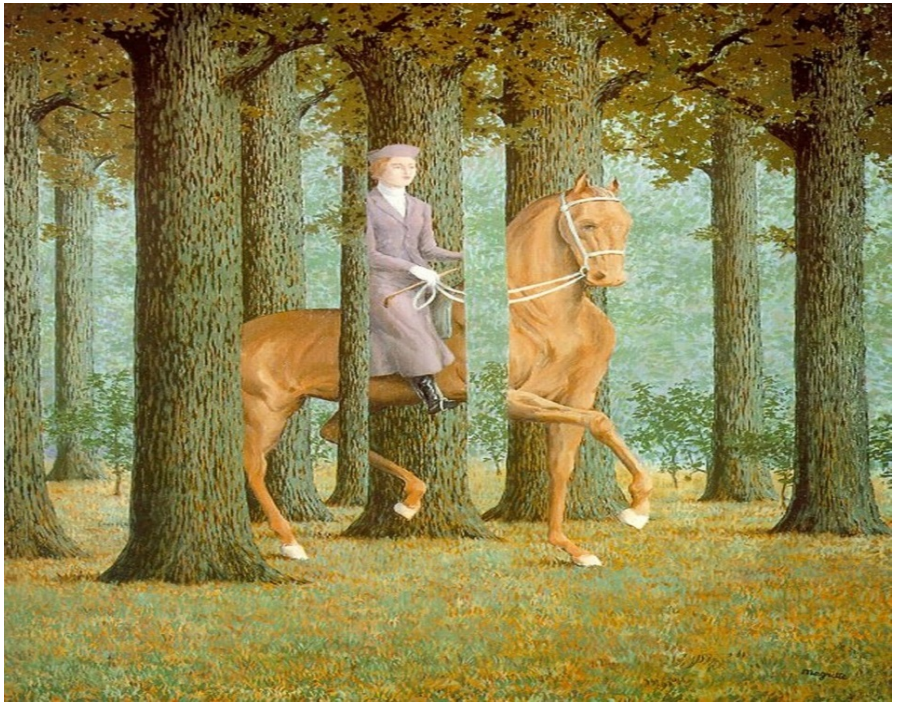
- Architecture mismatches (development, operational, certification, sustainment, COTS)
- Operating with dual authoritative data systems, cutover, synchronization
- Relationships between Application Components (Legacy versus TRA, COTS)

It's Worrisome

- **Is there too much code/ data coupling and spaghetti code to partition for migration?**
- Are the current business processes and screens appropriate?
- Is the CPI (and TRA) stable?
- Are COBOL to Java transformation tools up to the job?
- Can we operate with 2 systems overlapping authoritative data?
- Do we have sufficient technology expertize in: legacy system, target reference architecture, discovery and analysis tools, transformation tools?
- **Is the business logic only understandable in the legacy code?**
- How can we overcome the lack of architectural documentation?
- Will the entire testing and certification process change?
- **Will I be creating a maintenance nightmare?**
- It's a 24/7/365 operation- no downtime!



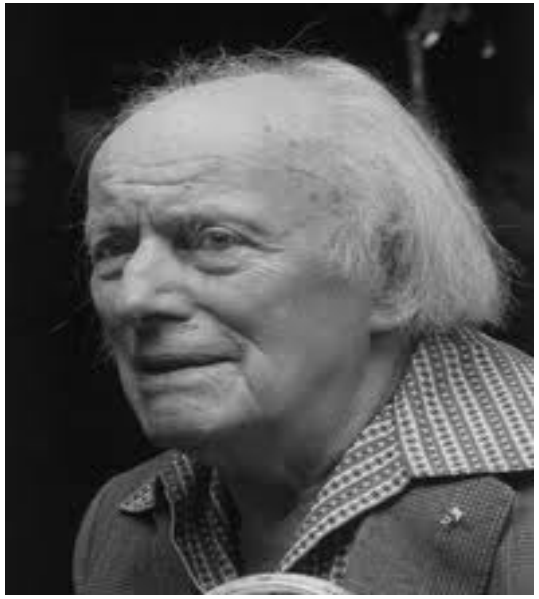
Keep the Right Vision



- **The TRA is a good start, but an application architecture with data modeling is needed**
- **Synchronize with phased TRA and CPI infrastructure deployment**
- Operating with multiple sources of authoritative data
- **Using new technology (e.g. customized security to infrastructure security)**
- Capability of transformative toolsets

