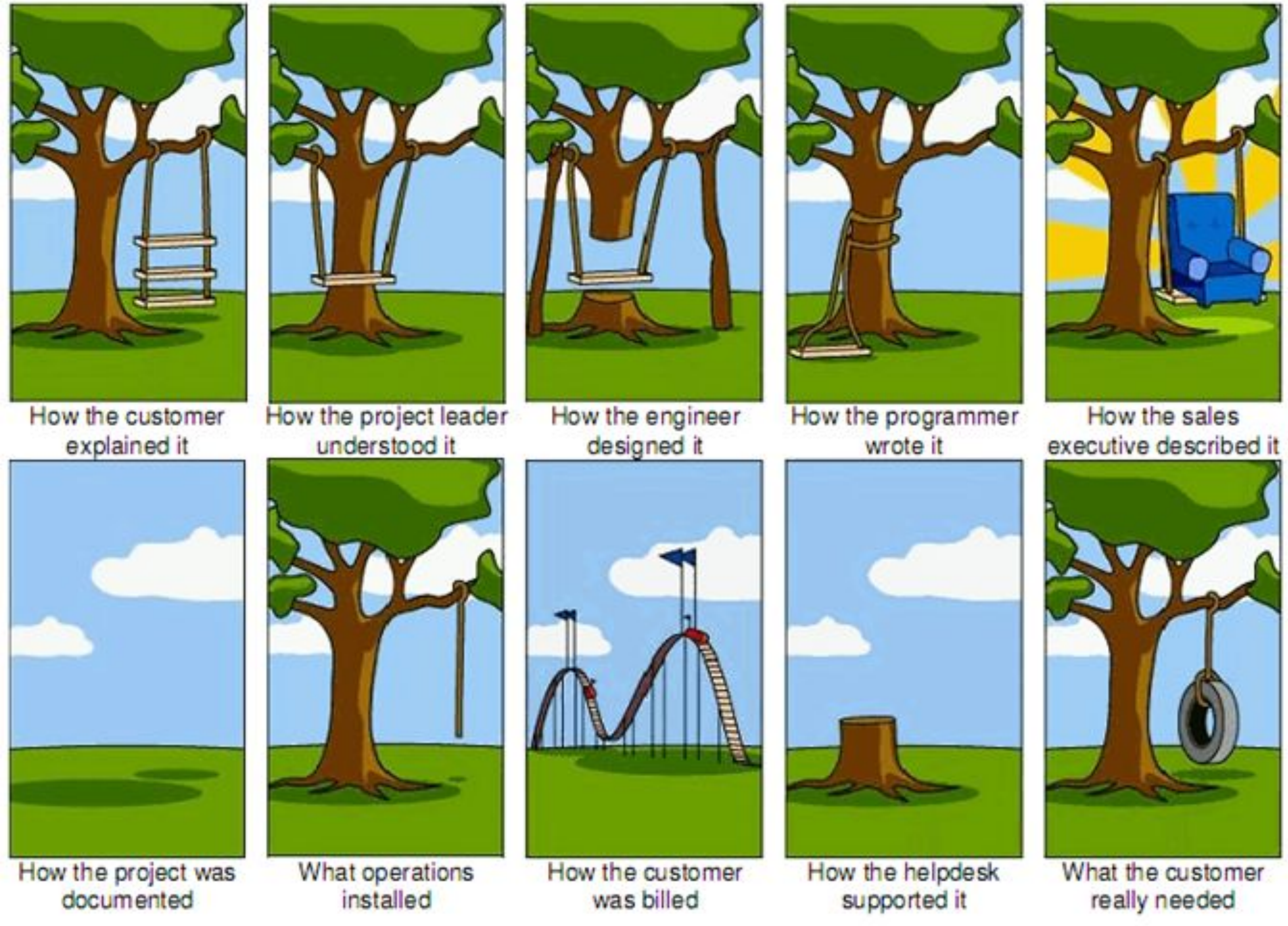
Requirements in the corporate context

Preparing for real-world software engineering

Developed by: Steven Jacobs, Eck Doerry

Consequences of Bad Requirements Engineering


How the customer explained it
How the project leader understood it
How the engineer designed it
How the programmer wrote it
How the sales executive described it
How the project was documented
What operations installed
How the customer was billed
How the helpdesk supported it
What the customer really needed

http://www.knovelblogs.com/2012/08/30/the-importance-of-requirements-engineering/

