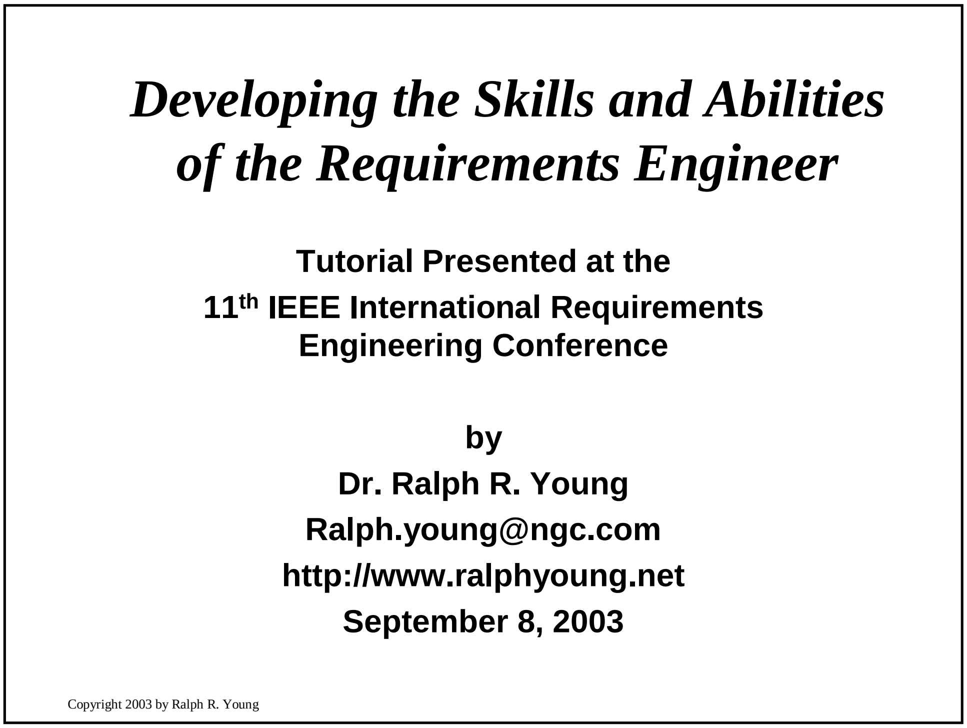


Developing the Skills and Abilities of the Requirements Engineer

Tutorial Presented at the

11th IEEE International Requirements Engineering Conference

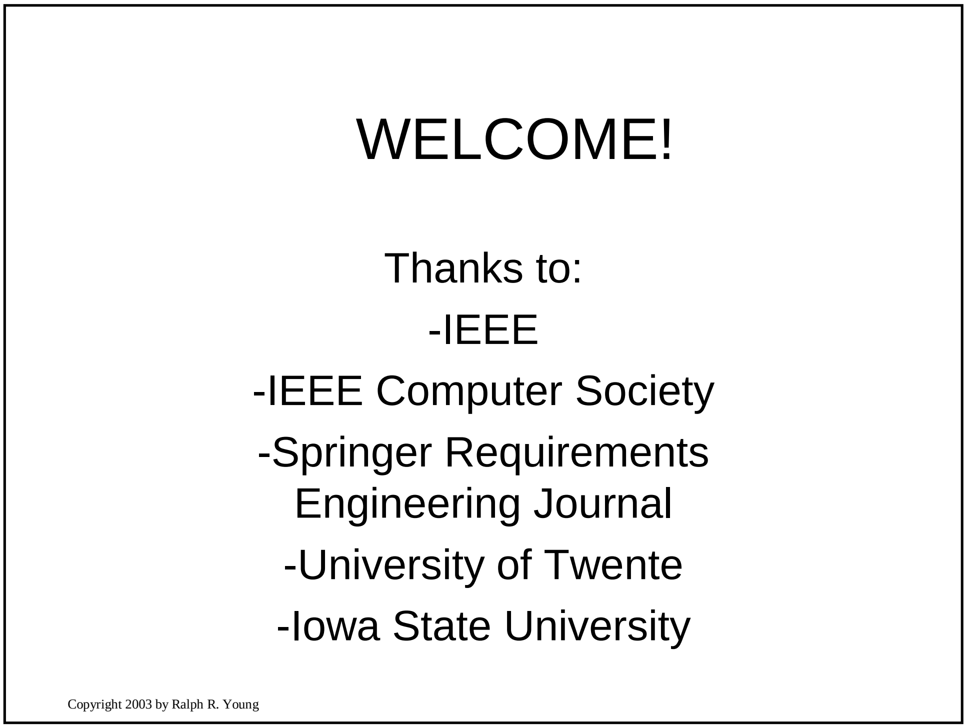
by

Dr. Ralph R. Young

Ralph.young@ngc.com

http://www.ralphyoung.net

September 8, 2003



WELCOME!

Thanks to:

-IEEE

-IEEE Computer Society

-Springer Requirements Engineering Journal

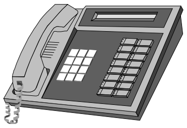
-University of Twente

-Iowa State University

Housekeeping

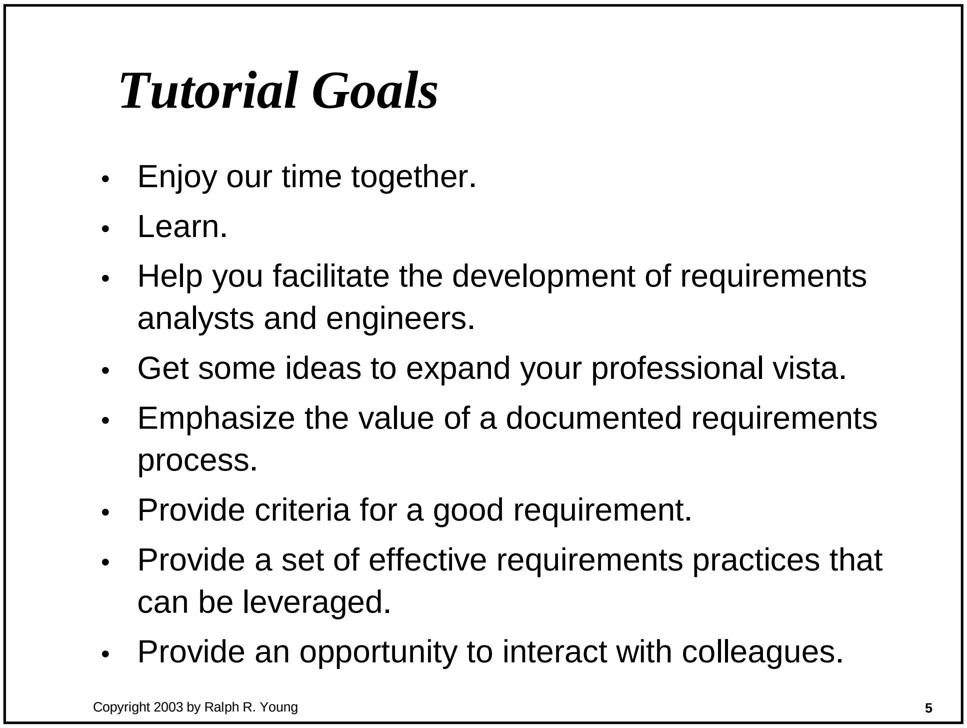
- Restrooms
- Please avoid distractions
- Time guidelines/hours of the tutorial
- Breaks
- Cell phones
- Jump in and participate

Please keep your Tutorial Evaluation handy and use it throughout the morning to provide feedback, make comments, and contribute suggestions and ideas. Practice “continuous improvement.” Think about and suggest tutorial topics that you’d like.



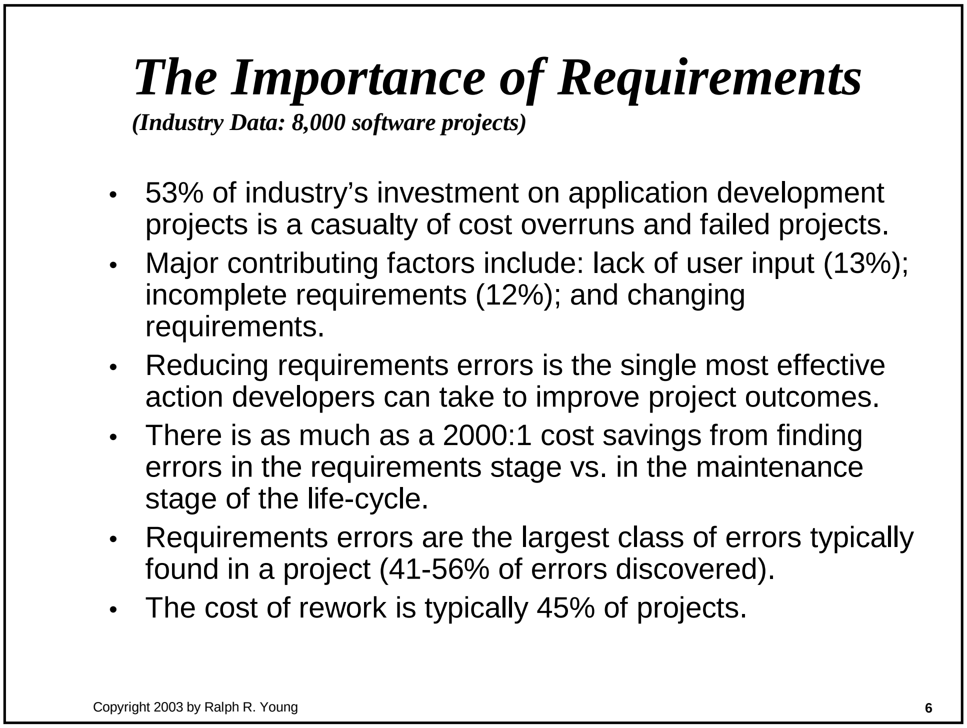
Tutorial Outline

- Welcome
- Tutorial Goals
- The Importance of Requirements
- The Roles of the Requirements Analyst
- Break
- Skills and Characteristics of an Effective Requirements Engineer
- Exercise: Improving Requirements Engineering
- Break
- Best Practices for Requirements Engineers
- Requirements Analysts’ Specialty Skills
- An Integrated Quality Approach
- Leveraging Effective Requirements Practices
- On Prioritizing Requirements
- The Bottom Line
- Back-up Slides



Tutorial Goals

- Enjoy our time together.
- Learn.
- Help you facilitate the development of requirements analysts and engineers.
- Get some ideas to expand your professional vista.
- Emphasize the value of a documented requirements process.
- Provide criteria for a good requirement.
- Provide a set of effective requirements practices that can be leveraged.
- Provide an opportunity to interact with colleagues.



The Importance of Requirements

(Industry Data: 8,000 software projects)

- 53% of industry's investment on application development projects is a casualty of cost overruns and failed projects.
- Major contributing factors include: lack of user input (13%); incomplete requirements (12%); and changing requirements.
- Reducing requirements errors is the single most effective action developers can take to improve project outcomes.
- There is as much as a 2000:1 cost savings from finding errors in the requirements stage vs. in the maintenance stage of the life-cycle.
- Requirements errors are the largest class of errors typically found in a project (41-56% of errors discovered).
- The cost of rework is typically 45% of projects.