Don’t Live in a Dreamworld

- Big-Bang changes usually fail
- **Conducting transactions across networks and keeping response times satisfied!**
- Transforming spaghetti code automatically
  - Separating presentation from business processing from data access!
  - Moving from green screens to windows
- **Making multiple types of changes simultaneously!**
- **Edict no changes to the legacy system!**



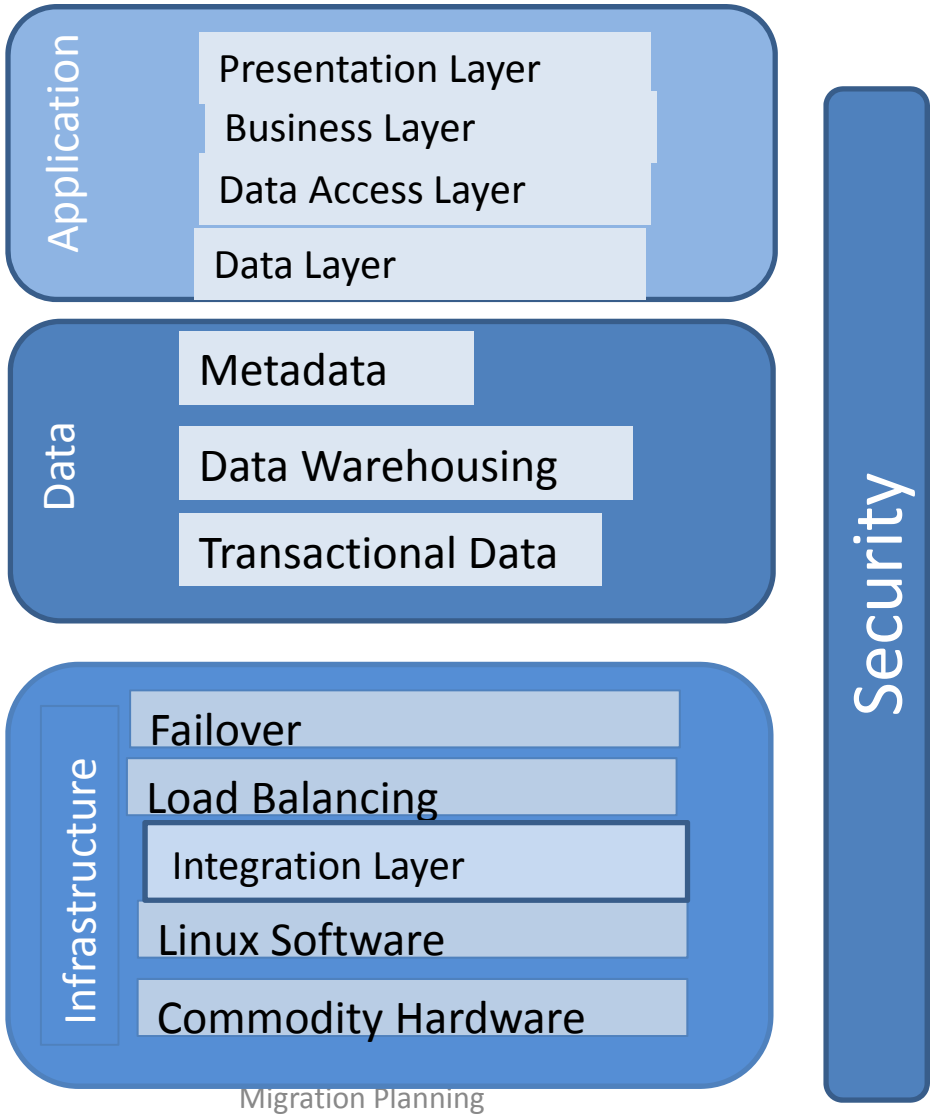
Paul Delvaux

Target Reference Architecture

This is a good start
But it is not enough

Need a system and application software architecture

- End-state
- Each Release



Migration Approaches

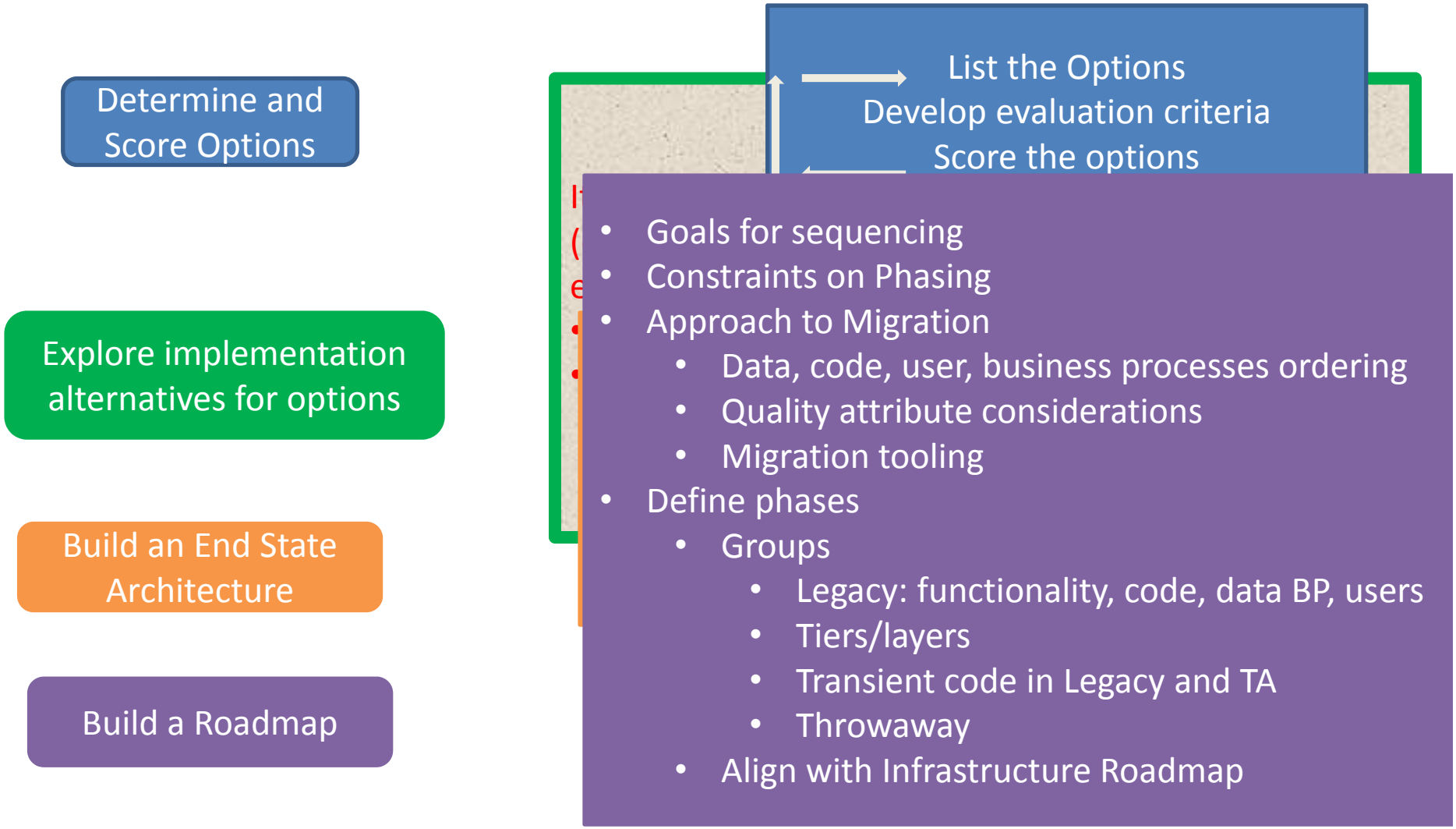
An RFI was conducted and multiple approaches suggested