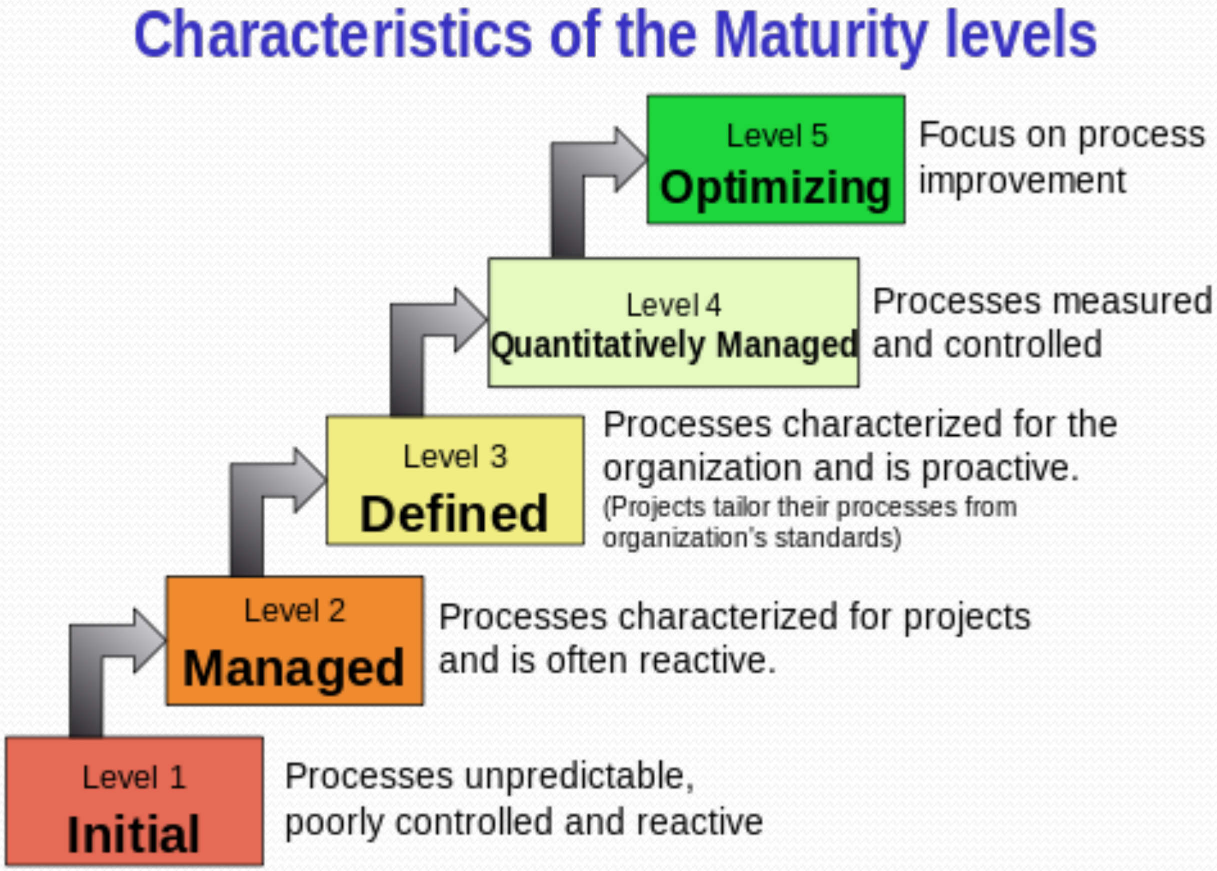
Building an efficient organization

- **Success in modern business world depends on optimization**
  - Correct, consistent operational, production and business processes
  - Continuous monitoring, evaluation, improvement
  - In any contracting biz, “requirements engineering” is the central process
    - Requirements Development: Clear effective process for figuring out what is needed.
    - Requirements Management: Tracing those (changing) requirements through to final product.
  - In software biz, requirements focuses on “software requirements”

- **How to measure how effective your organization is?**
  - How well-structured are your processes? How robust?
  - How well developed is process quality control/improvement?

- **Need a *framework for assessing* your organization!**

CMMI: Capability Maturity Model Integration



Characteristics of the Maturity levels

Level 5 **Optimizing** — Focus on process improvement

Level 4 **Quantitatively Managed** — Processes measured and controlled

Level 3 **Defined** — Processes characterized for the organization and is proactive. (Projects tailor their processes from organization's standards)

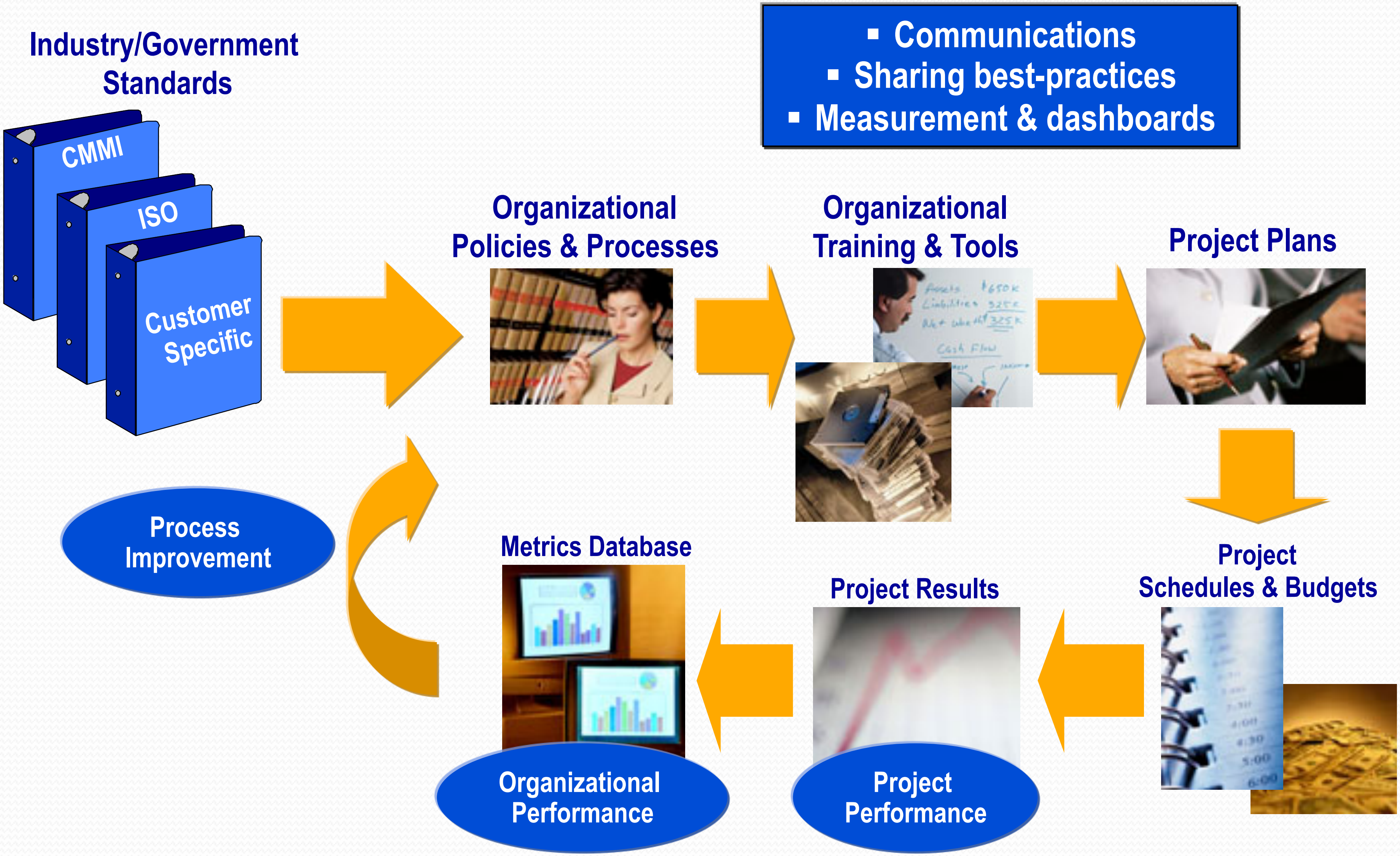
Level 2 **Managed** — Processes characterized for projects and is often reactive.