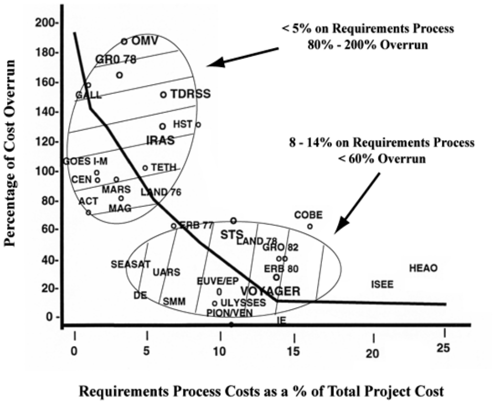
What Do We Mean By “Real” Requirements?

- Industry experience is that the customer’s and user’s *stated* requirements are <u>never</u> the *real* requirements!
  - This accounts for many of our problems.
  - A *joint effort* of the customer and system supplier is required to identify the real requirements.
  - The system supplier needs people who are domain/subject matter experts.
- Use established criteria for “a good requirement.”
- Then, *emerge* real requirements using the “joint team.”
- Document the <u>rationale</u> for each requirement.
- Maintain information about each requirement in your requirements tool (source, history, priority, other attributes).

What’s Needed to Enable a Successful Requirements Approach?

- An organizational requirements <u>policy</u>
- Senior management support-”<u>sponsorship</u>”
- A requirements <u>process</u>
  - Designed by performers in the organizations who are concerned with requirements
    - <u>An organizational “Requirements Working Group”</u>
- A requirements <u>process description</u> (narrative)
- <u>Investment in the requirements process</u> of 8-14% of total project costs
- Recommended <u>methods and techniques</u> for each part of the requirements process
- <u>Training</u> for how to address requirements in your organization
- <u>An effective automated requirements tool</u> (essential, not optional)
- A few useful <u>metrics</u> that are <u>tracked and used</u>
- Suggested <u>reading</u> (for more information, advice, suggestions, recommendations, lessons-learned from others)
- <u>Effective communication</u>
- <u>A way (that is, a mechanism) to control changes to requirements</u>

Effect of Requirements Process Investment on Program Costs



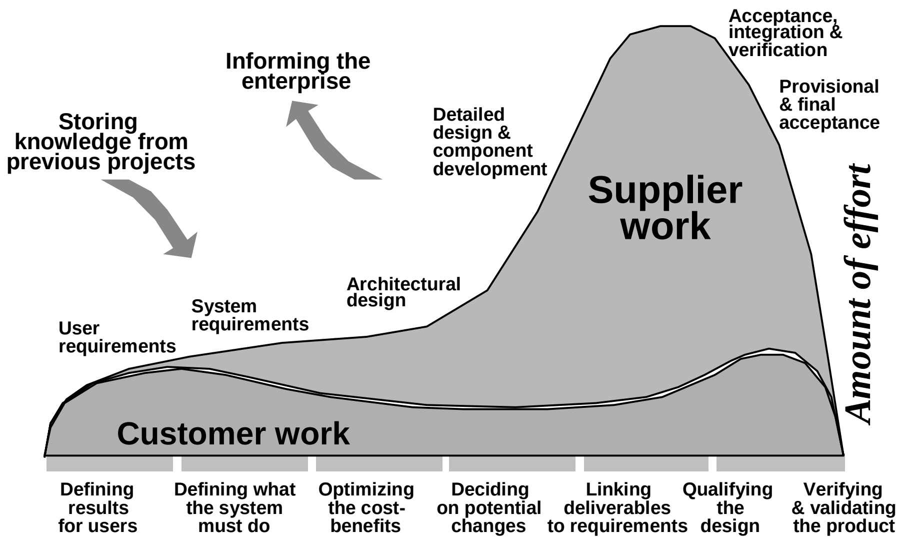
Nationally known requirements expert Ivy Hooks has collected data from actual projects which show that the probability of project success (as measured by meeting its Target Cost) is greatest when 8-14% of the Total Project Cost is invested in requirements activities. Source: Compliance Automation, Inc.

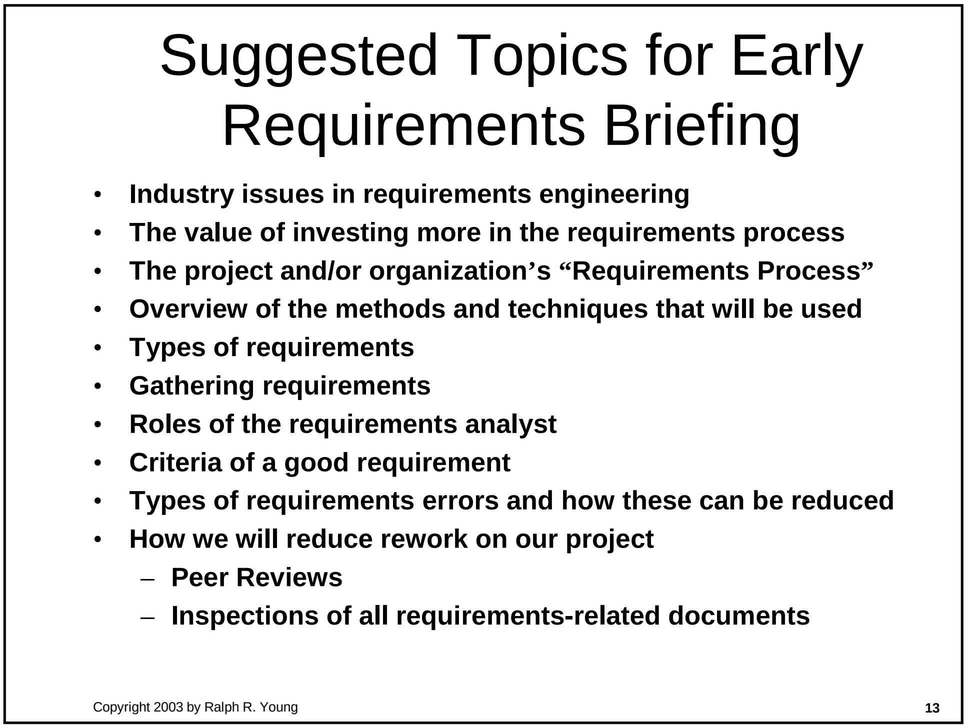
The Roles of the Requirements Analyst

1. Work collaboratively with customers and users to emerge the *real* requirements for a planned system or software development effort.
2. Work effectively with customers and users to manage new and changed requirements so that the project stays in control.
3. Be alert to new technologies that may help.
4. Facilitate the project reusing artifacts and achieving repeatability.
5. Assist the project and its customers in envisioning a growth path from the first release or version of products through a set of staged releases to the "ultimate system or products."
6. Advise the project (and customer) of methods, techniques, and automated tools that are available to best support requirements-related project work and activities.
7. Use metrics to measure, track, and control requirements-related project work activities and results.
8. Be able to facilitate discussions and to mediate conflicts.
9. Study the domain of the area in which the system or software is being used.

Please enjoy a break,
meet someone you don’t know,
and return to your seat
by

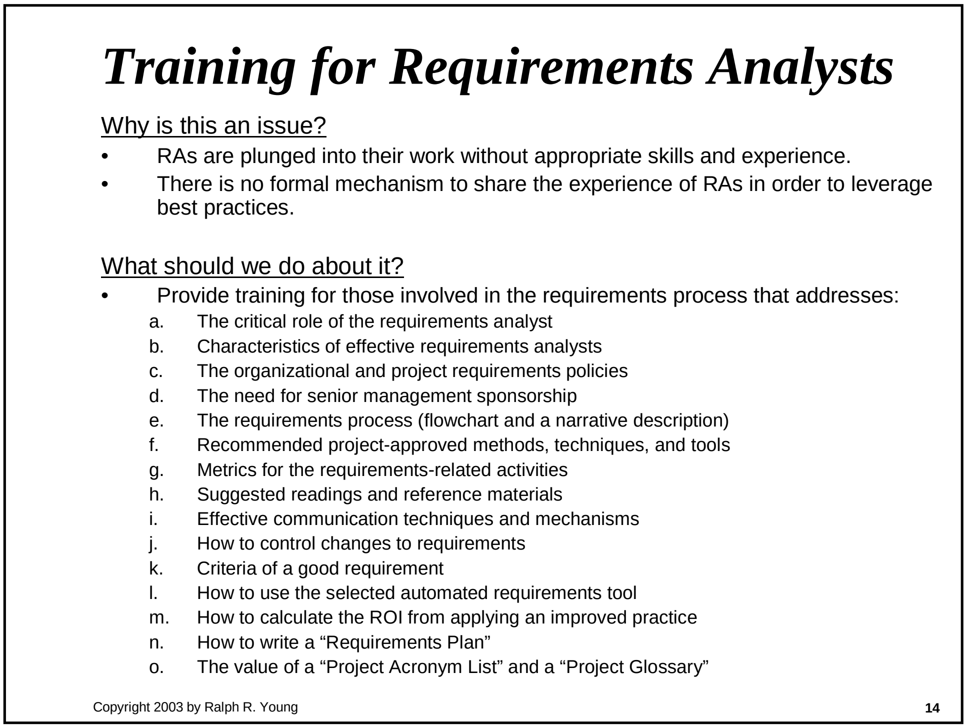
Involve Your Customers!


Storing knowledge from previous projects
Informing the enterprise
Detailed design & component development
Acceptance, integration & verification
Provisional & final acceptance
Supplier work
Amount of effort
Architectural design
System requirements
User requirements
Customer work
Defining results for users
Defining what the system must do
Optimizing the cost-benefits
Deciding on potential changes
Linking deliverables to requirements
Qualifying the design
Verifying & validating the product



Suggested Topics for Early Requirements Briefing

- **Industry issues in requirements engineering**
- **The value of investing more in the requirements process**
- **The project and/or organization's "Requirements Process"**
- **Overview of the methods and techniques that will be used**
- **Types of requirements**
- **Gathering requirements**
- **Roles of the requirements analyst**
- **Criteria of a good requirement**
- **Types of requirements errors and how these can be reduced**
- **How we will reduce rework on our project**
  - **Peer Reviews**
  - **Inspections of all requirements-related documents**



Training for Requirements Analysts

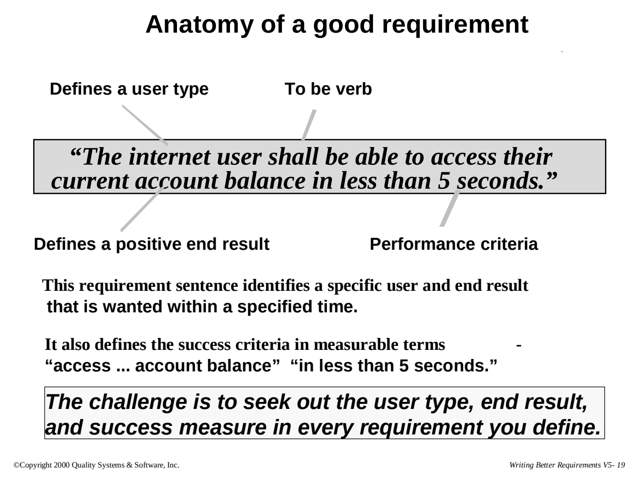
<u>Why is this an issue?</u>

- RAs are plunged into their work without appropriate skills and experience.
- There is no formal mechanism to share the experience of RAs in order to leverage best practices.

