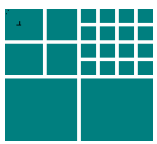
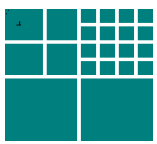
Using the Experience Factory to Improve the Software Acquisition Process

Victor R. Basili
University of Maryland
and
Fraunhofer Center for Experimental Software Engineering– Maryland



Motivation for an Empirical Approach

- Software acquisition teams need to understand the right models and techniques to support their activities. For example:
  - What level of information do I need from a contractor to keep track of and understand the progress towards my goals?
  - How should you select and tailor an acquisition lifecycle model for the particular environment?
  - How do you judge the credibility of the cost estimates provided by the bidder?
- Too often, such decisions are based on opinion and personal experience, made without a reasonable basis for judgement
- How do other disciplines build knowledge about
  - the elements of their discipline, e.g., their products and processes
  - the relationships between those elements



Examples of Using Empirical Results

Minimizing Acquisition Process Steps

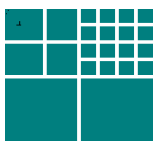
When can I get away with a minimal level of process in my acquisition processes, i.e., only the absolutely necessary activities?

There is evidence that

- a minimal process is possible for projects that are less than 10 months, under $50K, and less than 10 people, have stable requirements, and use a known technology

Implications for empirically based software acquisition:

- From a cost effectiveness point of view, I can identify the minimum set of processes that have been demonstrated necessary in past projects and concentrate on only those.



Maximizing Acquisition Process Steps

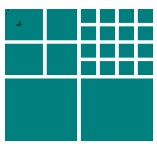
When do I need a robust software acquisition process with a high level of detail, i.e., high degree of formality, full set of steps, … ?

There is evidence that

- a robust process is needed for projects of more than 24 months, more than a million dollars, and more than 30 people, and have volatile requirements using new technology.

Implications for empirically based software acquisition:

- I need to put a full acquisition process in place, including full lifecycle planning, for large systems.



Process Customization

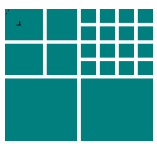
What level of process detail is needed for customizing acquisition processes?

There is evidence that there are at least three levels of detail available in process

- minimal process
- controlled process, needed for projects that are 12 to 36 months, under a million dollars, and less than 30 people
- a robust process

Implications for empirically based software acquisition:

- The better you can articulate your project characteristics, the more effectively you can choose and tailor process.



Building empirical evidence

for evolving software acquisition processes

Create a corporate memory - baselines/models of current practices
e.g., how much will a new project cost?

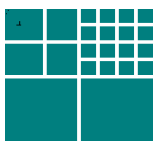
Plan, track and control the acquisition process
e.g., what should happen, is it happening?

Determine strengths and weaknesses of the current process and product
e.g., are there problems with certain steps in the acquisition process?

Develop a rationale for adopting/refining acquisition techniques
e.g., what is the right level of process for a particular product acquisition?

Assess the impact of techniques
e.g., does our model provide the right cost estimates?

Evaluate the quality of the process/product
e.g., are we achieving the right product functionality/reliability?



One Motivation for the Approach

Experiences with the Software Engineering Laboratory (SEL)

Consortium of NASA/GSFC, CSC, UM, established in 1976

Goal to improve the process and product quality

- using observation, experimentation, learning, and model building

Learned a great deal (e.g., what worked and didn't work)

Observation played a key role

Measurement was used to capture knowledge and experiences

Feedback loops provided an environment for learning

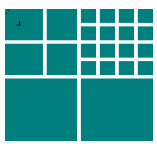
Generated lessons learned and packaged into the process, product and organizational structure

Made measurable improvements in the processes and products

The Software Engineering Laboratory was awarded the first

IEEE Computer Society Award for Software Process Achievement in 1994

for demonstrable, sustained, measured, significant process improvement



Basic Concepts for Empirical Software Engineering

The following concepts have been applied in a number of organizations

Quality Improvement Paradigm (QIP)

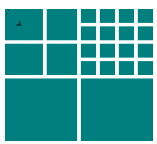
An evolutionary learning paradigm tailored for the software business

Goal/Question/Metric Paradigm (GQM)