Level 1 **Initial** — Processes unpredictable, poorly controlled and reactive

http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration

CMMI: A Process Improvement Process

- Communications
- Sharing best-practices
- Measurement & dashboards



Industry/Government Standards

CMMI

ISO

Customer Specific

Organizational Policies & Processes

Organizational Training & Tools

Project Plans

Project Schedules & Budgets

Project Results

Project Performance

Metrics Database

Organizational Performance

Process Improvement

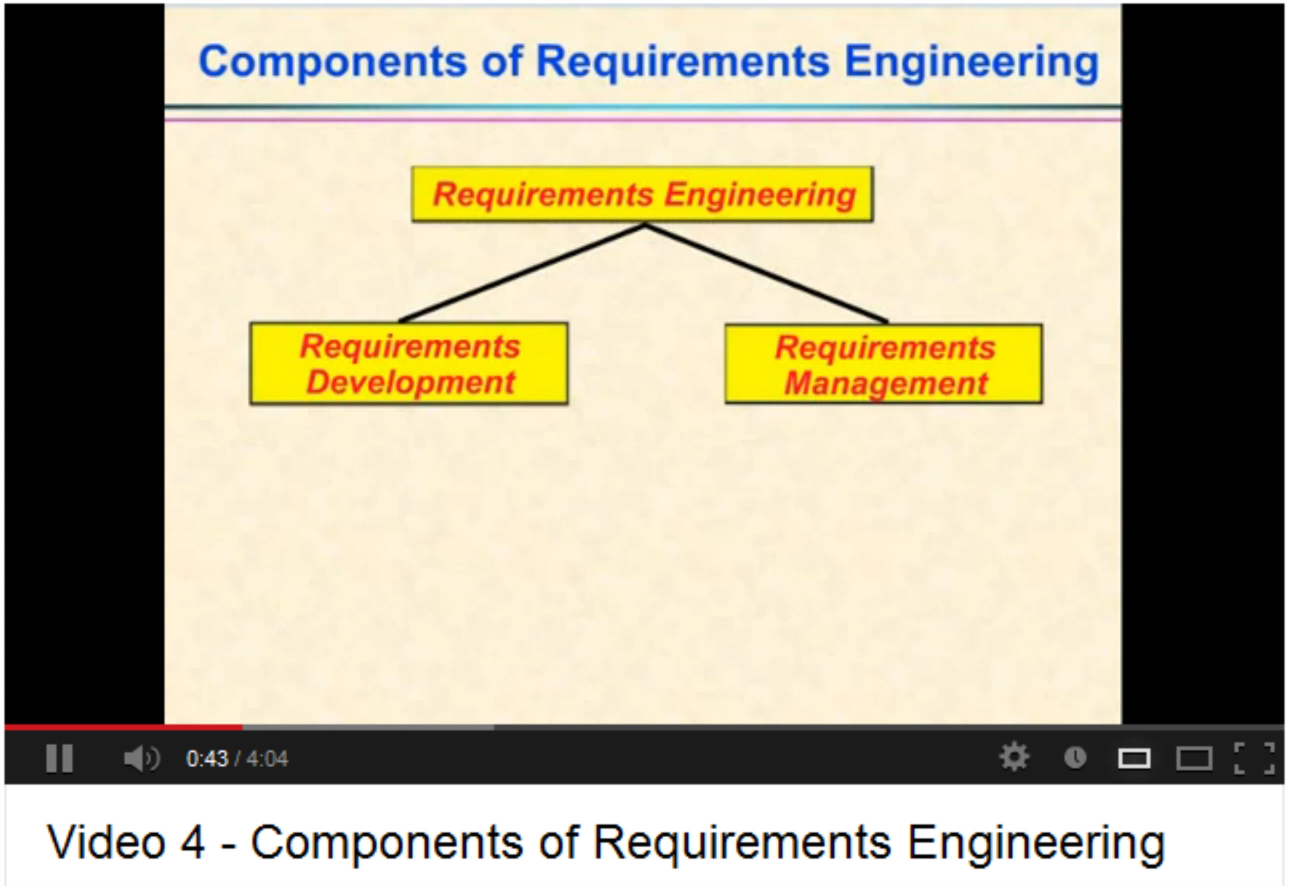
NORTHERN ARIZONA UNIVERSITY

CMMI: Summary

- **CMMI is a tool/framework for evaluating organizational "maturity"**
  - Basically measures how well-run a business is.
  - Score card for CEOs. And for investors.
  - CMMI is just one (widespread) tool. Some companies may use others.
- **Why do we care about this in Capstone?**
  - **Will we use it? No. Immature "companies"**
  - **Will you see it in industry? Yes!**
  - **Will you ever lead a major company?**
- **Software requirements are the major "process" to be optimized for a software business.**

Overview: Requirements Engineering

Video:



Components of Requirements Engineering

Requirements Engineering

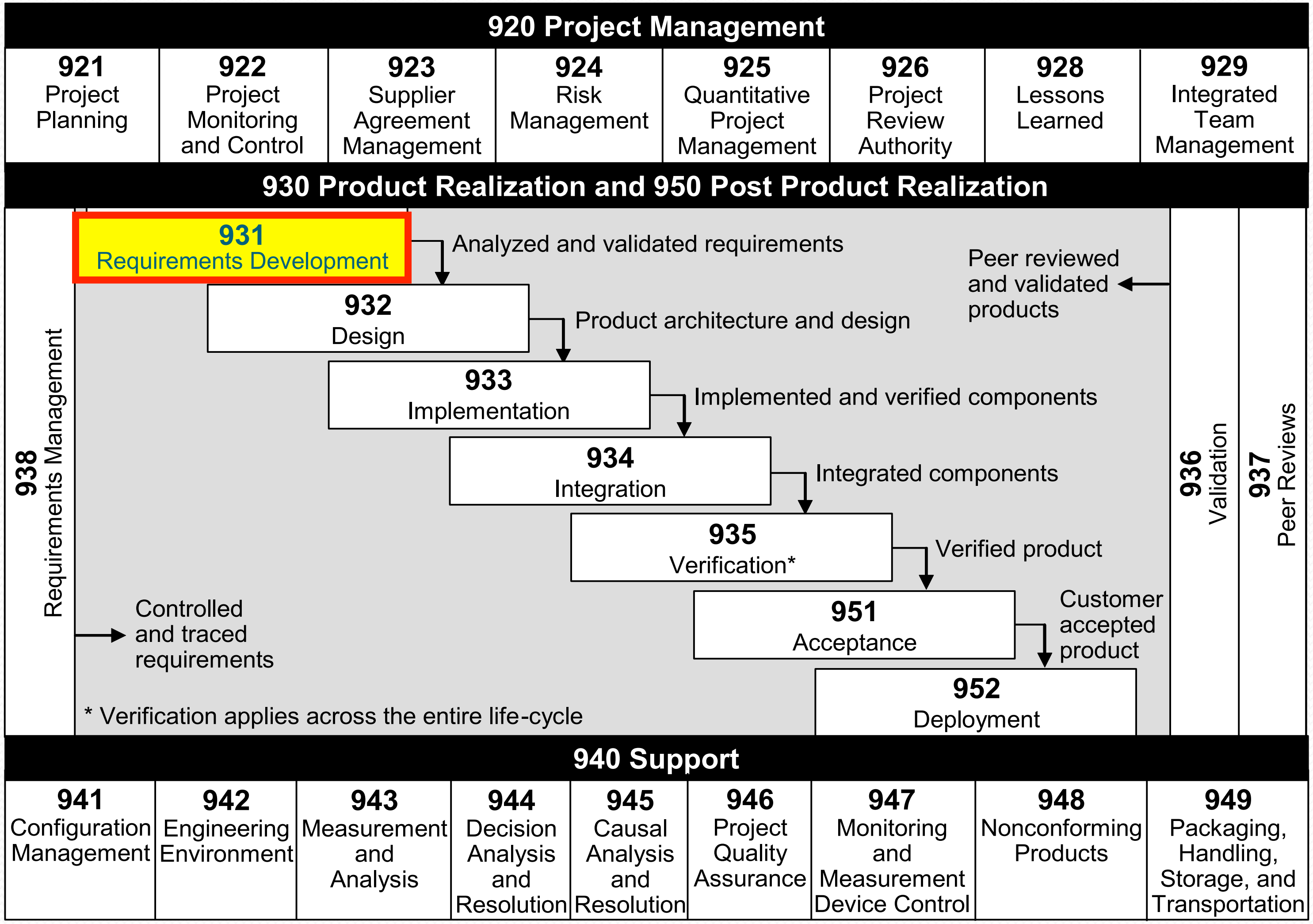
Requirements Development

Requirements Management

Video 4 - Components of Requirements Engineering

http://www.youtube.com/watch?v=a0BSre06ENg

Requirements Development in the Corporate Life-Cycle



920 Project Management

921	922	923	924	925	926	928	929
Project Planning	Project Monitoring and Control	Supplier Agreement Management	Risk Management	Quantitative Project Management	Project Review Authority	Lessons Learned	Integrated Team Management

930 Product Realization and 950 Post Product Realization

* Verification applies across the entire life-cycle

940 Support

941	942	943	944	945	946	947	948	949
Configuration Management	Engineering Environment	Measurement and Analysis	Decision Analysis and Resolution	Causal Analysis and Resolution	Project Quality Assurance	Monitoring and Measurement Device Control	Nonconforming Products	Packaging, Handling, Storage, and Transportation

Purpose of Requirements Development

- ***Requirements* are:**
  - A spec: agreed-upon definition of the characteristics of an acceptable solution to a customer's need
  - A contract: agreement between the customer and the responsible developer

- **The communicated need from a customer may or may not be sufficient to design or procure a solution**
  - It usually is not → need for requirements development

- **The *requirements development process* is a structured set of activities which translate a customer's need into a set of requirements sufficient to develop or procure a solution**

What is Requirements Development?

- **Requirements development:**
  - **Analyzing of customer/user needs, requirements, and objectives**

  - **In the context of:**
    - The mission or jobs that must be accomplished
    - Required performance of system (need measures of effectiveness!)
    - The environments in which the system will be used
    - Any constraints placed upon the system
    - System lifecycle factors (time to retirement, maintenance budget, etc.)
    - Relationships to other requirements