- Lift and Shift (L&S)
  - Reduce Operations and Mainten
    - Fast retirement of legacy infrastru
  - Move to the CPI
    - Don't come close to satisfying the
    - Make minimal changes- keep COBC
- Modernize
  - Phased transformation to TRA
    - Keep legacy architecture
    - Change technology- Java, S
    - Use Conversion tools?
- Re-engineer
  - Re-Develop the system in Java or
  - Developed system and application architecture
  - Satisfy TRA

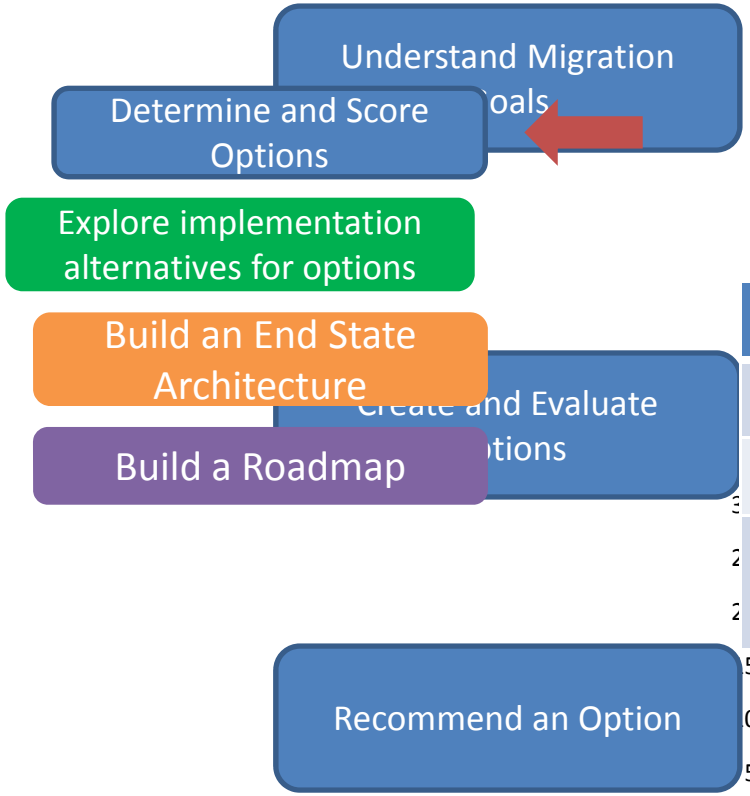
Options Considered

1. Do nothing (baseline)
2. L&S (baseline)
3. L&S and modernize
4. L&S and re-engineer
5. Re-engineer
6. Hybrid- L&S, modernize, re-engineer

Migration Process



Determine and Score Options



Goal- Examples

Reduce operation and maintenance (O&M) costs
- Licensing, obsolescence, electric, space, people
- Vendor lock

Create technical uniformity across organizational boundaries

Options

6 mentioned previously

Initial crude evaluation against TRA

Detailed evaluation against
- Cost, risk, performance factors (23)

