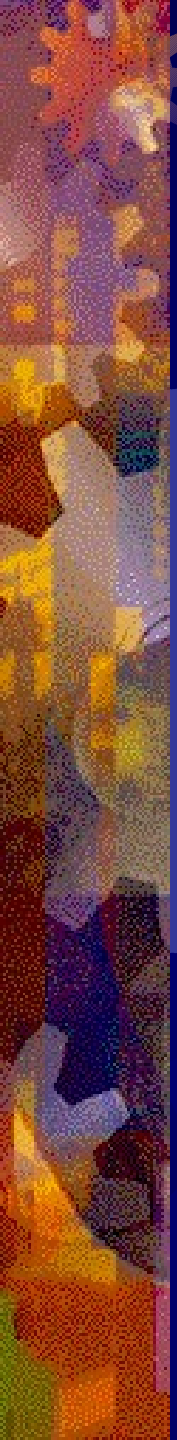


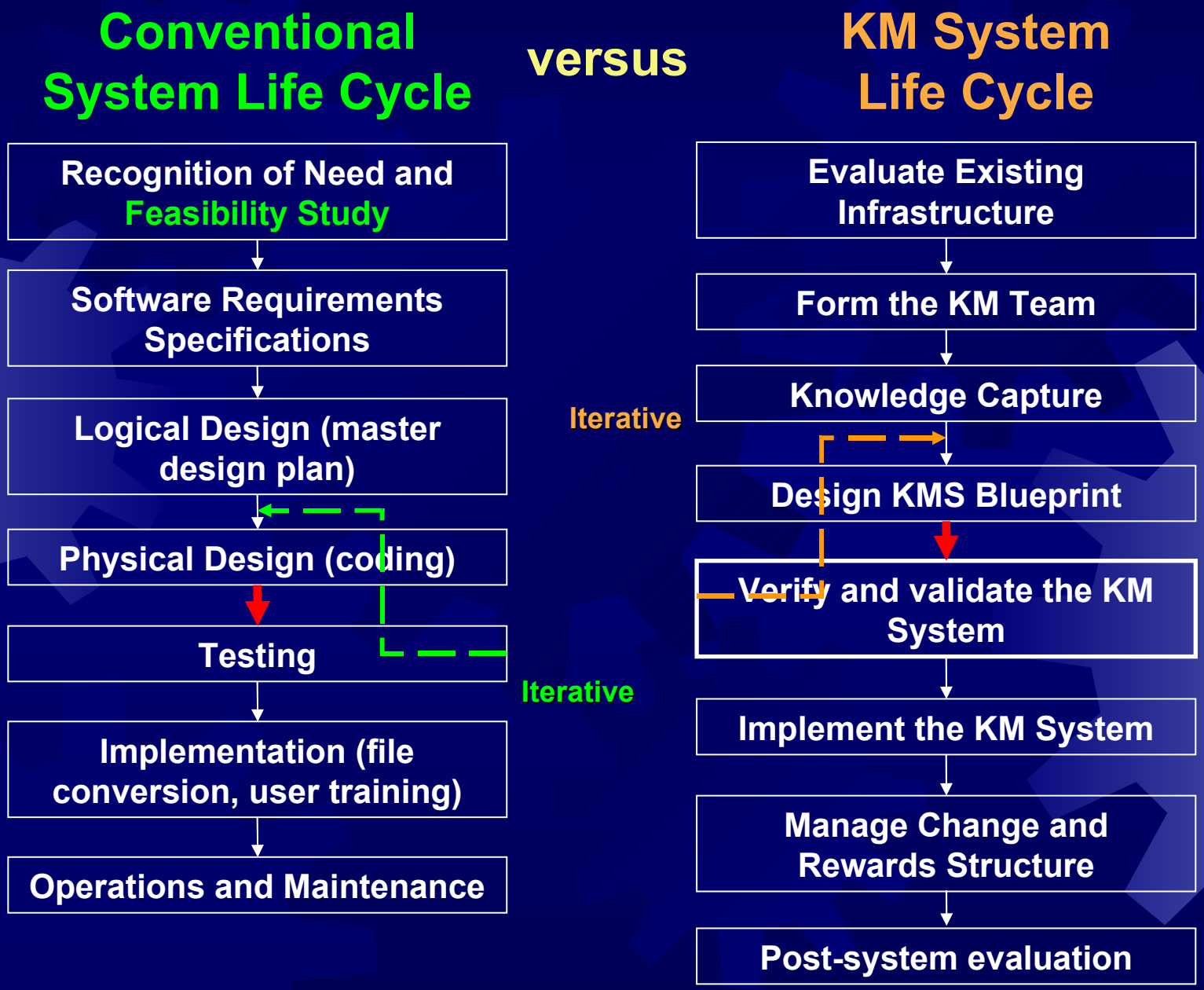
KNOWLEDGE MANAGEMENT SYSTEMS LIFE CYCLE

Main Topics of This Lecture

- Challenges in building KM Systems
- Compare CSLC and KMSLC
- User's vs. Expert's Characteristics
- Stages of KMSLC

CHALLENGES IN BUILDING KM SYSTEMS

- **Culture**
  - — getting people to share knowledge
- **Knowledge evaluation**
  - — assessing the worth of knowledge across the firm
- **Knowledge processing**
  - — documenting how decisions are reached
- **Knowledge implementation**
  - — organizing knowledge and integrating it with the processing strategy for final deployment


Conventional System Life Cycle
versus
KM System Life Cycle
Recognition of Need and Feasibility Study
Software Requirements Specifications
Logical Design (master design plan)
Physical Design (coding)
Testing
Implementation (file conversion, user training)
Operations and Maintenance