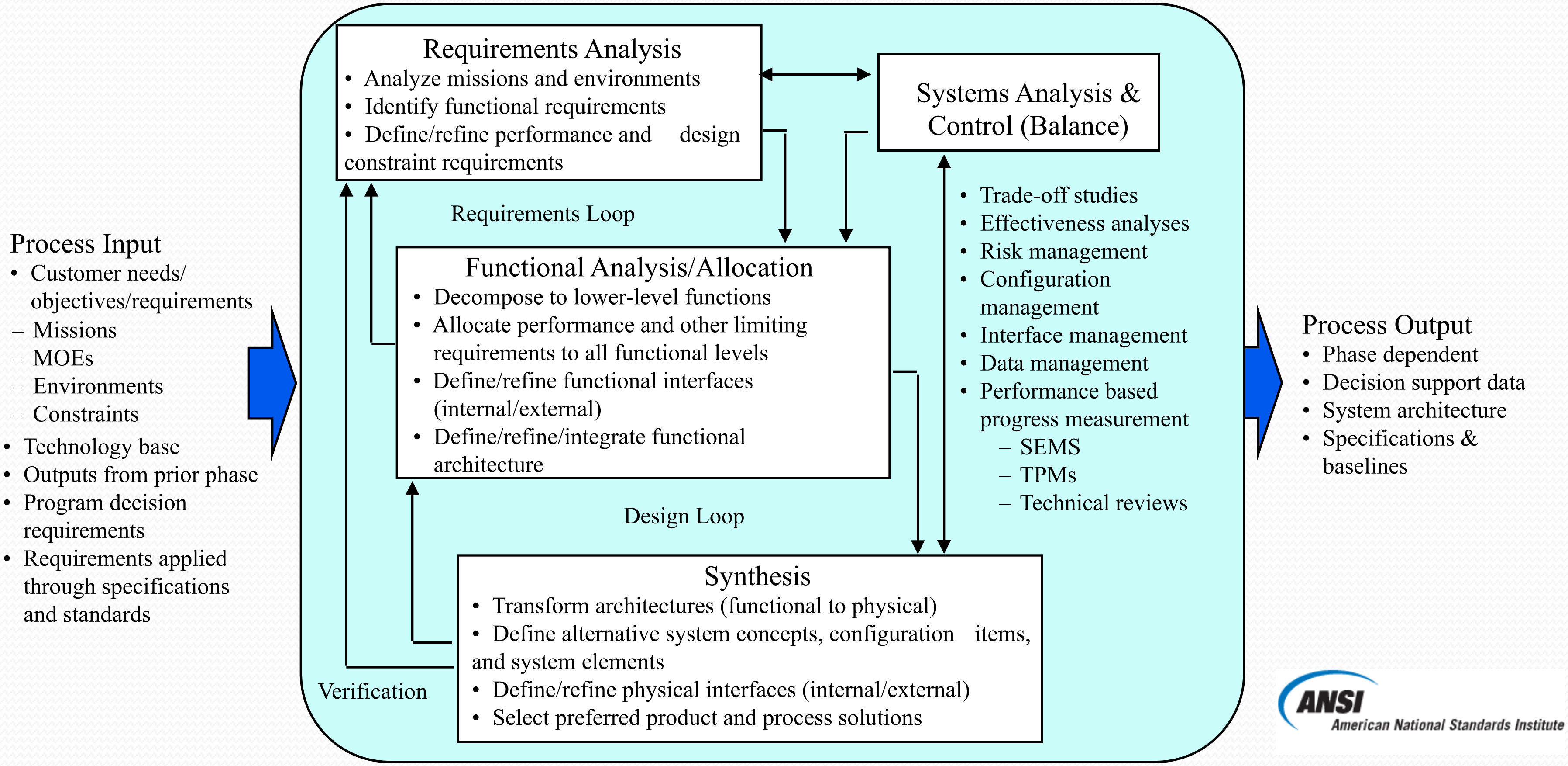
RD as integrated element of software dev. process

- **One can not perform requirements development in a vacuum.**
  - Can't just go on written CFP or other project proposal
  - Can't just be talking with a single perspective (procurement office)

- **It is an *iterative, integrated process* performed in conjunction with the other overlapping processes:**
  - Multiple stakeholder perspectives clarify system priorities and challenges.
  - Evolving requirements support the identification of alternative architectures…which feed back into stakeholder discussions.
  - Decision analysis techniques allow the trade-off of alternative architectures and the selection of the optimum
  - The baseline architecture decisions allow the partition and allocation of system requirements to key elements (modules/factoring)

ANSI/EIA-632 SE Process Model

- **ANSI/EIA-632 is a model which is helpful to clarify the iterative nature of the SE process**



Process Input
- Customer needs/ objectives/requirements
  - Missions
  - MOEs
  - Environments
  - Constraints
- Technology base
- Outputs from prior phase
- Program decision requirements
- Requirements applied through specifications and standards

Requirements Analysis
- Analyze missions and environments
- Identify functional requirements
- Define/refine performance and design constraint requirements

Systems Analysis & Control (Balance)

- Trade-off studies
- Effectiveness analyses
- Risk management
- Configuration management
- Interface management
- Data management
- Performance based progress measurement
  - SEMS
  - TPMs
  - Technical reviews

Requirements Loop

Functional Analysis/Allocation
- Decompose to lower-level functions
- Allocate performance and other limiting requirements to all functional levels
- Define/refine functional interfaces (internal/external)
- Define/refine/integrate functional architecture

Design Loop

Synthesis
- Transform architectures (functional to physical)
- Define alternative system concepts, configuration items, and system elements
- Define/refine physical interfaces (internal/external)
- Select preferred product and process solutions

Verification

Process Output
- Phase dependent
- Decision support data
- System architecture
- Specifications & baselines

http://www.geia.org/ANSI-EIA-632-Standard--PROCESSES-FOR-ENGINEERING-A-SYSTEM

Software Engineering Body of Knowledge (SWEBOK)

The SWEBOK Guide:

- characterizes the contents of the software engineering discipline
- promotes a consistent view of software engineering worldwide
- clarifies software engineering's place with respect to other disciplines
- provides a foundation for training materials and curriculum development, and
- provides a basis for certification and licensing of software engineers.