5.00
0.00
5.00
0.00

Choose an Option

Best Overall Measure - cost, risk, performance

Determine ordering: sequential, concurrently

Important Developmental and Sustainment QAs

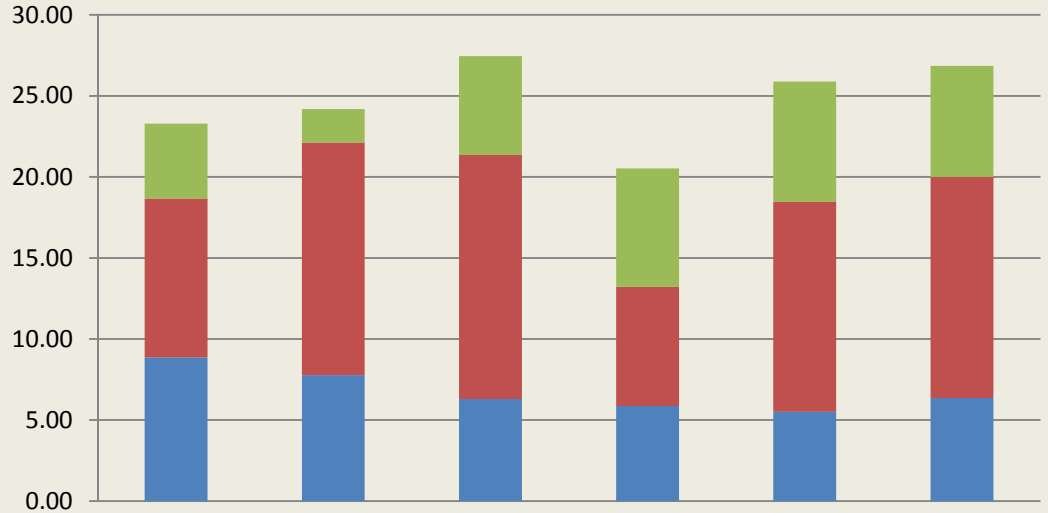
Business Process Threads
- Users, code , data, interfaces

Evaluation Factors

Cost

- How much will be the migratic labor cost?
- How much recovery of investment cost after cancellation
- How early will cost savings happen- license and suppo during migration?
- How much will be the on-g legacy license and/or suppo costs post migration?

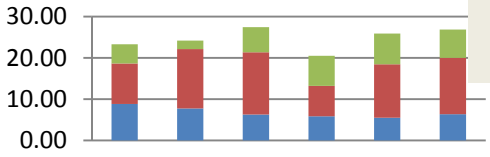
Performance



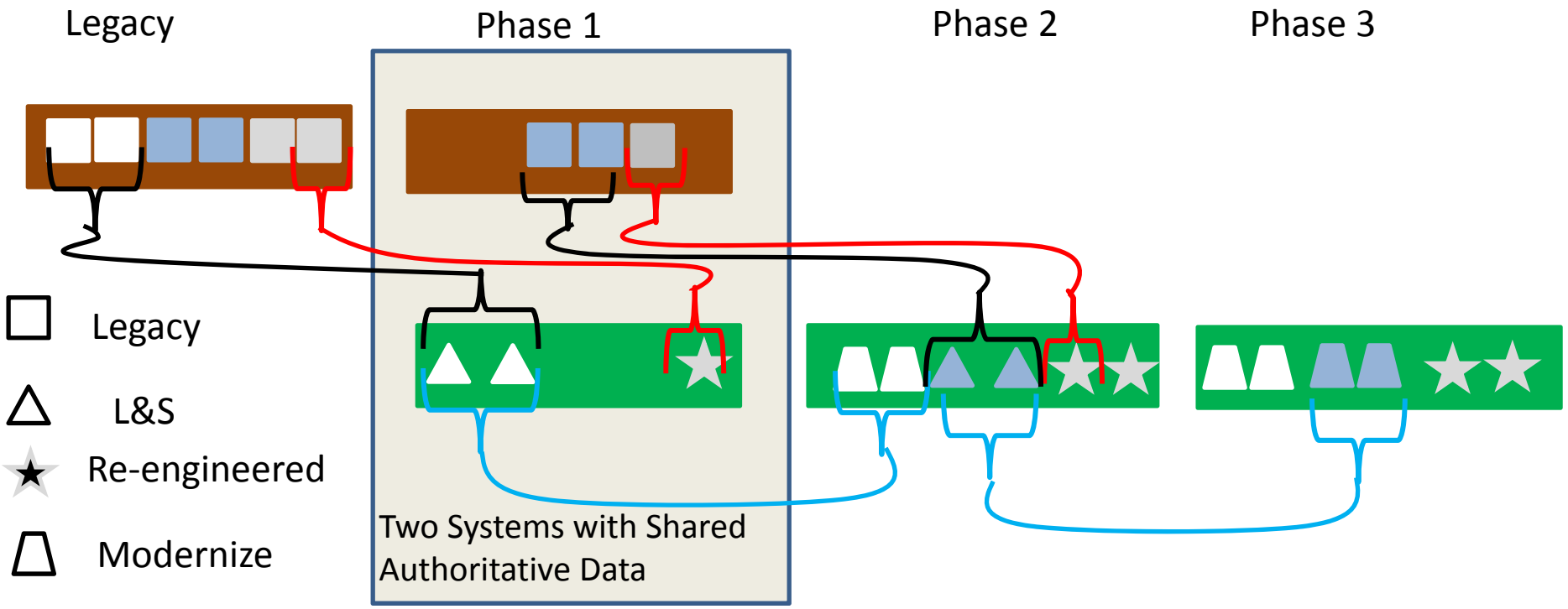
- How modernized are the user screens wrt the TRA