Iterative
Iterative
Evaluate Existing Infrastructure
Form the KM Team
Knowledge Capture
Design KMS Blueprint
Verify and validate the KM System
Implement the KM System
Manage Change and Rewards Structure
Post-system evaluation

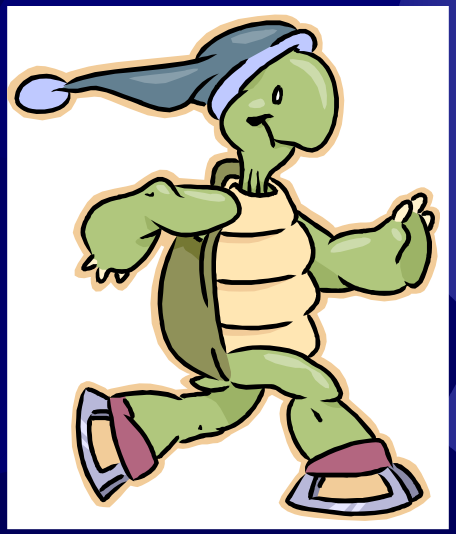
Key Differences

- ***Systems analysts*** deal with information from the user; ***knowledge developers*** deal with knowledge from domain experts
- ***Users*** know the problem but not the solution; ***domain experts*** know both the problem and the solution
- System development is primarily ***sequential***; KMSLC is ***incremental and interactive***.
- System testing normally at ***end*** of conventional system life cycle; KM system testing ***evolves from beginning*** of the cycle

Key Differences (cont'd)

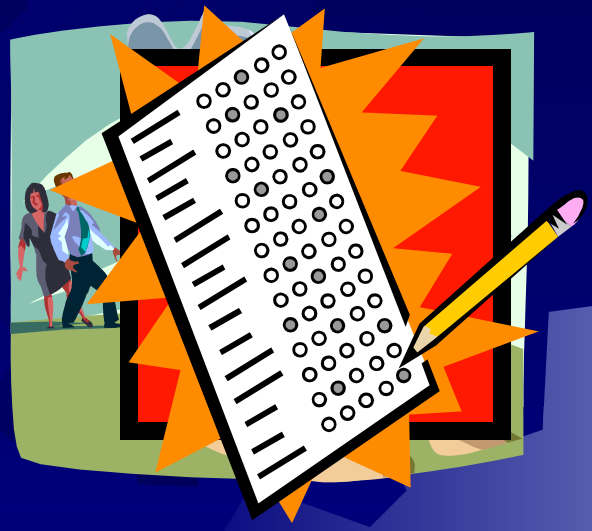
- Conventional system life cycle is ***process-driven*** "specify then build";