- **But it's a guideline, not a recipe**
  - Provides an outline of what best practices entail
  - Often referenced as a baseline for justifying specific company processes

http://www.computer.org/portal/web/swebok

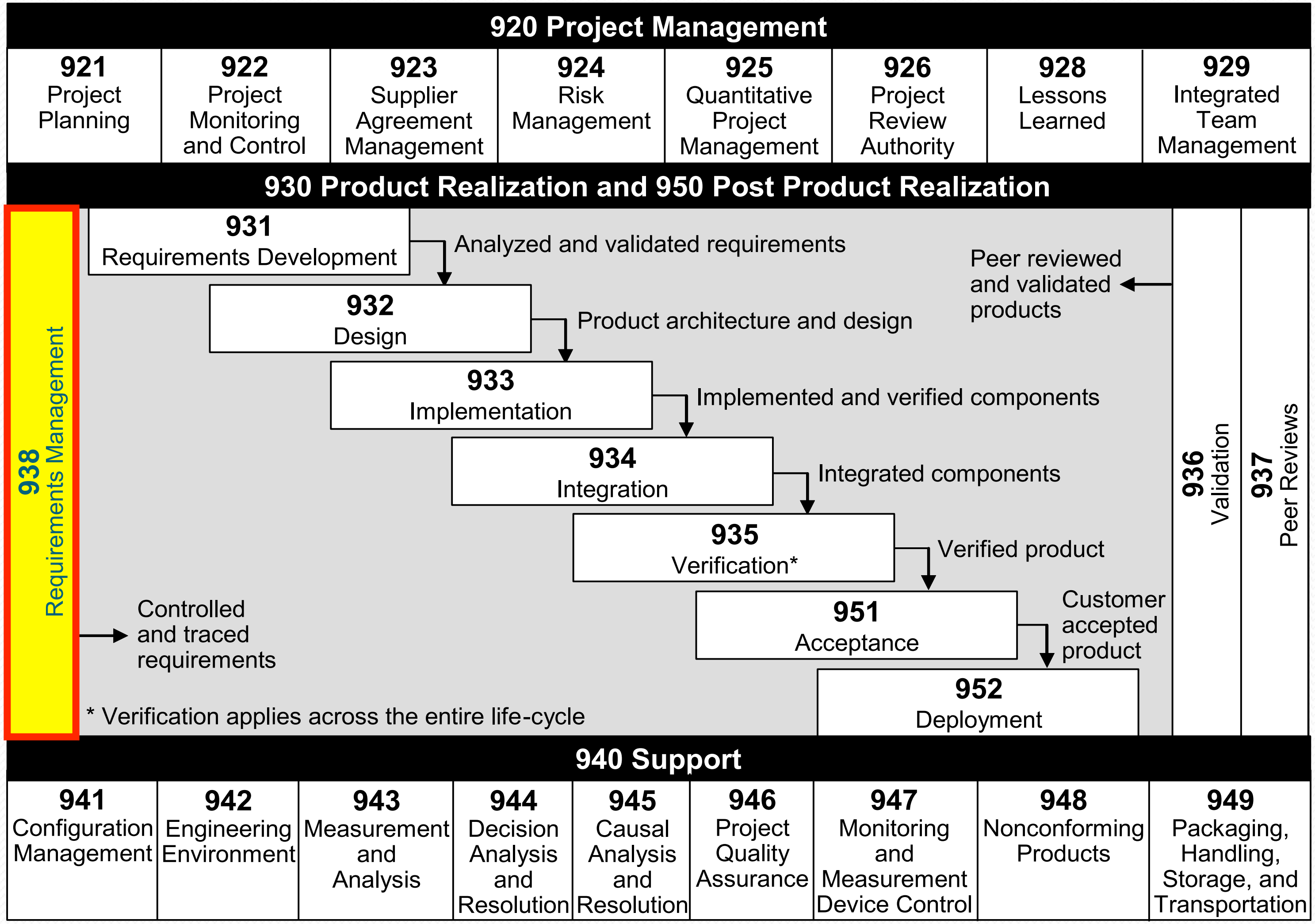
Implications of a Req. Dev. Methodology

- **There are a wide range of requirements development methodologies**
- **Choosing an appropriate methodology is influenced by:**
  - The type of system or subsystem
  - Where you are in the analysis (customer/mission, system, subsystem) or even upgrade to an existing system
  - Customer or organizational directives
  - Experience in one or more methodologies
  - Ability to reuse the analysis and requirements from similar systems
  - Methodology in use above or below you in the analysis
- **Whatever methodology: the basic elements of the process do not change**

Major Requirements Development Methodologies

Technique	Description	Typical Methods
Functional Analysis	Examines the sequential and hierarchical relationship of all functions accomplished by the system. Function oriented: “what” must happen.	- Decomposition - Functional Flows - R-Nets - N-Squared Charts - Operational Analysis
Structured Analysis	Examines the flow of data through the system. Identifies precisely how processes transform input data into output data.	- Data and Control Flow Diagrams - Data Dictionary - Process Specification - Entity Relationship Diagrams
Object Oriented Analysis	Examines the objects in the system and all operations performed against the object and the object’s data	- UML - Coad and Yourdon - Booch - Rumbaugh - Jacobson

Requirements Management in the Life-Cycle



920 Project Management

921	922	923	924	925	926	928	929
Project Planning	Project Monitoring and Control	Supplier Agreement Management	Risk Management	Quantitative Project Management	Project Review Authority	Lessons Learned	Integrated Team Management

930 Product Realization and 950 Post Product Realization

938 Requirements Management

931 Requirements Development

Analyzed and validated requirements

932 Design

Product architecture and design

933 Implementation

Implemented and verified components

934 Integration

Integrated components

935 Verification*

Verified product

951 Acceptance

Customer accepted product

952 Deployment

Controlled and traced requirements

Peer reviewed and validated products

936 Validation

937 Peer Reviews

* Verification applies across the entire life-cycle

940 Support

941	942	943	944	945	946	947	948	949
Configuration Management	Engineering Environment	Measurement and Analysis	Decision Analysis and Resolution	Causal Analysis and Resolution	Project Quality Assurance	Monitoring and Measurement Device Control	Nonconforming Products	Packaging, Handling, Storage, and Transportation

NORTHERN ARIZONA UNIVERSITY

What is Requirements Management (1 of 2)

- **Requirements Management (RM) is a combination of:**
  - Req. Baseline Management
    - Tracking changes to your starting point requirements over time
  - Req. Traceability
    - Being able to trace requirements through development and into the end product. Ensure that you don't lose (or add!) requirements.