Risk

W	Factor	1	2	3
3	How is data organized wrt TA : relational, normalized, distributed, partitioned	1	1	5



Selected Hybrid Option- Example



Modernize

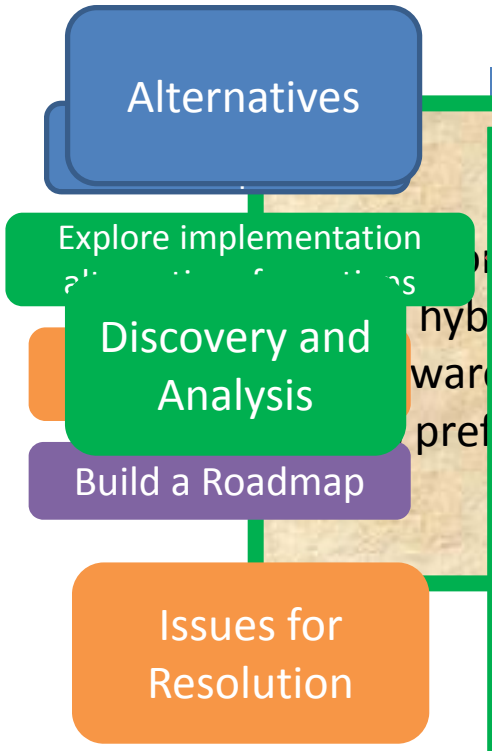
Lift and Shift

Re-engineer

How to partition: L&S and modernize; and Re-engineer

- Significant Business Process Changes – re-engineer
- Layered stable legacy code- L&S and modernize
- Experiment with modernization tools