<u>What should we do about it?</u>

- Provide training for those involved in the requirements process that addresses:
  a. The critical role of the requirements analyst
  b. Characteristics of effective requirements analysts
  c. The organizational and project requirements policies
  d. The need for senior management sponsorship
  e. The requirements process (flowchart and a narrative description)
  f. Recommended project-approved methods, techniques, and tools
  g. Metrics for the requirements-related activities
  h. Suggested readings and reference materials
  i. Effective communication techniques and mechanisms
  j. How to control changes to requirements
  k. Criteria of a good requirement
  l. How to use the selected automated requirements tool
  m. How to calculate the ROI from applying an improved practice
  n. How to write a "Requirements Plan"
  o. The value of a "Project Acronym List" and a "Project Glossary"

Anatomy of a Good Requirement



Anatomy of a good requirement

"The internet user shall be able to access their current account balance in less than 5 seconds."

This requirement sentence identifies a specific user and end result that is wanted within a specified time.

It also defines the success criteria in measurable terms -
"access ... account balance" "in less than 5 seconds."

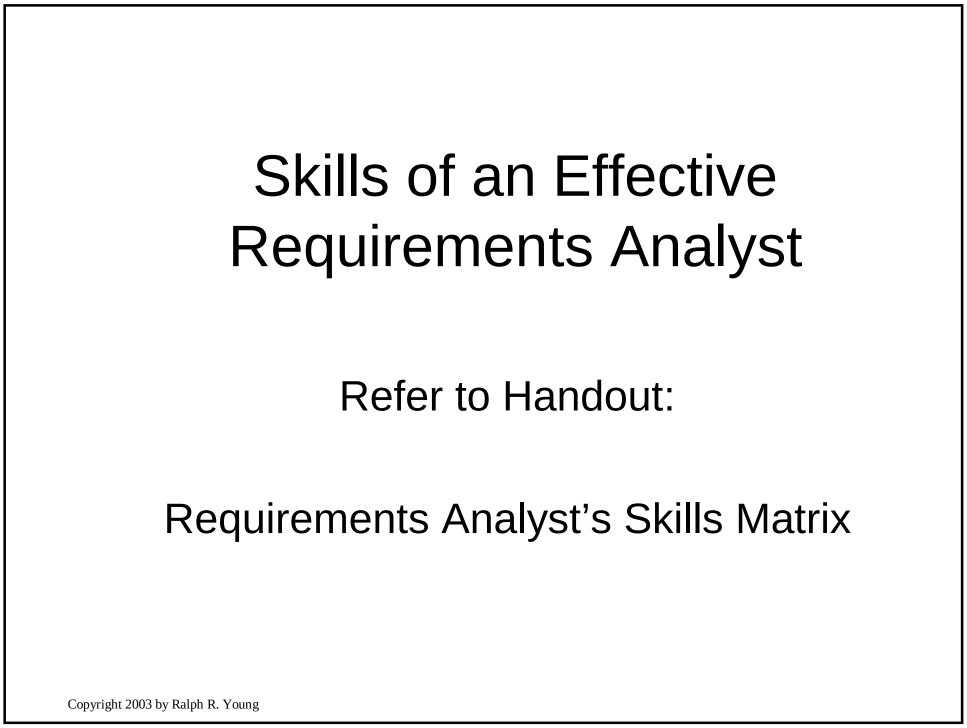
The challenge is to seek out the user type, end result, and success measure in every requirement you define.

©Copyright 2000 Quality Systems & Software, Inc. *Writing Better Requirements V5- 19*

The Criteria of a Good Requirement

Each individual requirement should be:

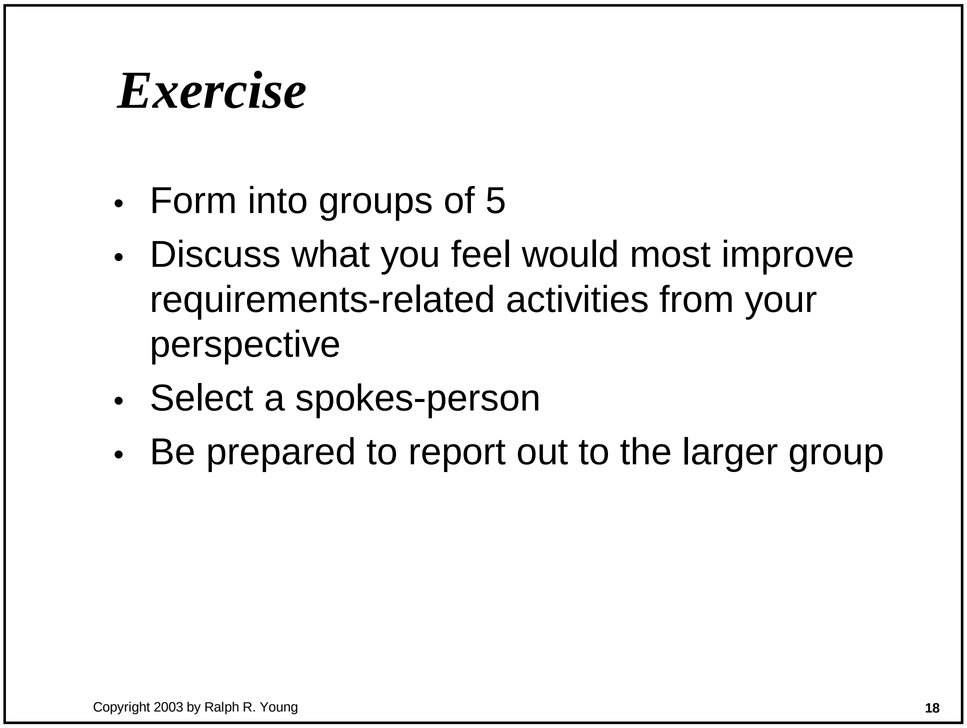
- Necessary -- If the system can meet prioritized real needs without the requirement, it isn't necessary
- Feasible -- The requirement is doable and can be accomplished within cost and schedule
- Correct -- The facts related to the requirement are accurate and it is technically and legally possible
- Concise – The requirement is stated simply
- Unambiguous – The requirement can be interpreted in only one way
- Complete -- All conditions under which the requirement applies are stated and it expresses a whole idea or statement
- Consistent -- Not in conflict with other requirements
- Verifiable -- Implementation of the requirement in the system can be proved
- Traceable -- Can trace to the source of the requirement and it can be tracked throughout the system (e.g., to the design, code, test, and documentation)
- Allocated --The requirement is assigned to a component of the designed system
- Design independent -- Does not pose a specific implementation solution
- Non-redundant -- Not a duplicate requirement
- Standard construct – The requirement is stated as an imperative using "shall"
- Unique identifier – Each requirement shall have a unique identifying number
- Devoid of escape clauses such as "if, when, but, except, unless, and although" and not speculative or general (i.e., avoid wording such as "usually, generally, often, normally, and typically"



Skills of an Effective Requirements Analyst

Refer to Handout:

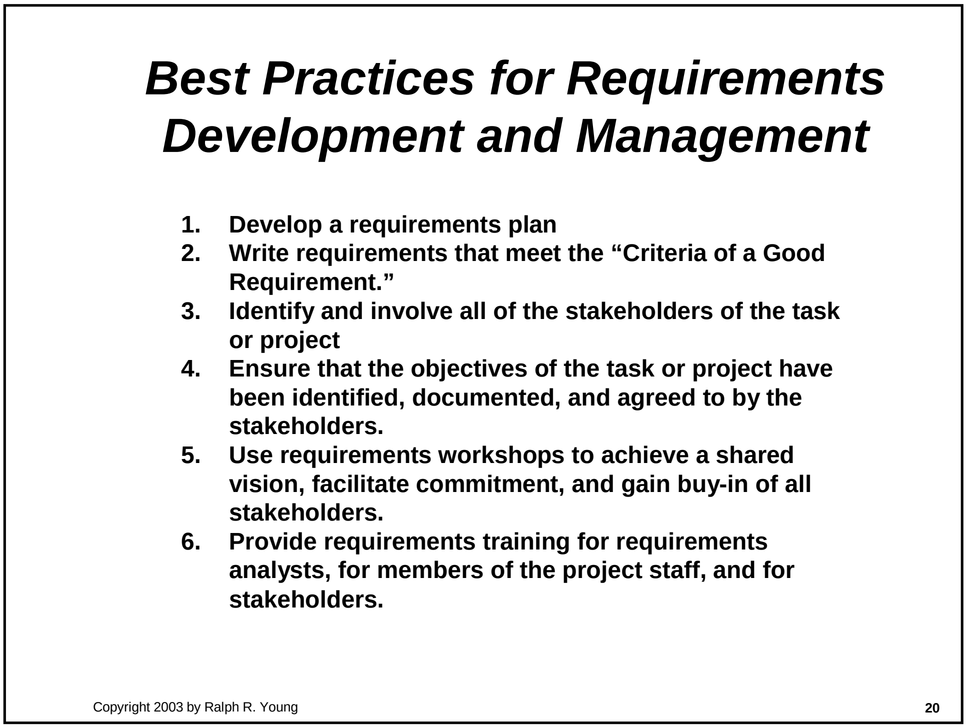
Requirements Analyst’s Skills Matrix



Exercise

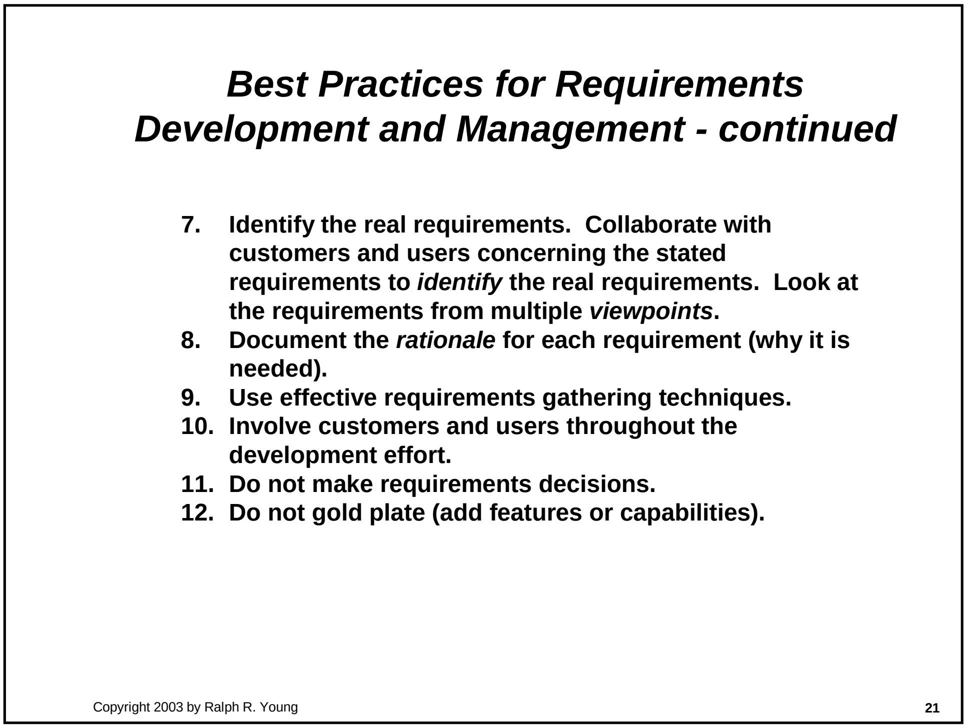
- Form into groups of 5
- Discuss what you feel would most improve requirements-related activities from your perspective
- Select a spokes-person
- Be prepared to report out to the larger group