- **RM is considered so fundamentally important it is necessary to achieve CMMI Level 2.**

- **Requirements Management occurs throughout a system lifecycle including new requirements and changes to existing requirements**
  - Requirements may be received at contract start
  - New/modified requirements result from Requirements Development
  - Requirement changes may occur throughout the system lifecycle

What is Requirements Management (2 of 2)

- **RM activities facilitate a common understanding of the requirements baseline**
  - Understand what is in the baseline (Version 1.0)
  - Understand what has changed or is being proposed for change
  - Understand the impact of changes
  - Facilitate the management of changes
  - Minimize “requirements creep”
- **Provide the means to assess the state of the requirements**
  - How many
  - What state
  - Relationships

Industry Perspective on Importance of Requirements Management

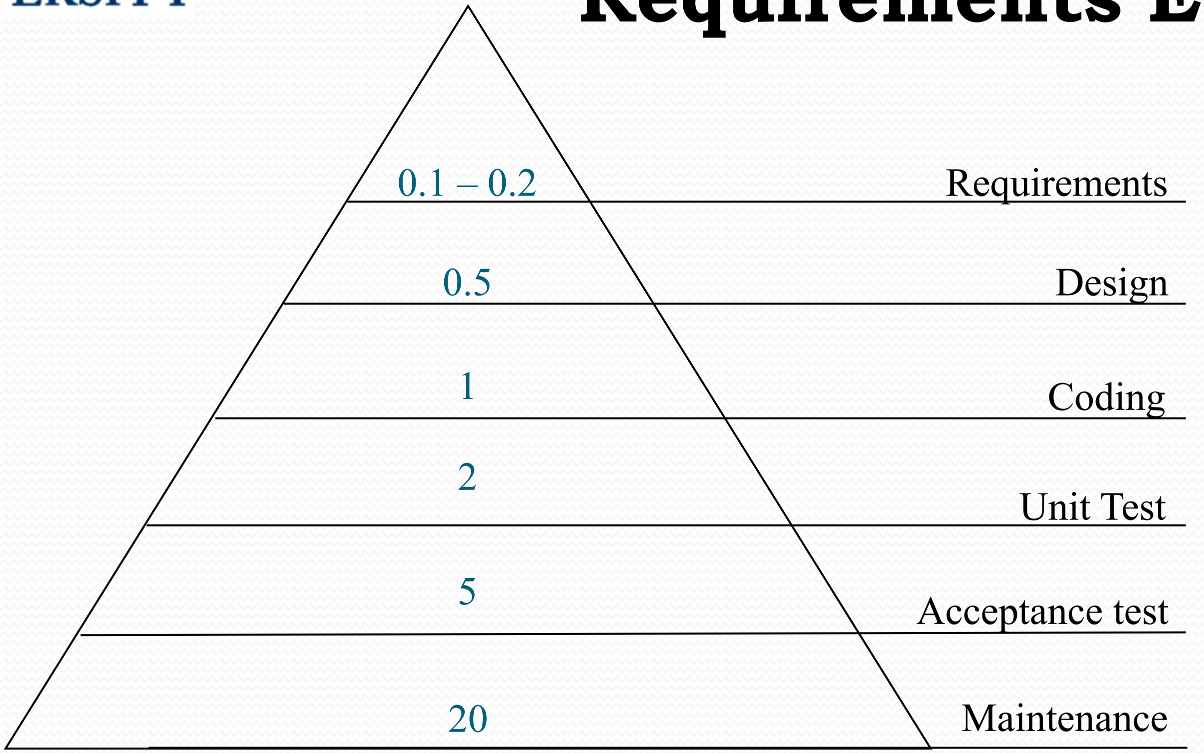
- Research from The Standish Group indicates that requirements are "a primary source of software project risk and software defects."

- Most commonly cited reasons for "challenged projects"
  - Lack of user input: 13% of all projects
  - Incomplete requirements and specifications: 12% of all projects
  - Changing requirements and specifications: 12% of all projects

- Projects should assess requirements risk as part of project start-up and during execution [risk management]
  - Requirements Risk: How fluid/changeable are the requirements likely to be over time? (Based on product type, market changes, clients, etc.)

Source: Leffingwell and Widrig, Managing software Requirements, Addison-Wesley, 2003

The High Cost of Requirements Errors



0.1 – 0.2	Requirements
0.5	Design
1	Coding
2	Unit Test
5	Acceptance test
20	Maintenance

Relative Cost to Repair a Defect by Lifecycle Phase

200:1 cost savings by detecting defects in the requirements stage vs. in the maintenance stage!

Source: Davis, Software Requirements: Objects, Functions and States, Prentice-Hall, 1993

Implementing Requirements Management

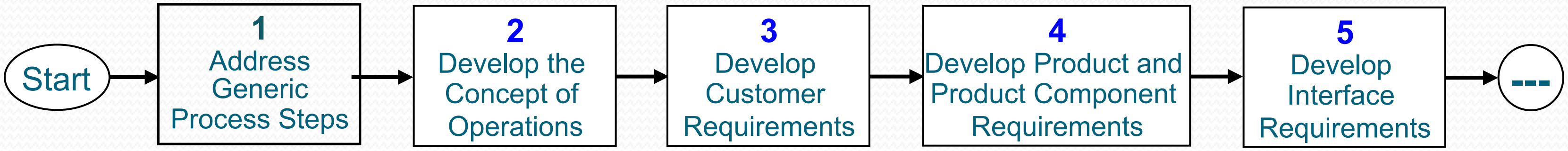
- **Establish a basis for effective Req. Mgmt.**
  - Develop an understanding with requirements providers and relevant stakeholders on the requirements
  - Obtain commitment to requirements