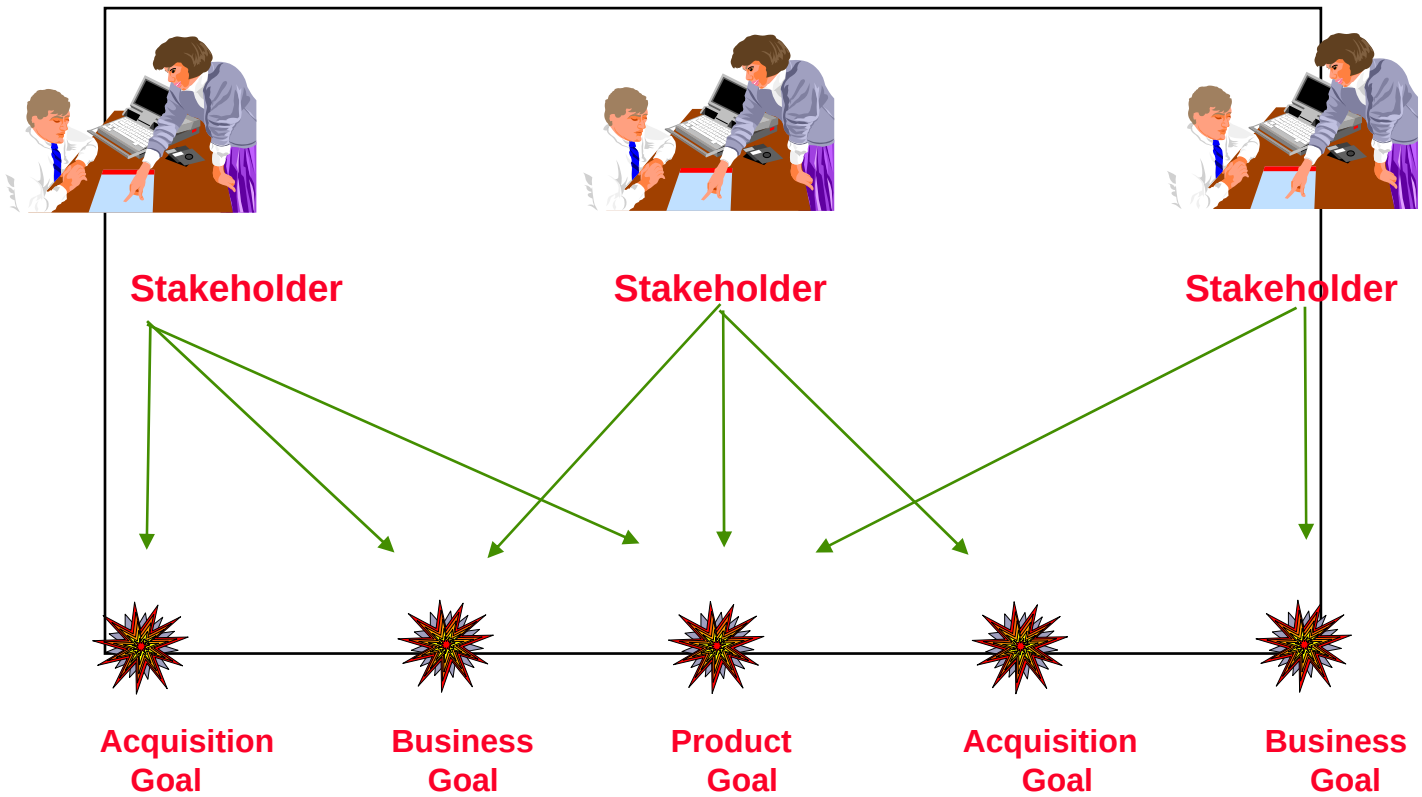
An approach for establishing project and corporate goals and a mechanism for measuring against those goals

Experience Factory (EF)

An organizational approach for building software competencies and supplying them to projects