Please enjoy a break,
meet someone you don’t know,
and return to your seat
by



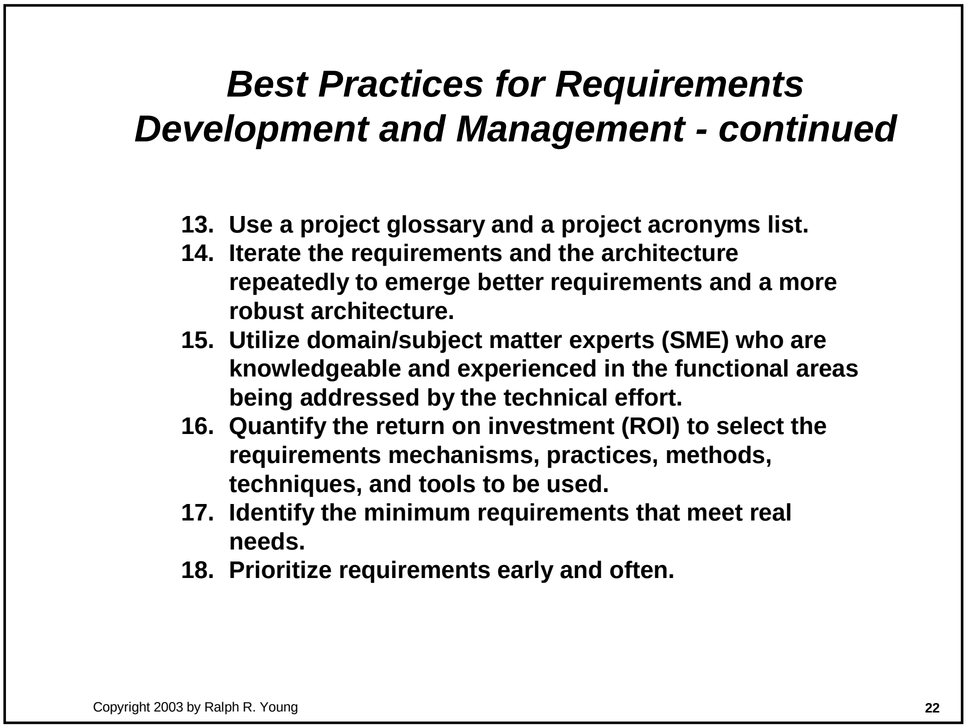
Best Practices for Requirements Development and Management

1. Develop a requirements plan
2. Write requirements that meet the “Criteria of a Good Requirement.”
3. Identify and involve all of the stakeholders of the task or project
4. Ensure that the objectives of the task or project have been identified, documented, and agreed to by the stakeholders.
5. Use requirements workshops to achieve a shared vision, facilitate commitment, and gain buy-in of all stakeholders.
6. Provide requirements training for requirements analysts, for members of the project staff, and for stakeholders.



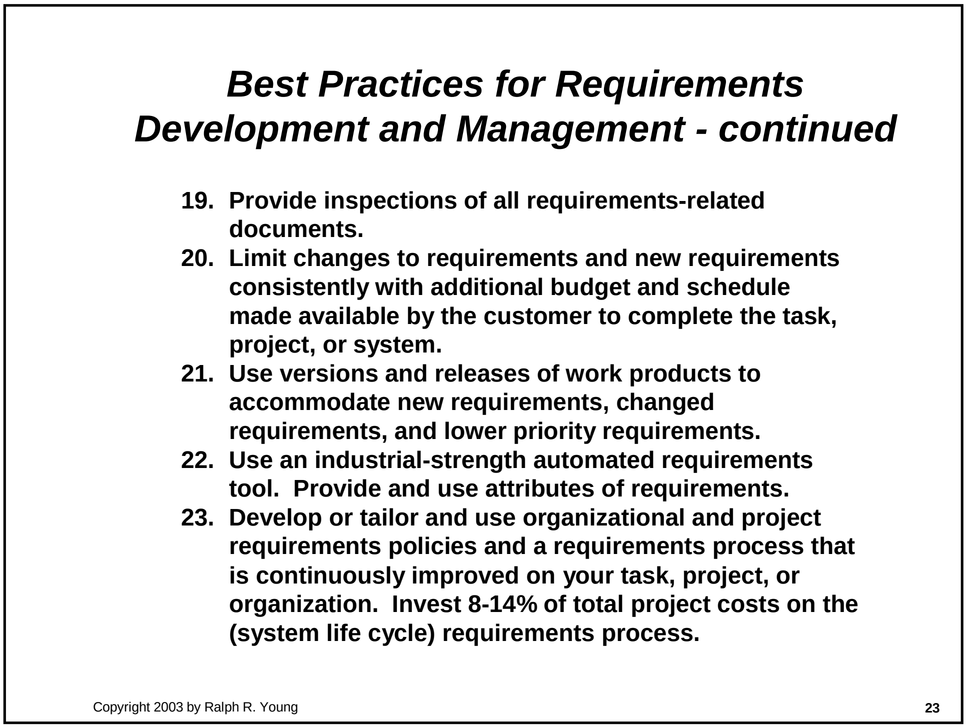
Best Practices for Requirements Development and Management - continued

7. Identify the real requirements. Collaborate with customers and users concerning the stated requirements to *identify* the real requirements. Look at the requirements from multiple *viewpoints*.
8. Document the *rationale* for each requirement (why it is needed).
9. Use effective requirements gathering techniques.
10. Involve customers and users throughout the development effort.
11. Do not make requirements decisions.
12. Do not gold plate (add features or capabilities).



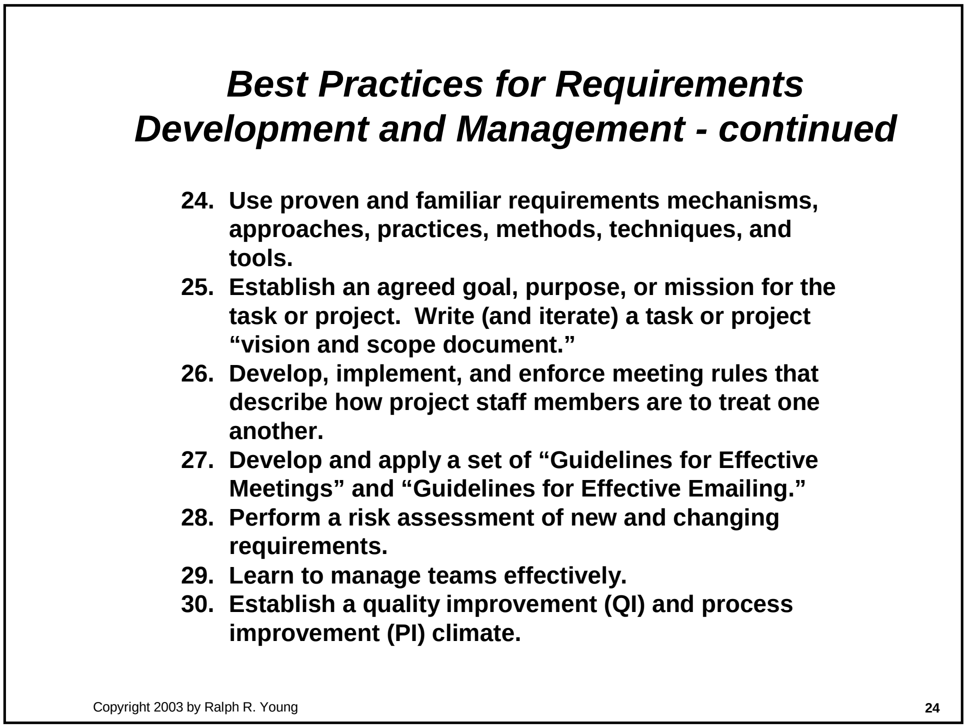
Best Practices for Requirements Development and Management - continued

13. Use a project glossary and a project acronyms list.
14. Iterate the requirements and the architecture repeatedly to emerge better requirements and a more robust architecture.
15. Utilize domain/subject matter experts (SME) who are knowledgeable and experienced in the functional areas being addressed by the technical effort.
16. Quantify the return on investment (ROI) to select the requirements mechanisms, practices, methods, techniques, and tools to be used.
17. Identify the minimum requirements that meet real needs.
18. Prioritize requirements early and often.



Best Practices for Requirements Development and Management - continued

19. Provide inspections of all requirements-related documents.
20. Limit changes to requirements and new requirements consistently with additional budget and schedule made available by the customer to complete the task, project, or system.
21. Use versions and releases of work products to accommodate new requirements, changed requirements, and lower priority requirements.
22. Use an industrial-strength automated requirements tool. Provide and use attributes of requirements.
23. Develop or tailor and use organizational and project requirements policies and a requirements process that is continuously improved on your task, project, or organization. Invest 8-14% of total project costs on the (system life cycle) requirements process.



Best Practices for Requirements Development and Management - continued

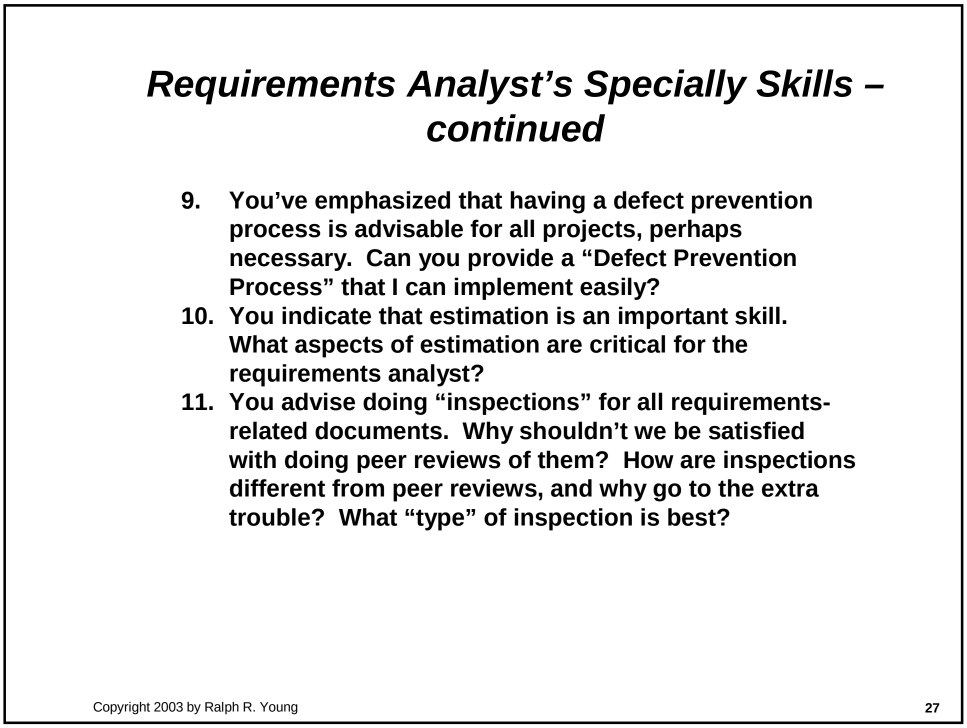
24. Use proven and familiar requirements mechanisms, approaches, practices, methods, techniques, and tools.
25. Establish an agreed goal, purpose, or mission for the task or project. Write (and iterate) a task or project “vision and scope document.”
26. Develop, implement, and enforce meeting rules that describe how project staff members are to treat one another.
27. Develop and apply a set of “Guidelines for Effective Meetings” and “Guidelines for Effective Emailing.”
28. Perform a risk assessment of new and changing requirements.
29. Learn to manage teams effectively.
30. Establish a quality improvement (QI) and process improvement (PI) climate.