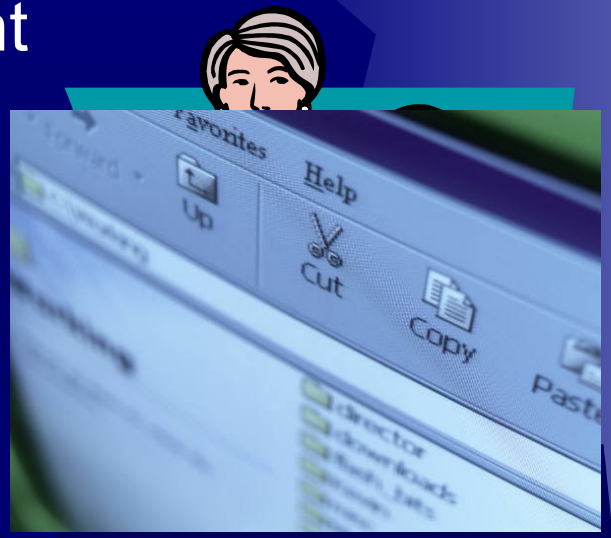
- KMSLC is ***result-oriented*** "start slow and grow"



Key Similarities

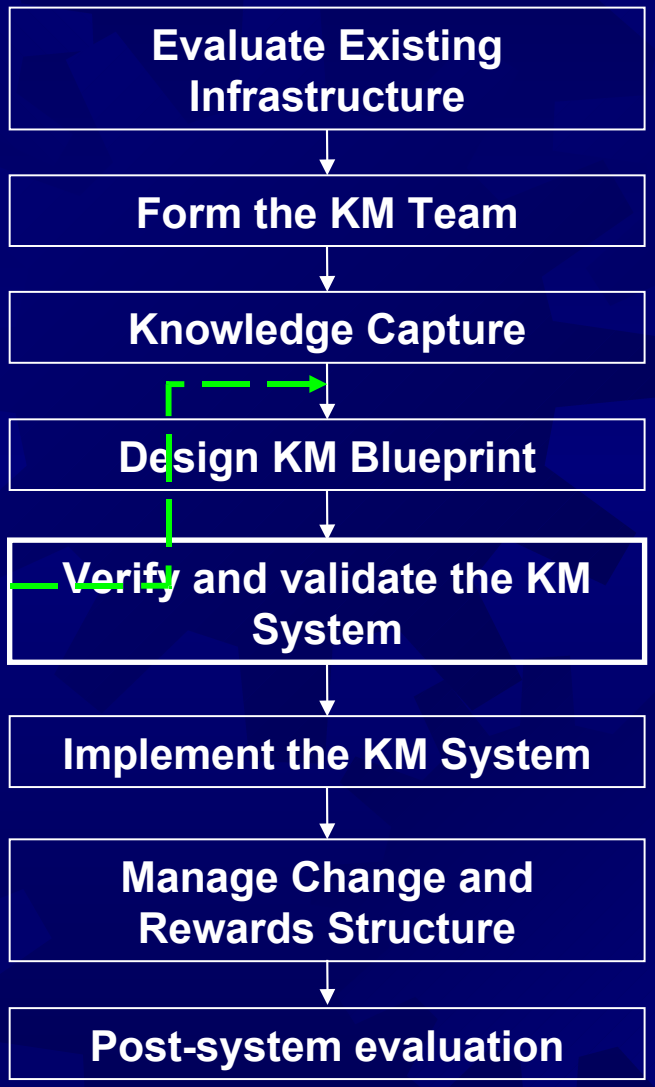
- Both begin with a problem and end with a solution
- Both begin with information gathering or knowledge capture
- Testing is essentially the same to make sure the system is right and it is the right system
- Both developers must choose the appropriate tool(s) for designing their respective systems





Stages of KMSLC

Iterative Rapid



(1) Evaluate Existing Infrastructure

System justifications:

- What knowledge will be lost through retirement, transfer, or departure to other firms?
- Is the proposed KM system needed in several locations?
- Are experts available and willing to help in building a KM system?
- Does the problem in question require years of experience and tacit reasoning to solve?