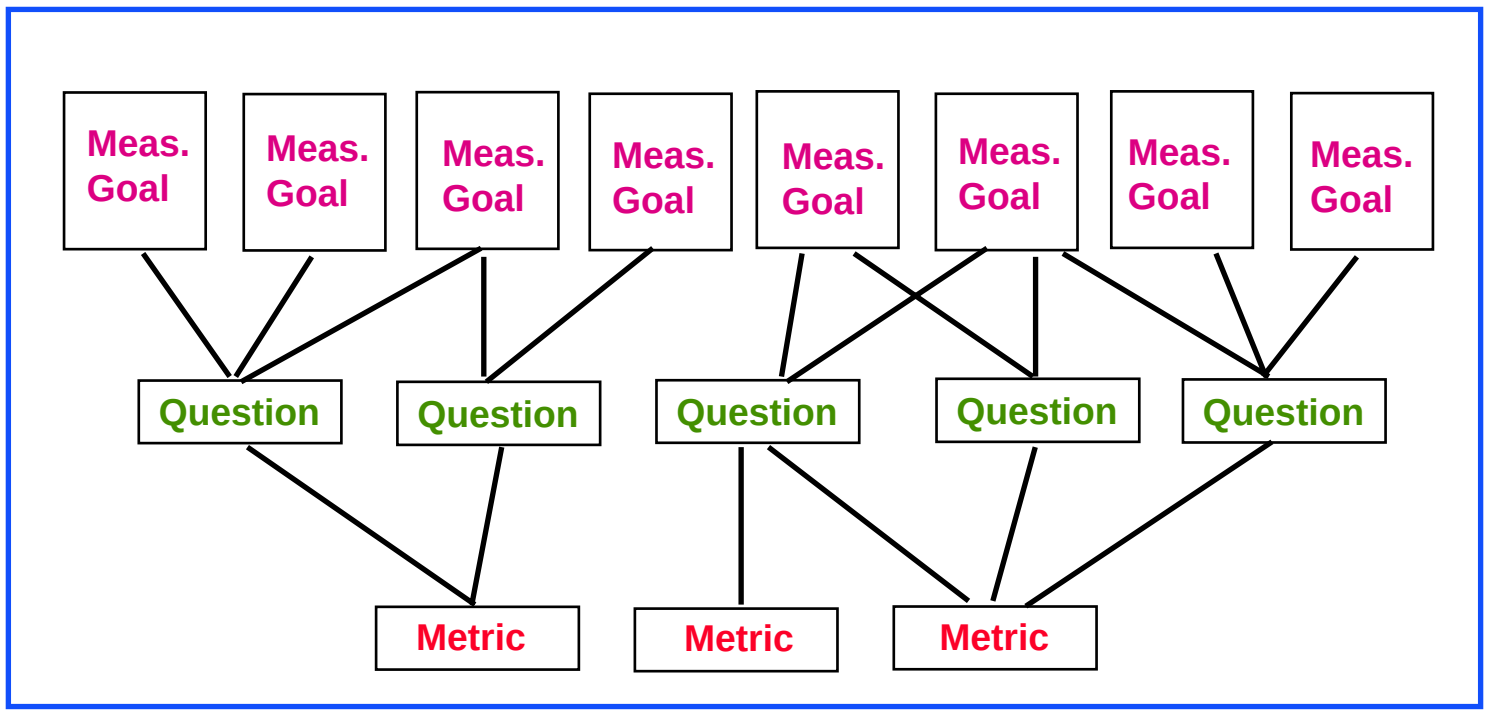


THE MEASUREMENT INFRASTRUCTURE



- Internal and external customers have their own goals
- Well defined goals enable business success

THE MEASUREMENT INFRASTRUCTURE
Goal Based Measurement



- Each metric supports multiple goals
- Questions focus metric selection and in-process analysis

Example COTS Acquisition Process

Business Goal: Reduce the cost of the COTS acquisition process

Measurement Goal: Characterize the costs involved in the pre-selection process

What are the pre-selection activities?

Gather information on available sources

Survey several contractor's offerings

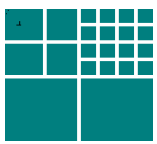
Solicit multiple qualified suppliers

Prepare short list

Compare vendor history and experience

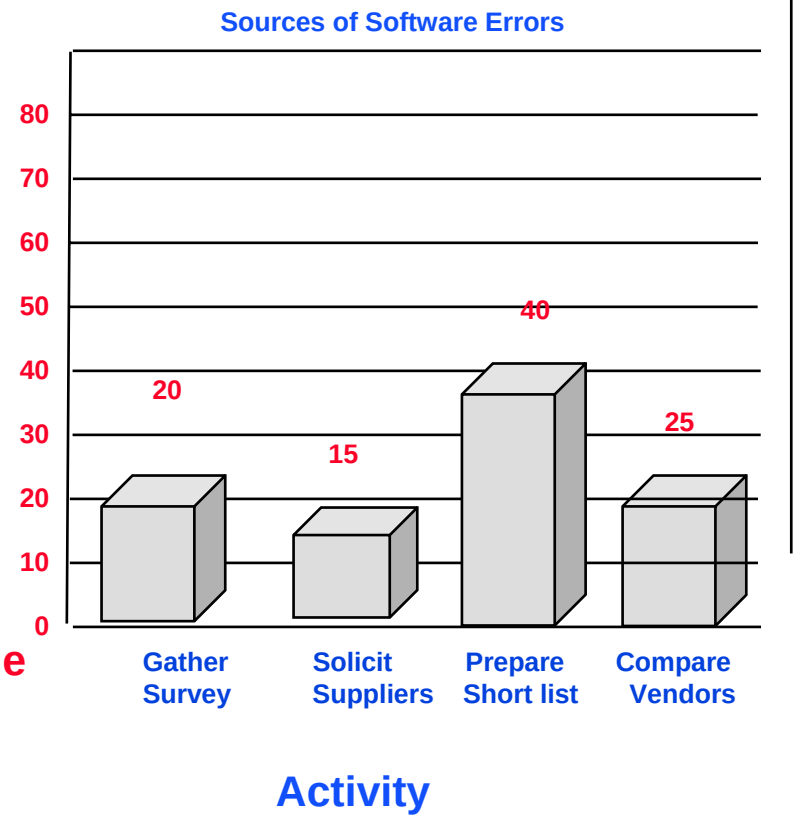
Question: What is the relative cost of each activity?