Explore Implementation Alternatives- Lift and Shift



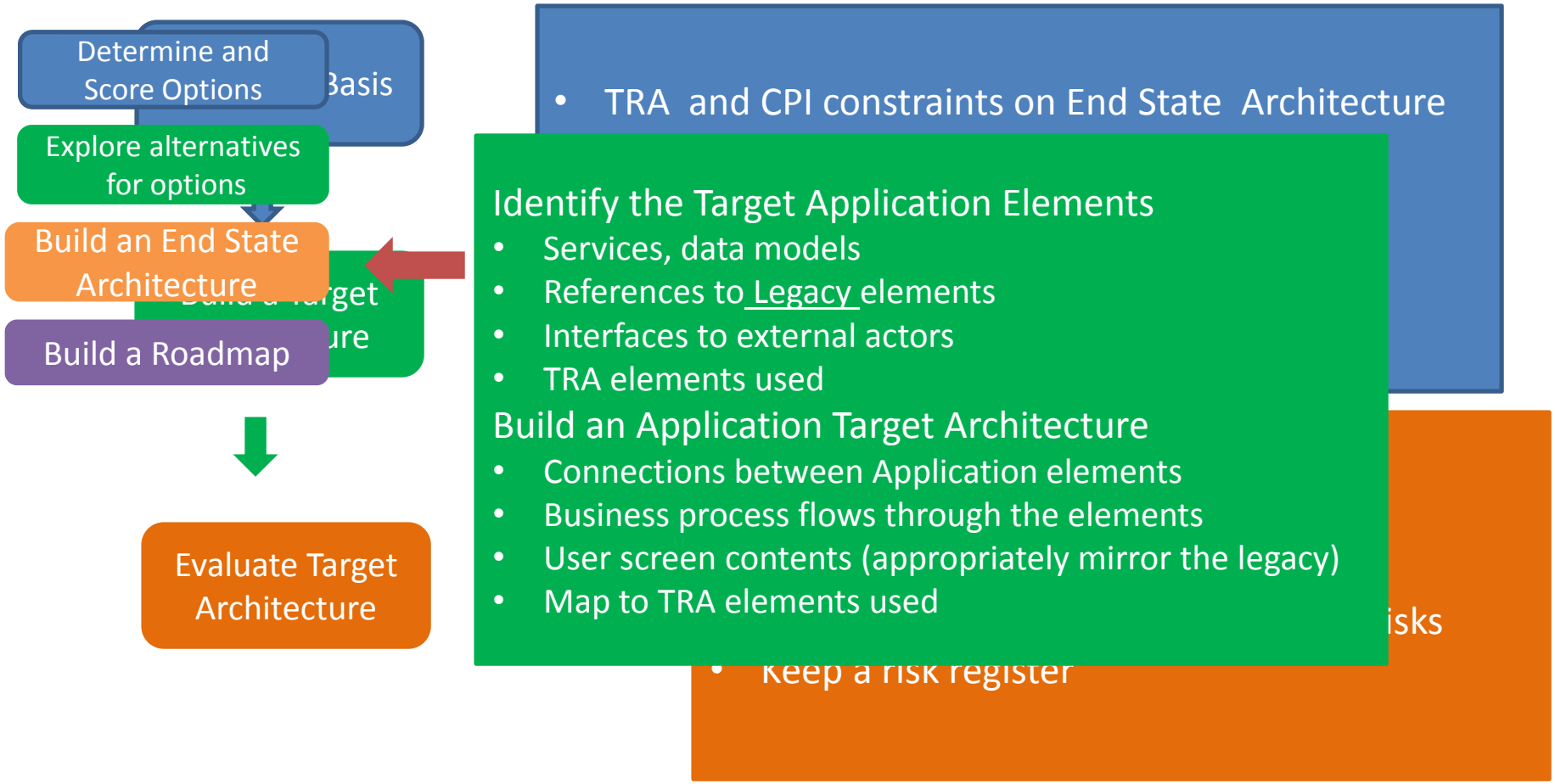
Finds the relationships between

- The subsystems and their programs
- The Data Stores and their files
- The subsystems and the data stores
- End-to-end business threads/User classes/subsystems/data stores
- Identify Dead code and dead data
- Legacy API's usage
- Legacy transactions usage
- Batch usage