Requirements Analyst’s Specially Skills

1. Why are requirements errors so devastating and how can requirements analysts help address the problem of requirements errors?
2. What does the requirements analyst need to know about configuration management?
3. What does the requirements analyst need to know about UML?
4. What if I’m supporting a “small” project? Does any of this stuff still apply? How can I convince the PM and my co-workers to incorporate a degree of discipline and process into our approach?

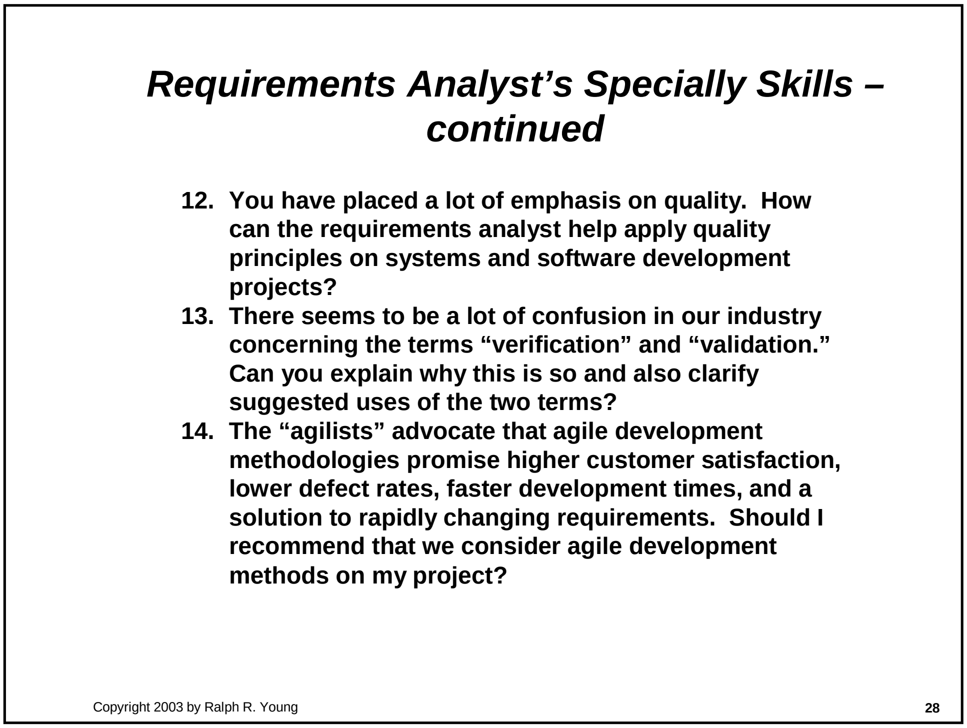
Requirements Analyst’s Specially Skills – continued

5. What is the difference between a “requirements specification” and “specifying the requirements?”
6. I notice “Impact Estimation” on the Requirements Skills Matrix – what is it, and how can I learn more about it?
7. You seem to suggest that the requirements analyst should be a leader on the project. Why do I need to be a leader? How can I be a leader? What should I lead?
8. You have stressed the role of the requirements analyst in facilitating discussions, presentations, meetings, training sessions, and workshops. What can I do to become a better facilitator?



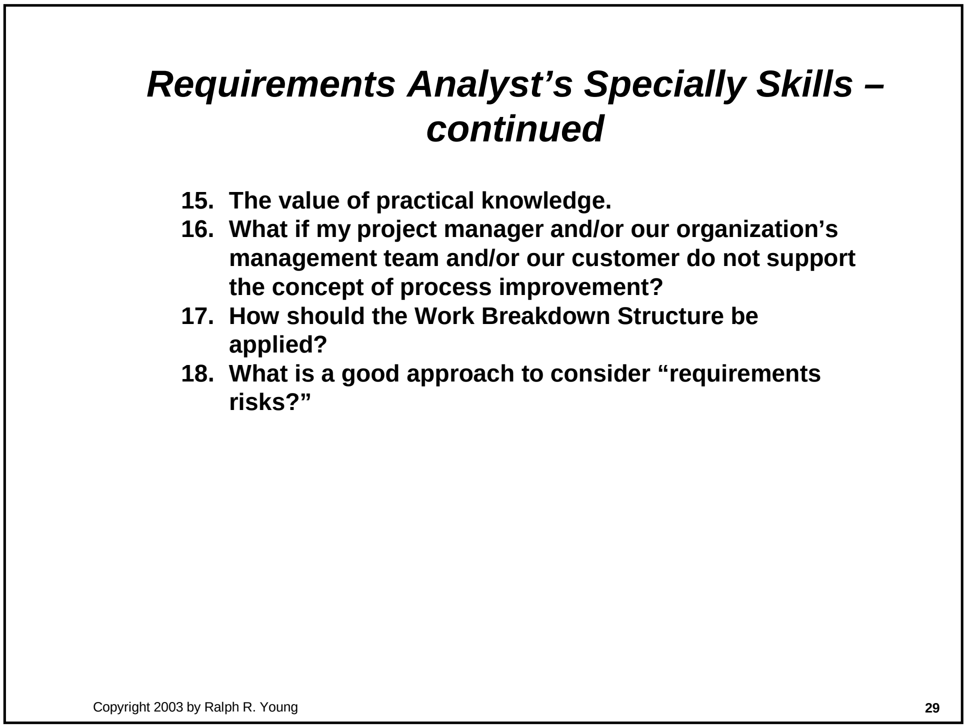
Requirements Analyst's Specially Skills – continued

9. **You've emphasized that having a defect prevention process is advisable for all projects, perhaps necessary. Can you provide a "Defect Prevention Process" that I can implement easily?**
10. **You indicate that estimation is an important skill. What aspects of estimation are critical for the requirements analyst?**
11. **You advise doing "inspections" for all requirements-related documents. Why shouldn't we be satisfied with doing peer reviews of them? How are inspections different from peer reviews, and why go to the extra trouble? What "type" of inspection is best?**



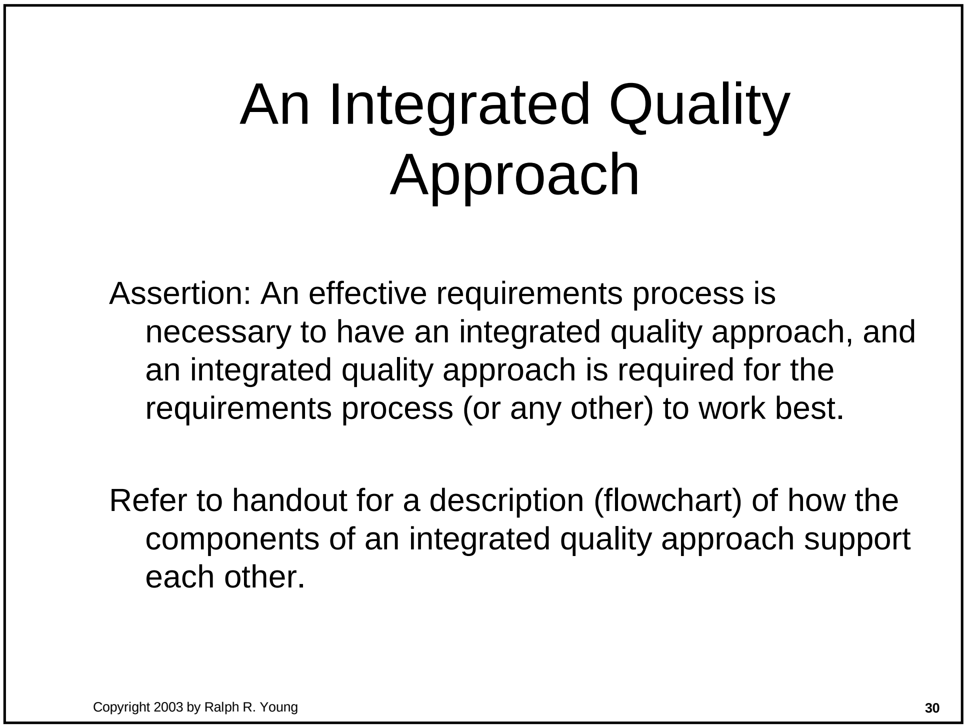
Requirements Analyst's Specially Skills – continued

12. **You have placed a lot of emphasis on quality. How can the requirements analyst help apply quality principles on systems and software development projects?**
13. **There seems to be a lot of confusion in our industry concerning the terms "verification" and "validation." Can you explain why this is so and also clarify suggested uses of the two terms?**
14. **The "agilists" advocate that agile development methodologies promise higher customer satisfaction, lower defect rates, faster development times, and a solution to rapidly changing requirements. Should I recommend that we consider agile development methods on my project?**



Requirements Analyst's Specially Skills – continued

15. **The value of practical knowledge.**
16. **What if my project manager and/or our organization's management team and/or our customer do not support the concept of process improvement?**
17. **How should the Work Breakdown Structure be applied?**
18. **What is a good approach to consider "requirements risks?"**

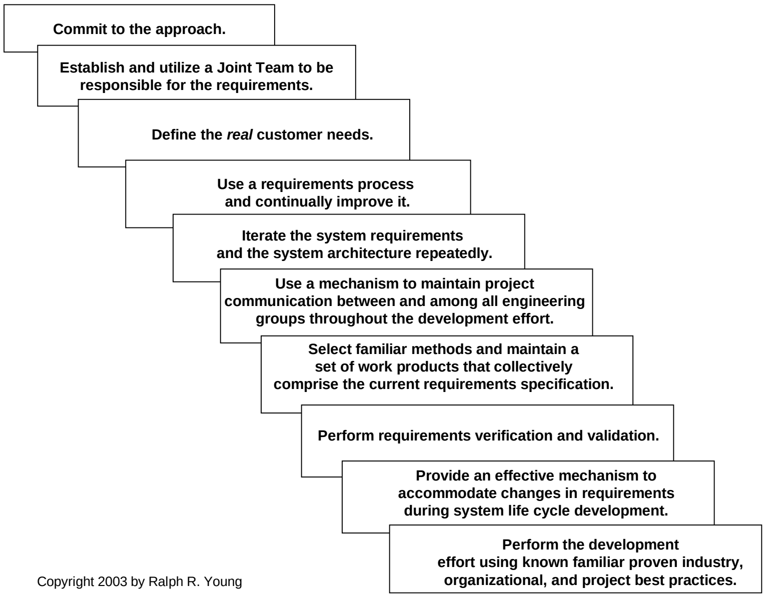


An Integrated Quality Approach

Assertion: An effective requirements process is necessary to have an integrated quality approach, and an integrated quality approach is required for the requirements process (or any other) to work best.

Refer to handout for a description (flowchart) of how the components of an integrated quality approach support each other.

Effective Requirements Practices



Commit to the approach.

Establish and utilize a Joint Team to be responsible for the requirements.

Define the *real* customer needs.

Use a requirements process and continually improve it.

Iterate the system requirements and the system architecture repeatedly.

Use a mechanism to maintain project communication between and among all engineering groups throughout the development effort.

Select familiar methods and maintain a set of work products that collectively comprise the current requirements specification.

Perform requirements verification and validation.

Provide an effective mechanism to accommodate changes in requirements during system life cycle development.