The Scope Factor

- Consider breadth and depth of the project within financial, human resource, and operational constraints
- Project must be completed quickly enough for users to foresee its benefits
- Check to see how current technology will match technical requirements of the proposed KM system

Role of Strategic Planning

- Risky to plunge into a KMS without strategy

Knowledge developer should consider:

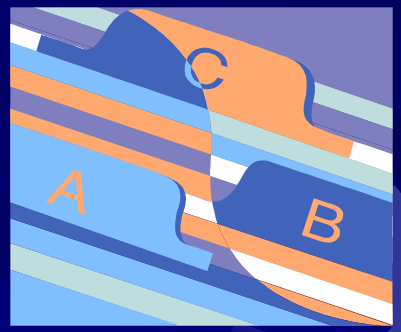
- ***Vision*** — Foresee what the business is trying to achieve, how it will be done, and how the new system will achieve goals
- ***Resources*** — Check on the affordability of the business to invest in a new KM system
- ***Culture*** — Is the company's political and social environment amenable to adopting a new KM system?

(2) Form the KM Team

- Identify the ***key stakeholders*** of the prospective KM system.
- Team success depends on:
  - Ability of team members
  - Team size
  - Complexity of the project
  - Leadership and team motivation
  - Not promising more than can be realistically delivered

(3) Knowledge Capture

- **Explicit** knowledge captured in repositories from various media
- **Tacit** knowledge captured from company experts using various tools and methodologies
- Knowledge developers capture knowledge from experts in order to build the **knowledge base**



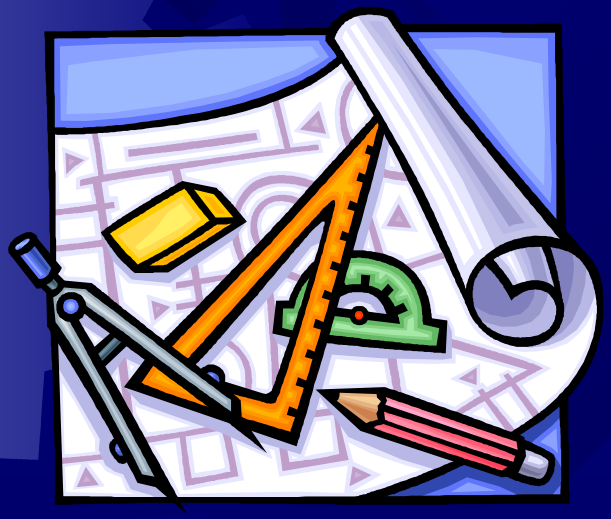
Selecting an Expert

- How does one know the expert is in fact an expert?
- How would one know that the expert will stay with the project?
- What backup should be available in case the project loses the expert?
- How could we know what is and what is not within the expert's area of expertise?

Role of the Knowledge Developer

- The **architect** of the system
- Job requires *excellent communication skills*, *knowledge of capture tools*, *conceptual thinking*, and a *personality that motivates people*
- Close contacts with the champion
- Rapport with top management for ongoing support

(4) Design the KM Blueprint

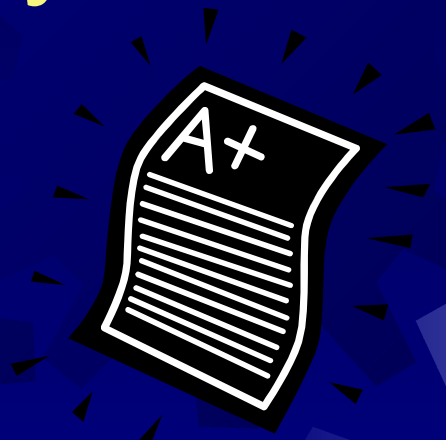


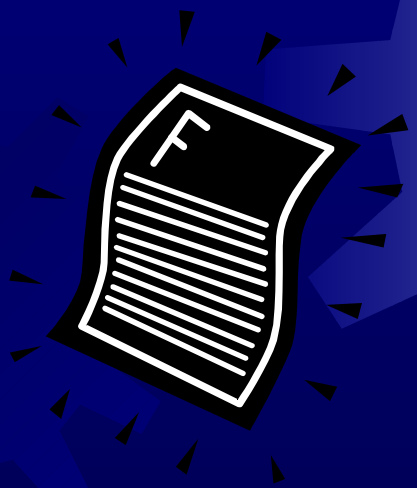
The KM blueprint addresses several issues:

- Finalize scope of proposed KM system with realized net benefits
- Decide on required system components
- Develop the key layers of the KM software architecture to meet company requirements
- System interoperability and scalability with existing company IT infrastructure