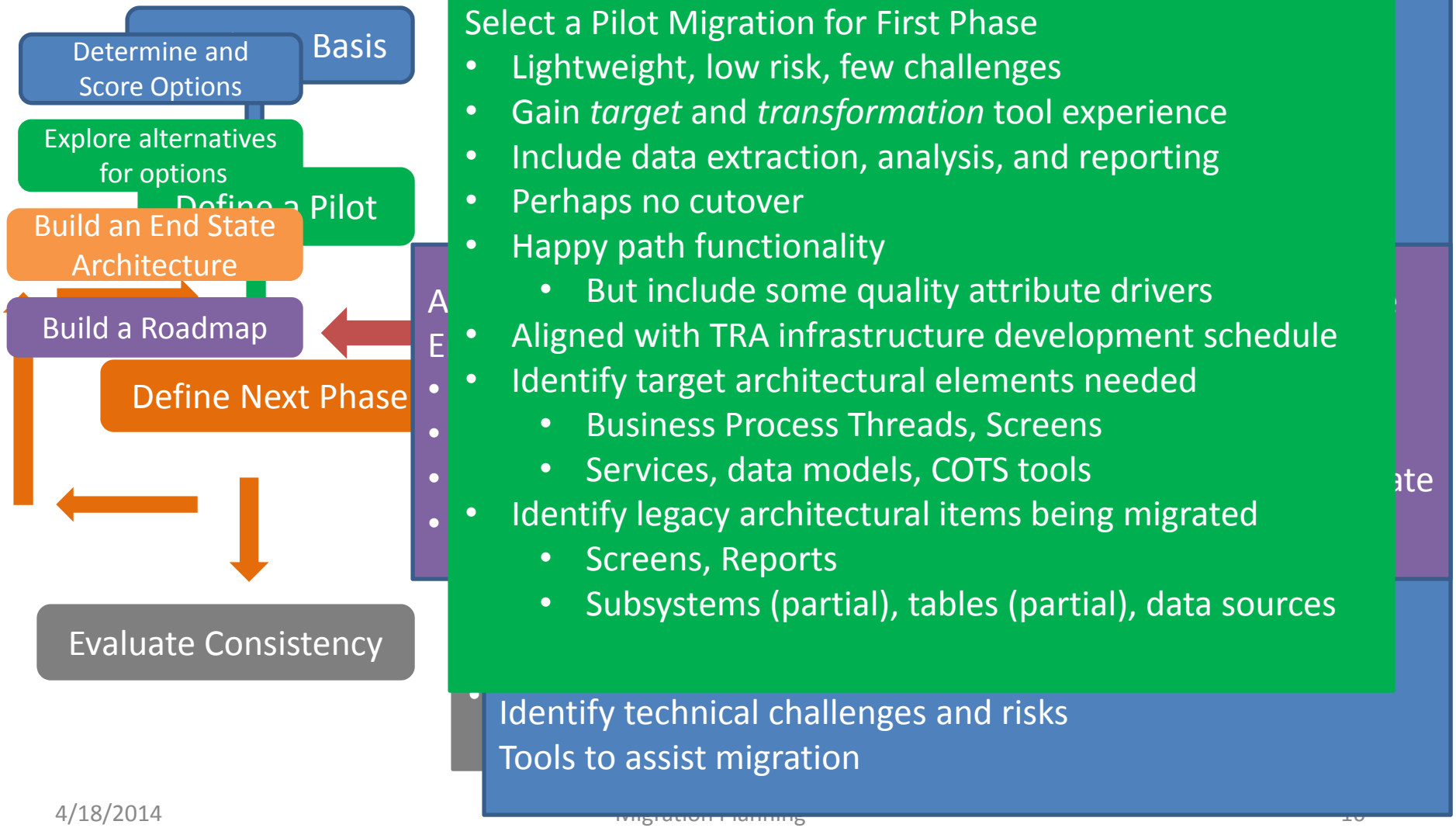
What commercial toolsets are best to do this?

	copying data, modifying APIs & JCL, writing adaptors?	
3	Replacing transactional support	
4	Should we do some cleanup before L&S	
5	Performance	

Build an End State Architecture



Build an Architectural Roadmap



Summary

Technical

- Criteria for selecting between options
  - OMB guidelines were good
- Plan, but don't overdo it
- Architect
  - Understand Legacy
  - Include quality attributes
  - High Level End State
- Roadmap
  - Detailed architecture at each phase

Management

- Changing political environment
- Drift and Re-direction can become a way of life