Perform the development effort using known familiar proven industry, organizational, and project best practices.

On Prioritizing Requirements

Why is this an issue?

- We assume that all requirements are of equal importance and priority.
- It's common sense that systems can't (and shouldn't try to) do "everything" for everyone.
- We don't understand that we should "Meet Minimum Requirements: Anything More is Too Much."
- We don't have enough time and money to do "everything."
- It's important that we achieve high quality objectives in everything that we do.

What should we do about it?

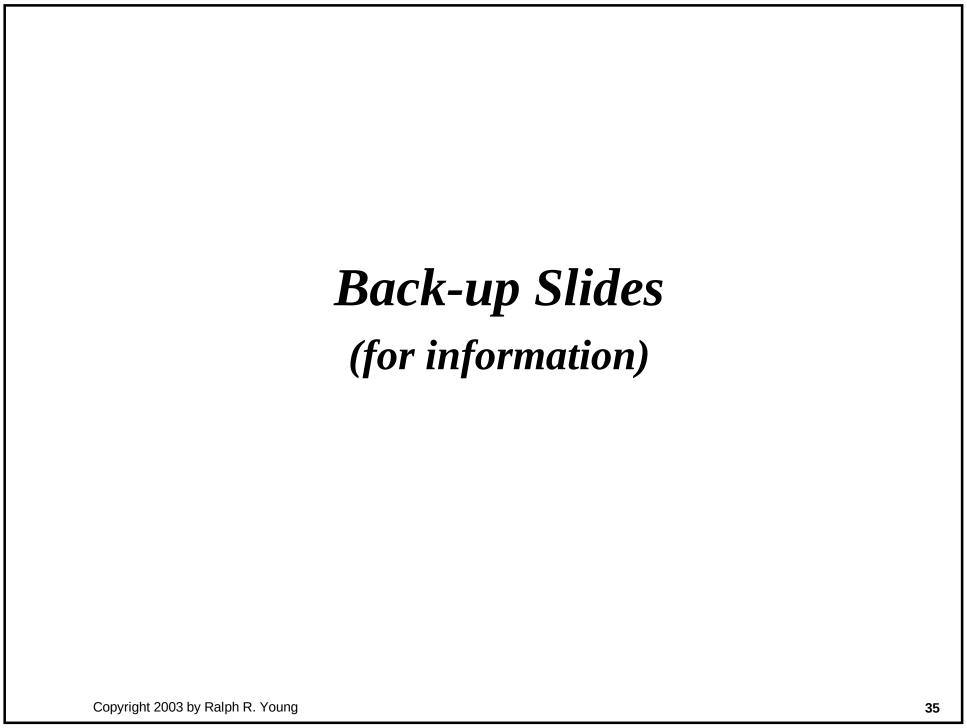
- Use the "Joint Team" mechanism to prioritize all requirements.
- Agree to address top priority requirements in the first baseline/release/version.
- Agree to address the next priority requirements in subsequent baselines/releases/versions.
- Agree not to include "low priority" requirements within the scope of the system, if appropriate.

The Bottom Line

- We need to provide better training and education for requirements analysts.
- Improving requirements practices is a high-leverage activity with good ROI.
- We can have a huge impact by changing a few of the things we do.
- A well kept industry secret: we don't actually change our practices, even when we know how to do better. [See Watts Hamphey, "Why Don't They Practice What We Preach?" - www.sei.cmu.edu/publications/articles/practice-preach/ practice-preach.html]
- Consider encouraging practitioners to develop their own personal "Commitment List," get committed to it, and never let go.
- It is only by making a few improvements in the actual practices that we will get better at what we do.
- Inculcating values of continuous improvement, teamwork, and support of each other will help.

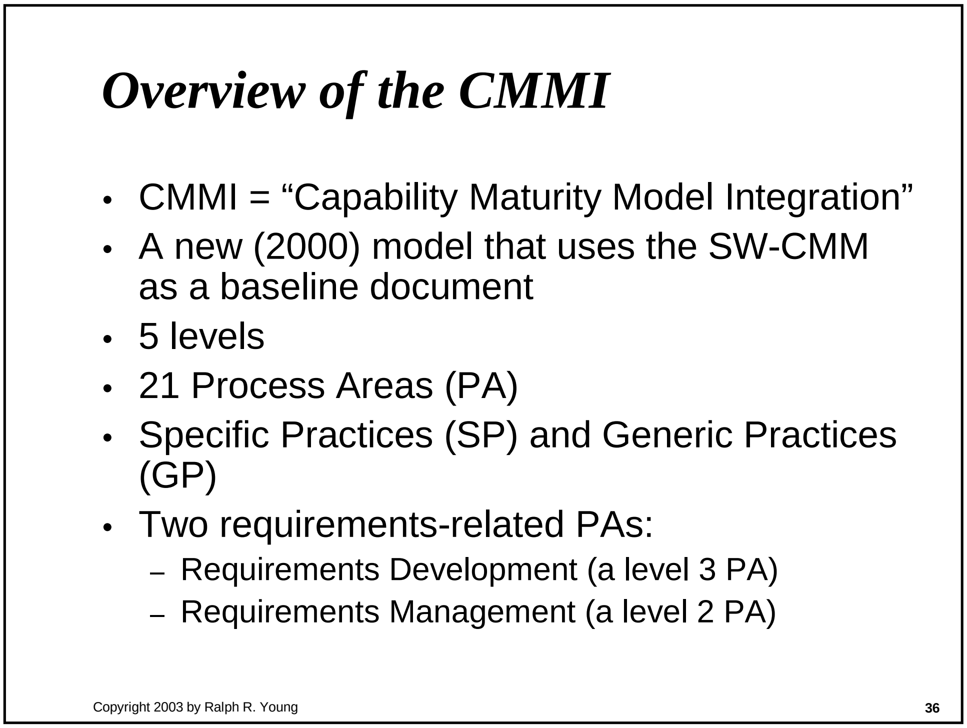
Thank you!

Please let me hear from you!



Back-up Slides

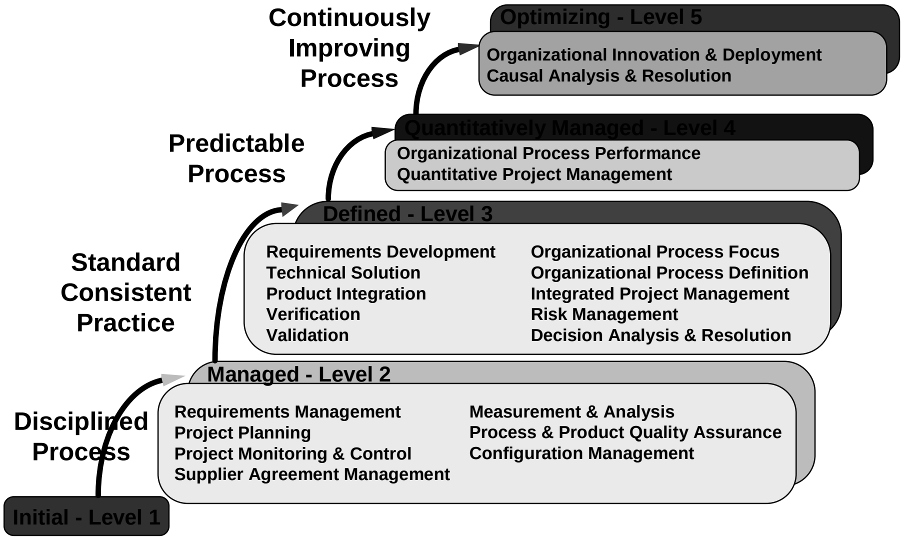
(for information)



Overview of the CMMI

- CMMI = “Capability Maturity Model Integration”
- A new (2000) model that uses the SW-CMM as a baseline document
- 5 levels
- 21 Process Areas (PA)
- Specific Practices (SP) and Generic Practices (GP)
- Two requirements-related PAs:
  - Requirements Development (a level 3 PA)
  - Requirements Management (a level 2 PA)

CMMI Process Areas (PAs) by Maturity Level


Continuously Improving Process
Optimizing - Level 5
Organizational Innovation & Deployment
Causal Analysis & Resolution
Predictable Process
Quantitatively Managed - Level 4
Organizational Process Performance
Quantitative Project Management
Standard Consistent Practice
Defined - Level 3
Requirements Development
Technical Solution
Product Integration
Verification
Validation
Organizational Process Focus
Organizational Process Definition
Integrated Project Management
Risk Management
Decision Analysis & Resolution
Disciplined Process
Managed - Level 2
Requirements Management
Project Planning
Project Monitoring & Control
Supplier Agreement Management
Measurement & Analysis
Process & Product Quality Assurance
Configuration Management
Initial - Level 1

The Requirements Development (RD) Process Area (PA) in the CMMI

- Produce and analyze customer, product, and product component requirements
- Activities
  - SP 1.1. Elicit needs
  - SP 1.2. Transform stakeholder needs, expectations, constraints, and interfaces into customer requirements
  - SP 2.1. Establish product and product component requirements
  - SP 2.2. Allocate product component requirements
  - SP 2.3. Identify interface requirements
  - SP 3.1. Establish operational concepts and scenarios
  - SP 3.2. Establish a definition of required functionality
  - SP 3.3. Analyze requirements
  - SP 3.4. Evaluate product cost, schedule, and risk
  - SP 3.5. Validate requirements with comprehensive methods

The Requirements Management (REQM) PA in the CMMI

- Maintain the project’s product or component requirements and keep the project’s plans, activities, and work products consistent with them
- Activities
  - SP 1.1: Obtain an understanding of requirements
  - SP 1.2: Obtain commitment to requirements
  - SP 1.3: Manage requirements changes
  - SP 1.4: Maintain bi-directional traceability of requirements
  - SP 1.5: Identify inconsistencies between project work and requirements

How Can the CMM/CMMI be Used as a Framework for Engineering Improvement Efforts?