- **Document, manage, communicate, and control requirements changes**
  - Establish and maintain bi-directional traceability
    - Requirements ←→ lower- level requirements
    - Requirements ←→ design, implementation, and test
  - Identify and resolve inconsistencies between requirements, plans, and products

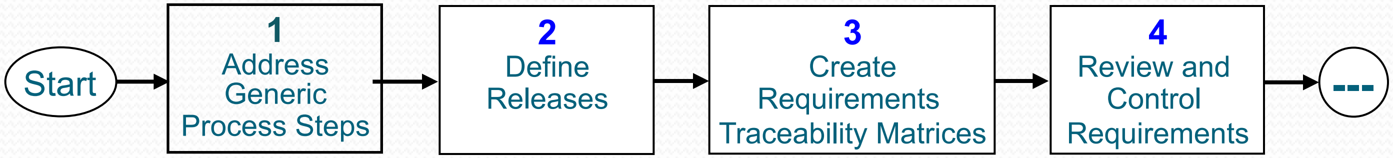
NORTHERN ARIZONA UNIVERSITY

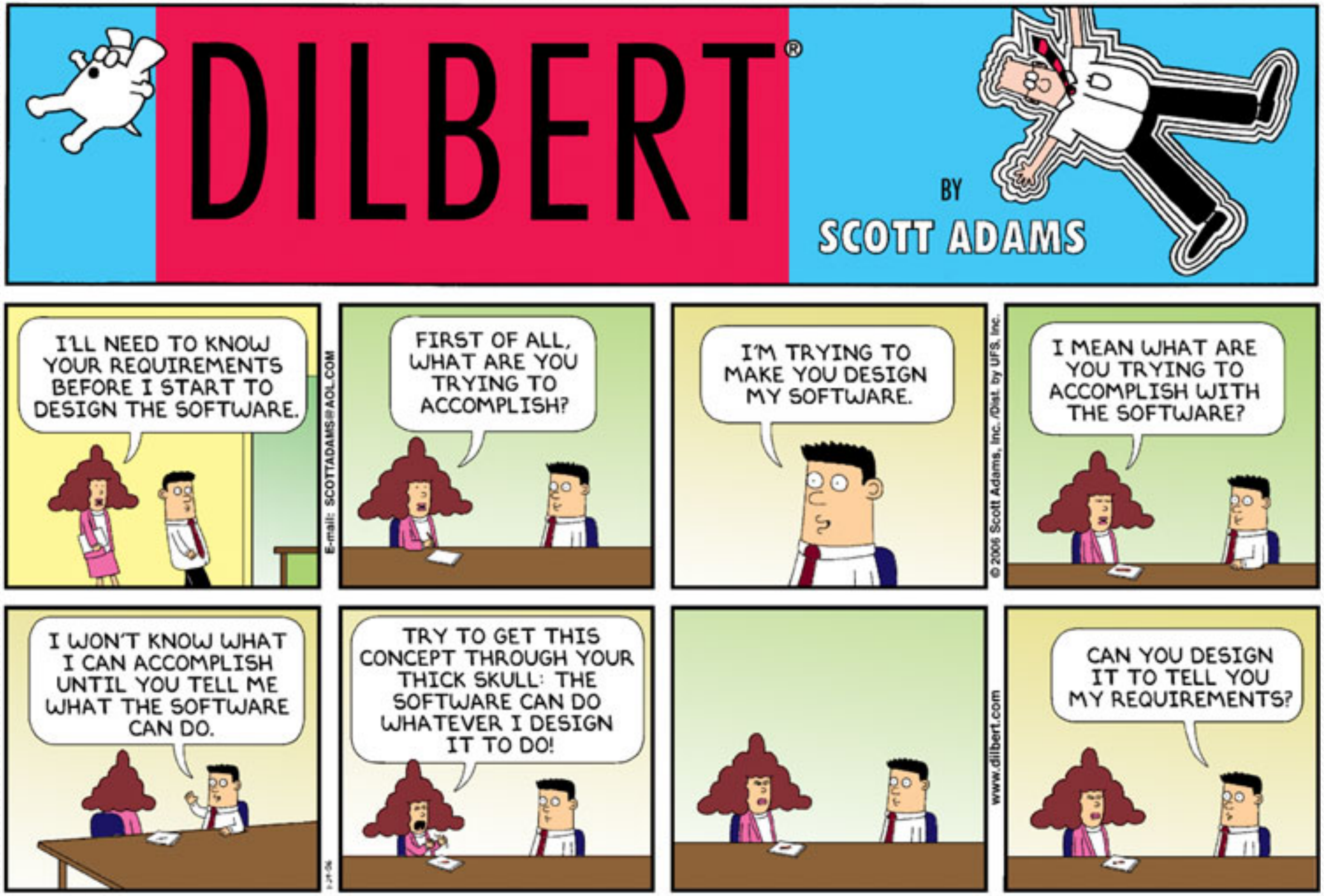
Process Descriptions
in the CMMI

- **Process overview for Requirements Development**


Start
1
Address Generic Process Steps
2
Develop the Concept of Operations
3
Develop Customer Requirements
4
Develop Product and Product Component Requirements
5
Develop Interface Requirements

- **Process overview for Requirements Management**


Start
1
Address Generic Process Steps
2
Define Releases
3
Create Requirements Traceability Matrices
4
Review and Control Requirements


DILBERT®
BY SCOTT ADAMS
I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.
E-mail: SCOTTADAMS@AOL.COM
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SOFTWARE.
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SOFTWARE?
I WON'T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SOFTWARE CAN DO.
TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SOFTWARE CAN DO WHATEVER I DESIGN IT TO DO!
www.dilbert.com
CAN YOU DESIGN IT TO TELL YOU MY REQUIREMENTS?

© Scott Adams, Inc./Dist. by UFS, Inc.