(5)Testing the KM System

- ***Verification*** procedure: ensures that the system has the right functions
- ***Validation*** procedure: ensures that the system has the right output
- **Validation of KM systems is not foolproof**





(6) Implement the KM System

- *Converting* a new KM system into actual operation
- This phase includes *conversion* of data or files
- This phase also includes user training
- **Quality assurance is important**, which includes checking for:
  - *Reasoning errors*
  - *Ambiguity*
  - *Incompleteness*
  - *False representation (false positive and false negative)*

(7) Manage Change and Rewards Structure

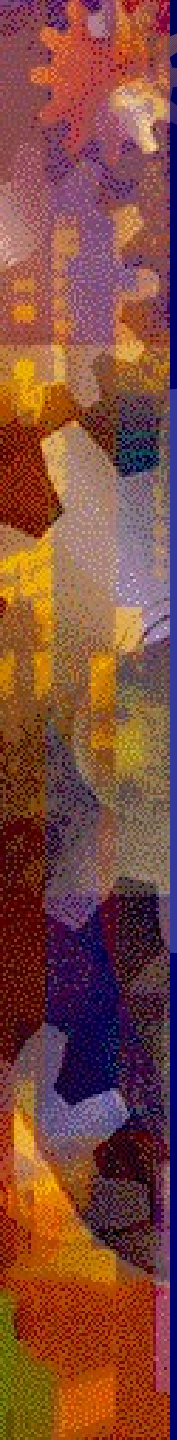
- Goal is to minimize resistance to change
- Experts
- Regular employees (users)
- Troublemakers
- Resistances via projection, avoidance, or aggression

(8) Post-system Evaluation

- Assess system impact in terms of effects on:
  - People
  - Procedures
  - Performance of the business
- Areas of concern:
  - Quality of decision making
  - Attitude of end users
  - Costs of Knowledge processing and update

Key Questions

- Has accuracy and timeliness of decision making improved?
- Has KMS caused organizational changes?
- What are users' reactions towards KMS?
- Has KMS changed the cost of operating the business?
- Have relationships among users affected?
- Does KMS justify the cost of investment?



End of Lecture 2

- **Purpose**
- **Statement of Scope & Objectives**

2.1 System functions
2.2 Users and characteristics
2.3 Operating environment
2.4 User environment
2.5 Design/implementation constraints
2.6 Assumptions and dependencies

3. Functional Requirements

3.1 User interfaces
3.2 Hardware interfaces
3.3 Software interfaces
3.4 Communication protocols and interfaces

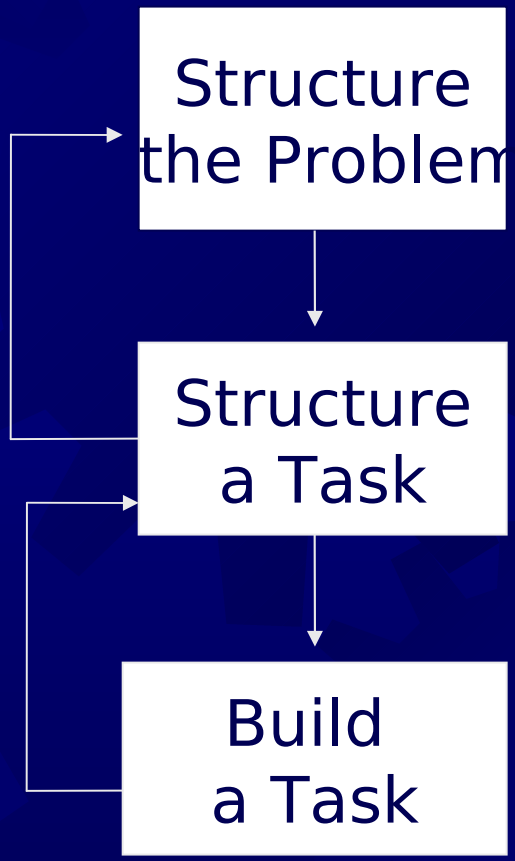
4. Nonfunctional Requirements

4.1 Performance requirements
4.2 Safety requirements
4.3 Security requirements
4.4 Software quality attributes
4.5 Project documentation
4.6 User documentation

Users Versus Experts

Attribute	***User***	***Expert***
Dependence on system	High	Low to nil
Cooperation	Usually cooperative	Cooperation not required
Tolerance for ambiguity	Low	High
Knowledge of problem	High	Average/low
Contribution to system	Information	Knowledge/expertise
System user	Yes	No
Availability for system builder	Readily available	Not readily available

Rapid Prototyping Process?