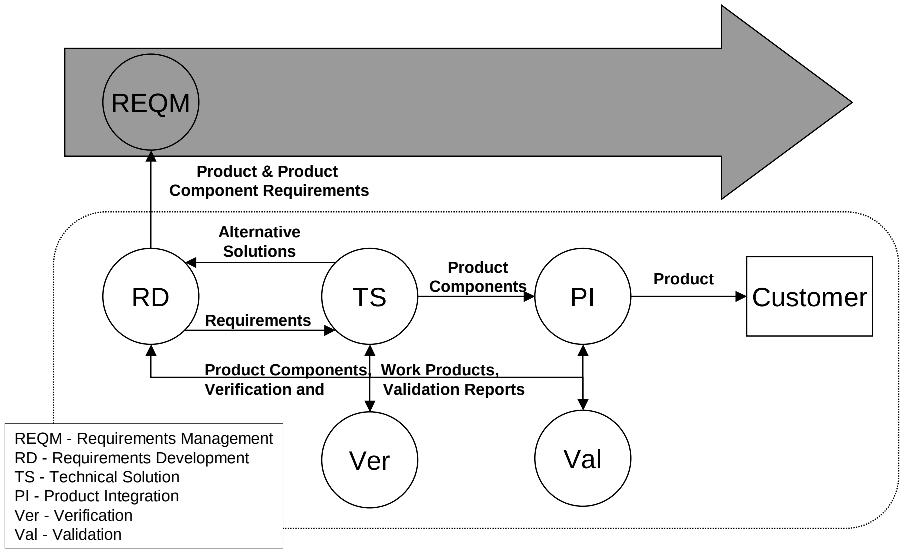
1
- Guidance
  - develop documented or undocumented processes

2
- Measurement
  - benchmark
  - baseline an organization
  - measure improvement

3
- Prioritization
  - prioritize initiatives using assessment data and/or staging and capability levels

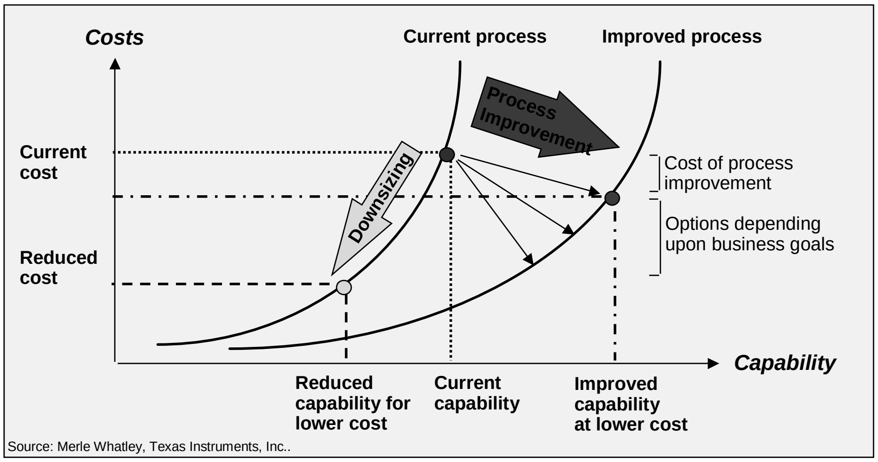
4
- Risk Reduction
  - criteria in source selection

Requirements PAs Related to Other CMMI PAs


REQM
Product & Product Component Requirements
Alternative Solutions
RD
Requirements
TS
Product Components
PI
Product
Customer
Product Components, Work Products,
Verification and Validation Reports
Ver
Val

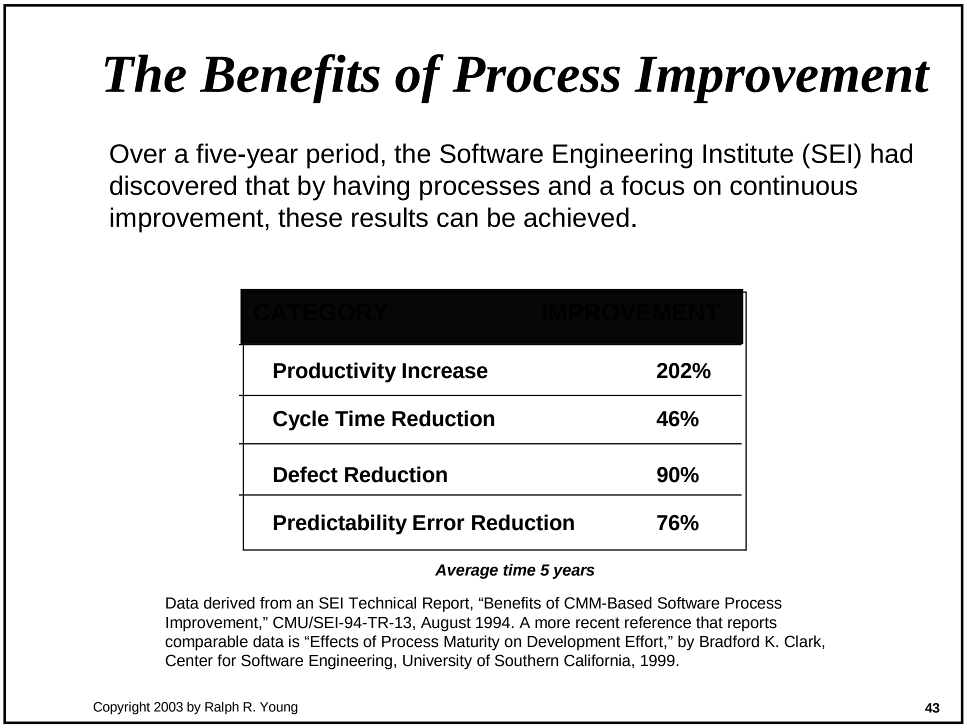
REQM - Requirements Management
RD - Requirements Development
TS - Technical Solution
PI - Product Integration
Ver - Verification
Val - Validation

The Rationale for Process Improvement



Source: Merle Whatley, Texas Instruments, Inc..

CMMI is a tool for improving the ability to transition to an improved process effectively



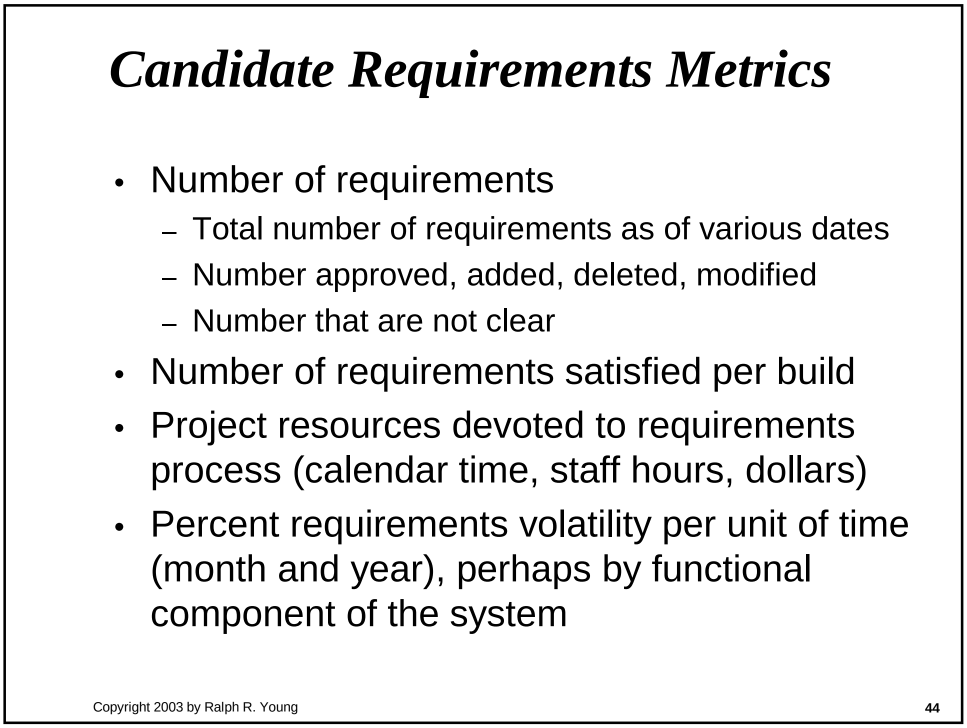
The Benefits of Process Improvement

Over a five-year period, the Software Engineering Institute (SEI) had discovered that by having processes and a focus on continuous improvement, these results can be achieved.

CATEGORY	IMPROVEMENT
Productivity Increase	**202%**
Cycle Time Reduction	**46%**
Defect Reduction	**90%**
Predictability Error Reduction	**76%**

Average time 5 years

Data derived from an SEI Technical Report, “Benefits of CMM-Based Software Process Improvement,” CMU/SEI-94-TR-13, August 1994. A more recent reference that reports comparable data is “Effects of Process Maturity on Development Effort,” by Bradford K. Clark, Center for Software Engineering, University of Southern California, 1999.

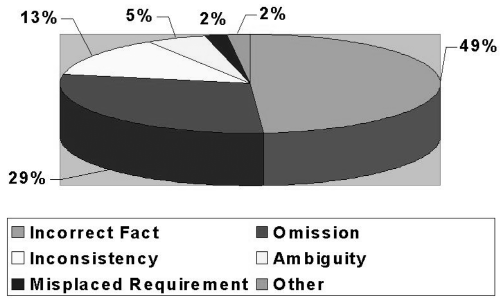


Candidate Requirements Metrics

- Number of requirements
  - Total number of requirements as of various dates
  - Number approved, added, deleted, modified
  - Number that are not clear
- Number of requirements satisfied per build
- Project resources devoted to requirements process (calendar time, staff hours, dollars)
- Percent requirements volatility per unit of time (month and year), perhaps by functional component of the system

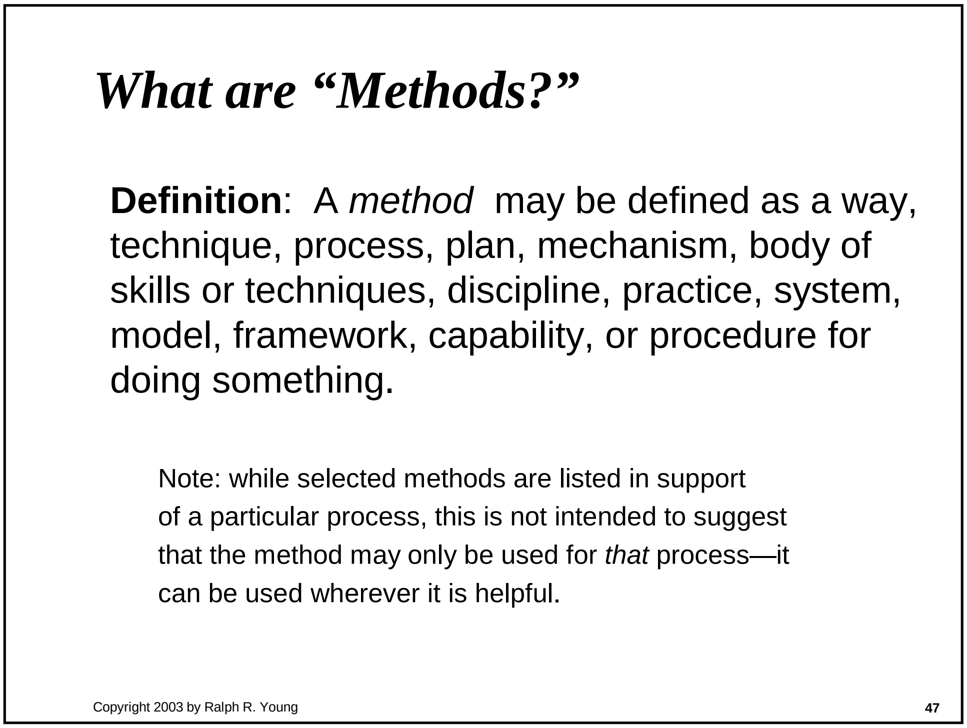
Types of Requirements Errors

(with thanks to Ivy Hooks)



Needed Management Commitment for Effective Requirements Engineering

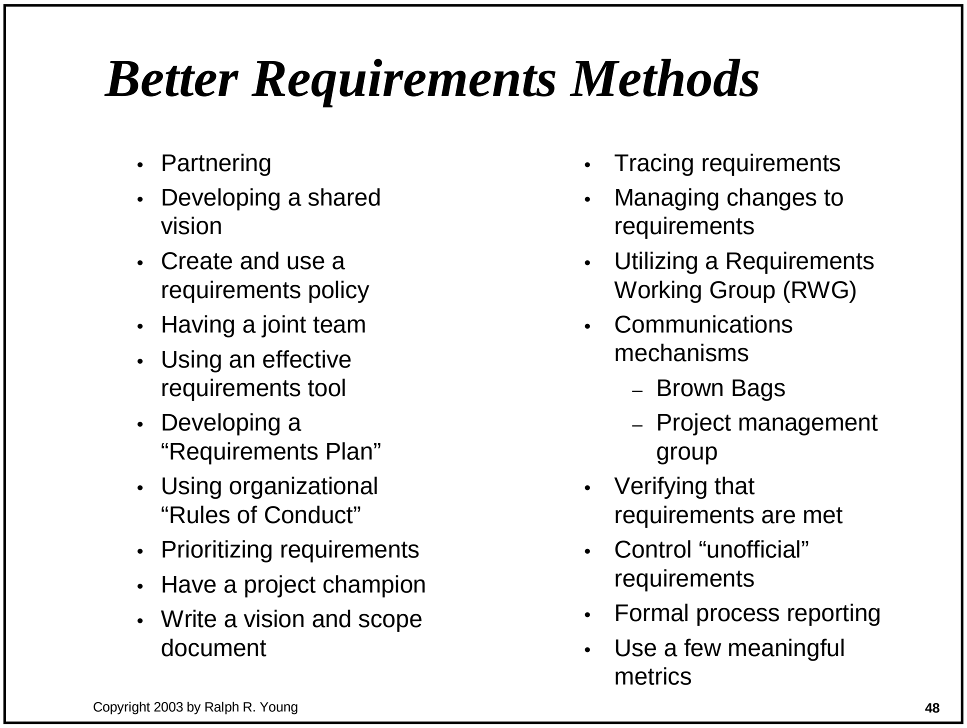
1. Investment of 8-14% of project costs in the system life-cycle requirements process (the industry average is 3%)
2. Sponsorship and funding of formal training for requirements analysts and quality improvement for all
3. Expectations that effective requirements practices are used, projects are successful (meet customer needs and are completed within 15% of budget and planned schedule), and rework is less than 15%
4. Support for strengthening teamwork, process design and use, continuous improvement, inter-personal skills, good relationships, and a "quality culture"
5. Sponsorship of an organizational requirements working group to provide a mechanism to share and improve requirements-related activities
6. Foster effective communication
7. Avoid blame



What are "Methods?"

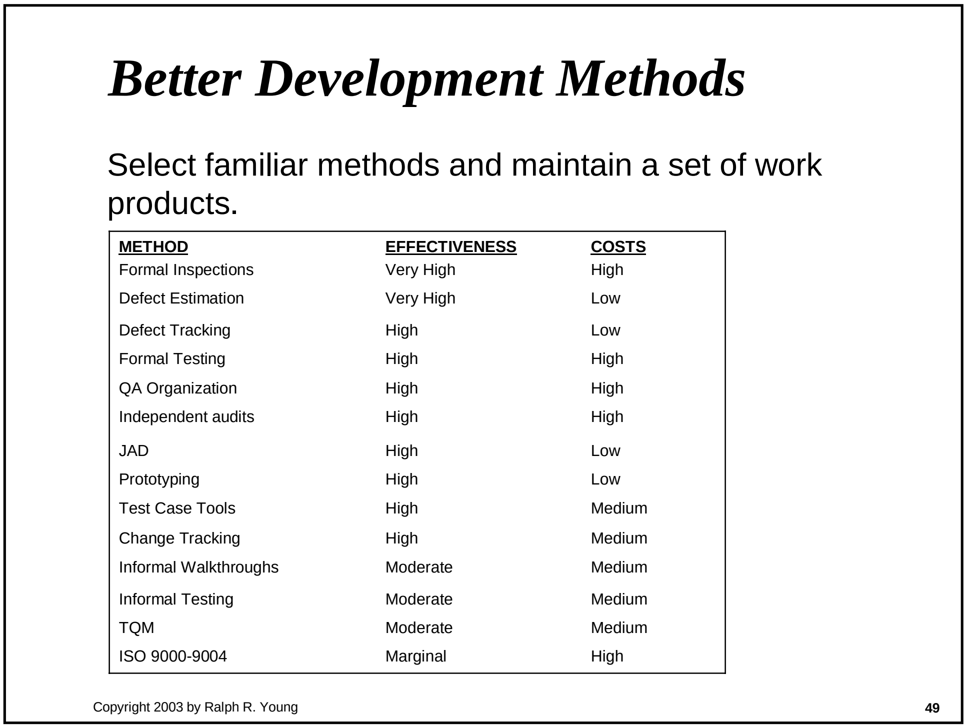
Definition: A *method* may be defined as a way, technique, process, plan, mechanism, body of skills or techniques, discipline, practice, system, model, framework, capability, or procedure for doing something.

Note: while selected methods are listed in support of a particular process, this is not intended to suggest that the method may only be used for *that* process—it can be used wherever it is helpful.



Better Requirements Methods

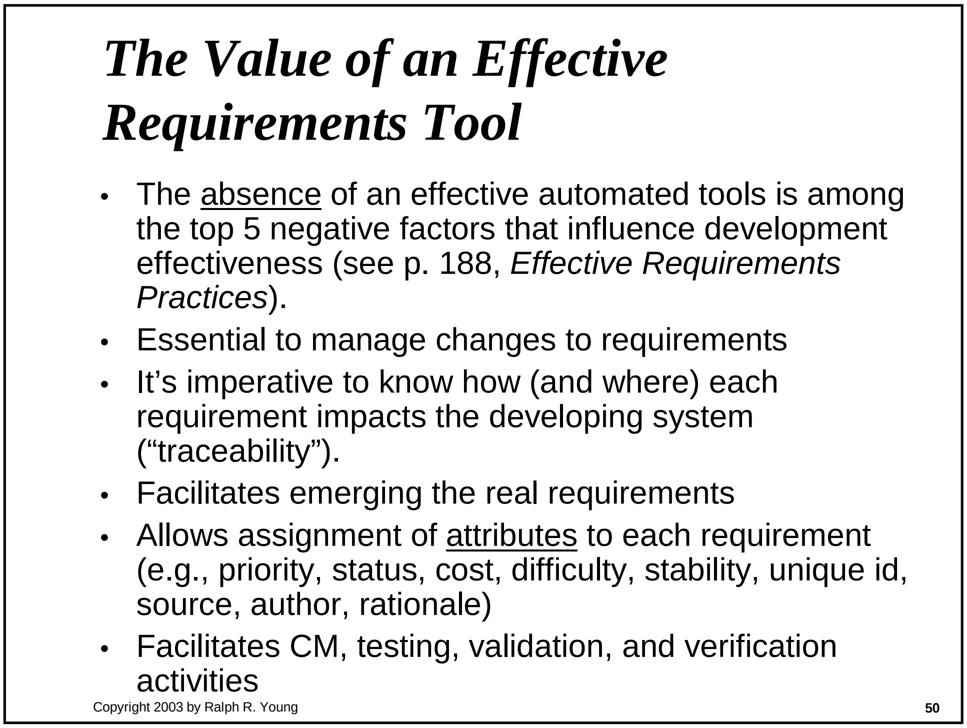
- Partnering
- Developing a shared vision
- Create and use a requirements policy
- Having a joint team
- Using an effective requirements tool
- Developing a "Requirements Plan"
- Using organizational "Rules of Conduct"
- Prioritizing requirements
- Have a project champion
- Write a vision and scope document
- Tracing requirements
- Managing changes to requirements
- Utilizing a Requirements Working Group (RWG)
- Communications mechanisms
  - Brown Bags
  - Project management group
- Verifying that requirements are met
- Control "unofficial" requirements
- Formal process reporting
- Use a few meaningful metrics



Better Development Methods

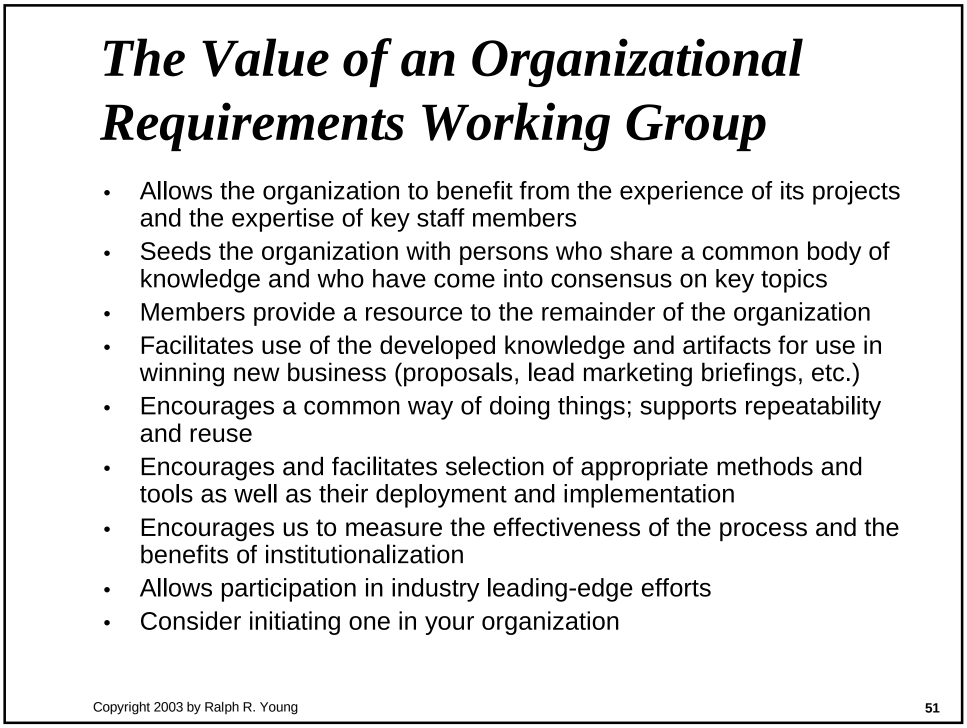
Select familiar methods and maintain a set of work products.

METHOD	EFFECTIVENESS	COSTS
Formal Inspections	Very High	High
Defect Estimation	Very High	Low
Defect Tracking	High	Low
Formal Testing	High	High
QA Organization	High	High
Independent audits	High	High
JAD	High	Low
Prototyping	High	Low
Test Case Tools	High	Medium
Change Tracking	High	Medium
Informal Walkthroughs	Moderate	Medium
Informal Testing	Moderate	Medium
TQM	Moderate	Medium
ISO 9000-9004	Marginal	High



The Value of an Effective Requirements Tool

- The <u>absence</u> of an effective automated tools is among the top 5 negative factors that influence development effectiveness (see p. 188, *Effective Requirements Practices*).
- Essential to manage changes to requirements
- It’s imperative to know how (and where) each requirement impacts the developing system (“traceability”).
- Facilitates emerging the real requirements
- Allows assignment of <u>attributes</u> to each requirement (e.g., priority, status, cost, difficulty, stability, unique id, source, author, rationale)
- Facilitates CM, testing, validation, and verification activities



The Value of an Organizational Requirements Working Group

- Allows the organization to benefit from the experience of its projects and the expertise of key staff members
- Seeds the organization with persons who share a common body of knowledge and who have come into consensus on key topics
- Members provide a resource to the remainder of the organization
- Facilitates use of the developed knowledge and artifacts for use in winning new business (proposals, lead marketing briefings, etc.)
- Encourages a common way of doing things; supports repeatability and reuse
- Encourages and facilitates selection of appropriate methods and tools as well as their deployment and implementation
- Encourages us to measure the effectiveness of the process and the benefits of institutionalization
- Allows participation in industry leading-edge efforts
- Consider initiating one in your organization