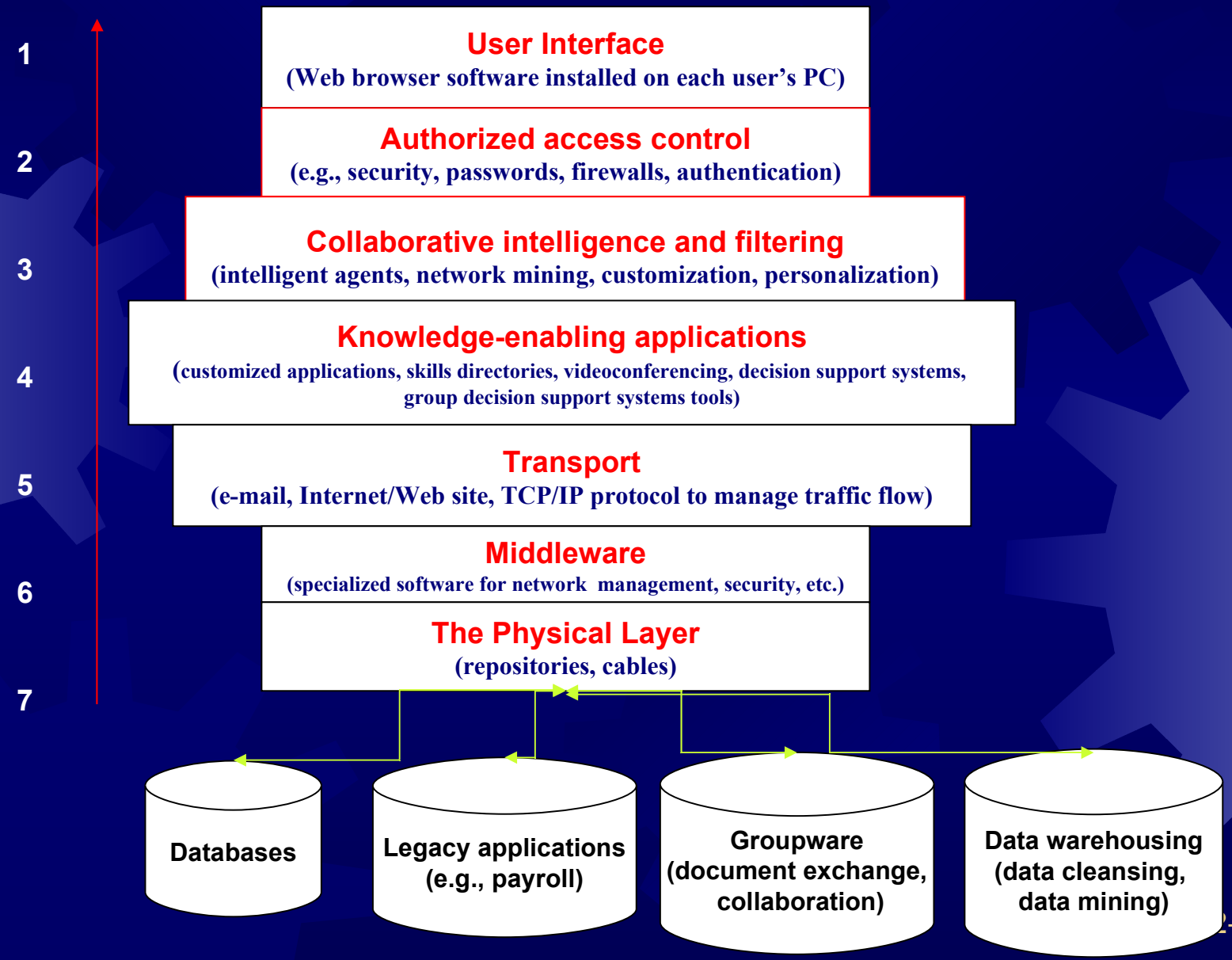
Reformulate the Problem

Repeated Cycle(s)

Make Modifications

Repeated Cycle(s)

Layers of KM Architecture



Knowledge Capture and Transfer Through Teams