Metrics: % time spent gathering, surveying, …

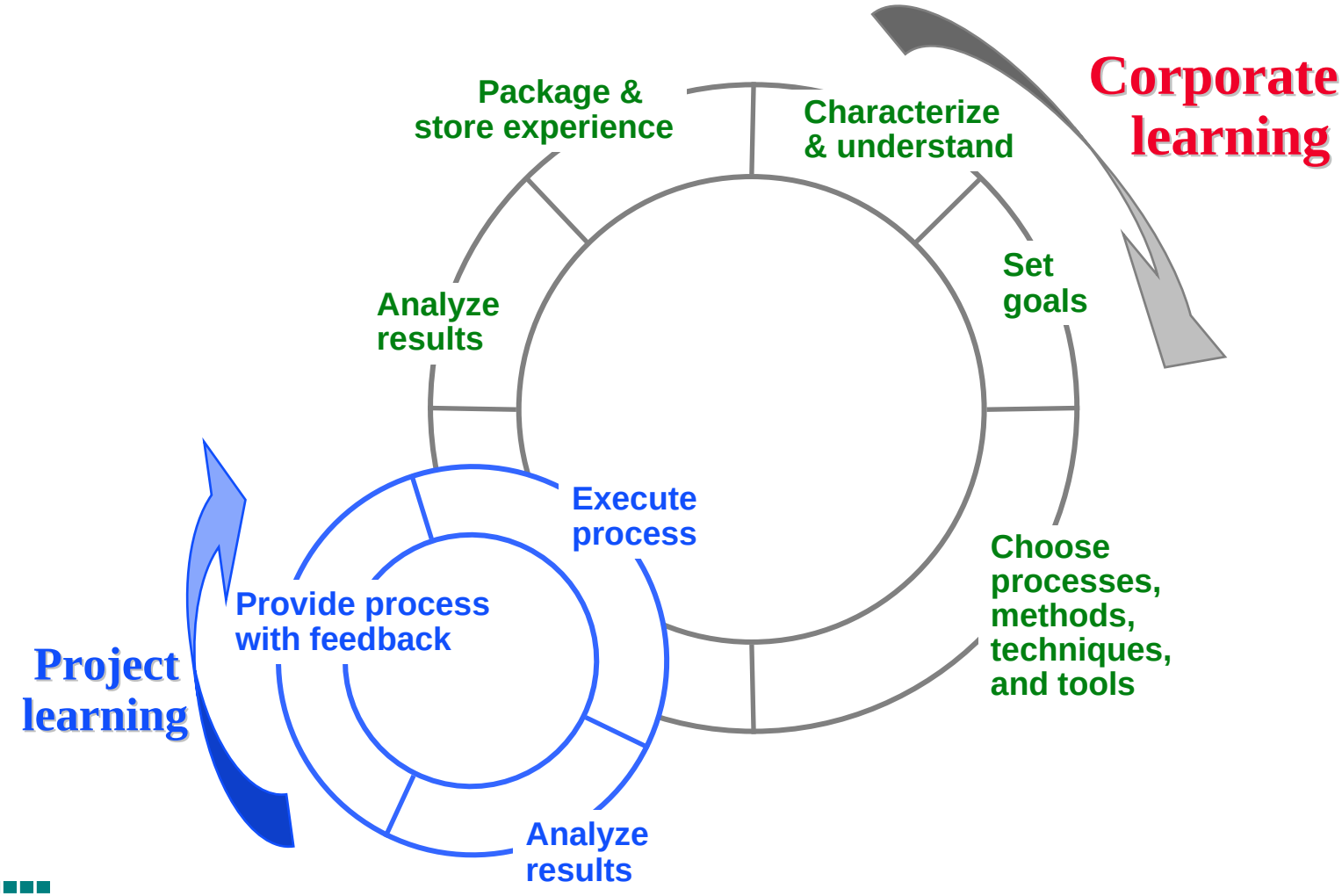


DEFINING MEASUREMENT GOALS
A GOAL/QUESTION/METRIC EXAMPLE

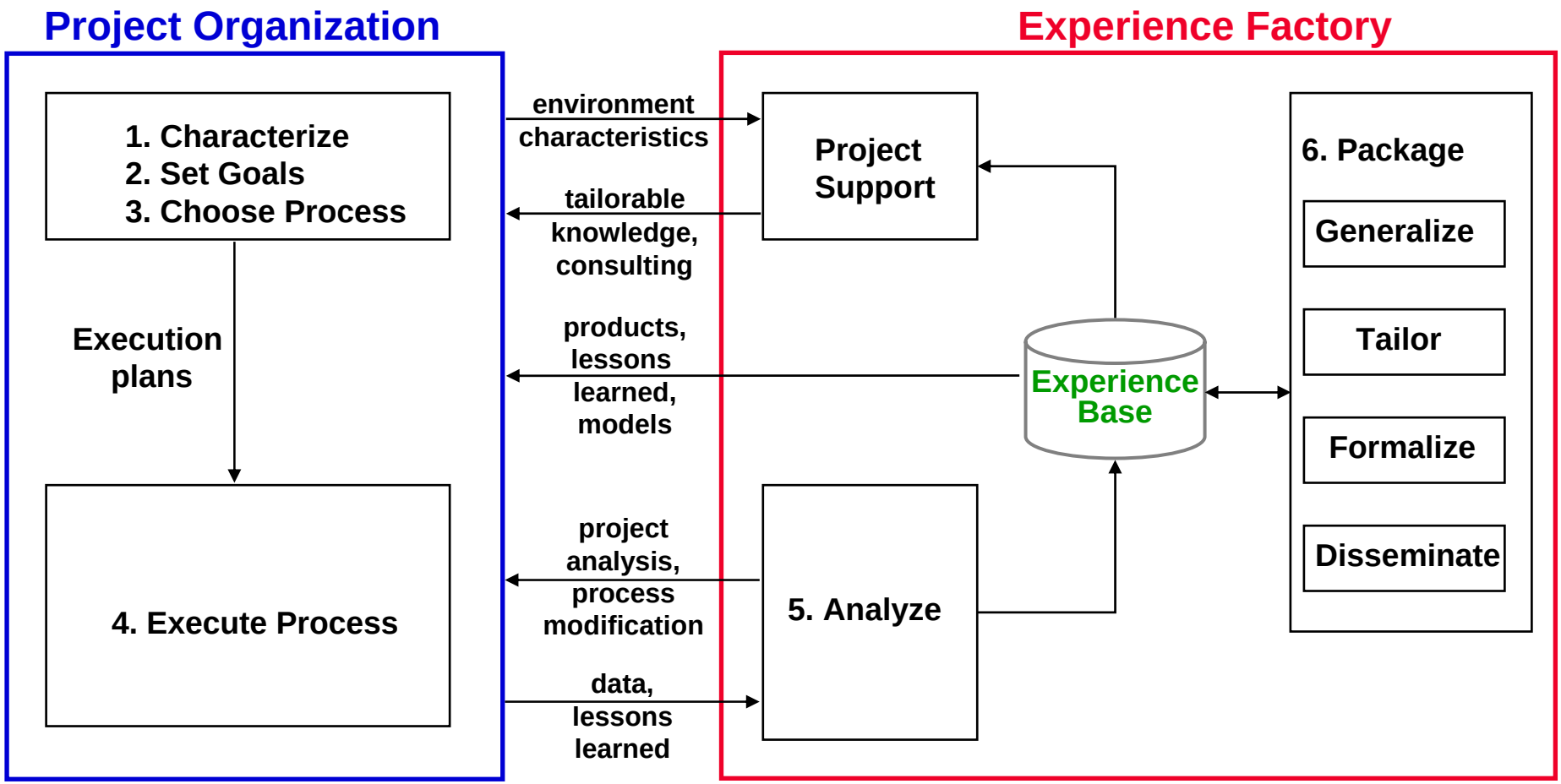
- **Business Goal**
  - Reduce the cost of the COTS acquisition process
- **A Measurement Goal**
  - Characterize the costs involved in the pre-selection process
- **Question**
  -What is the relative cost of each activity?
- **Metrics**
  - Time spent in gathering, surveying, …



Quality Improvement Paradigm

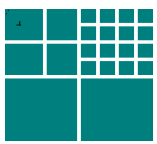


The Experience Factory Organization

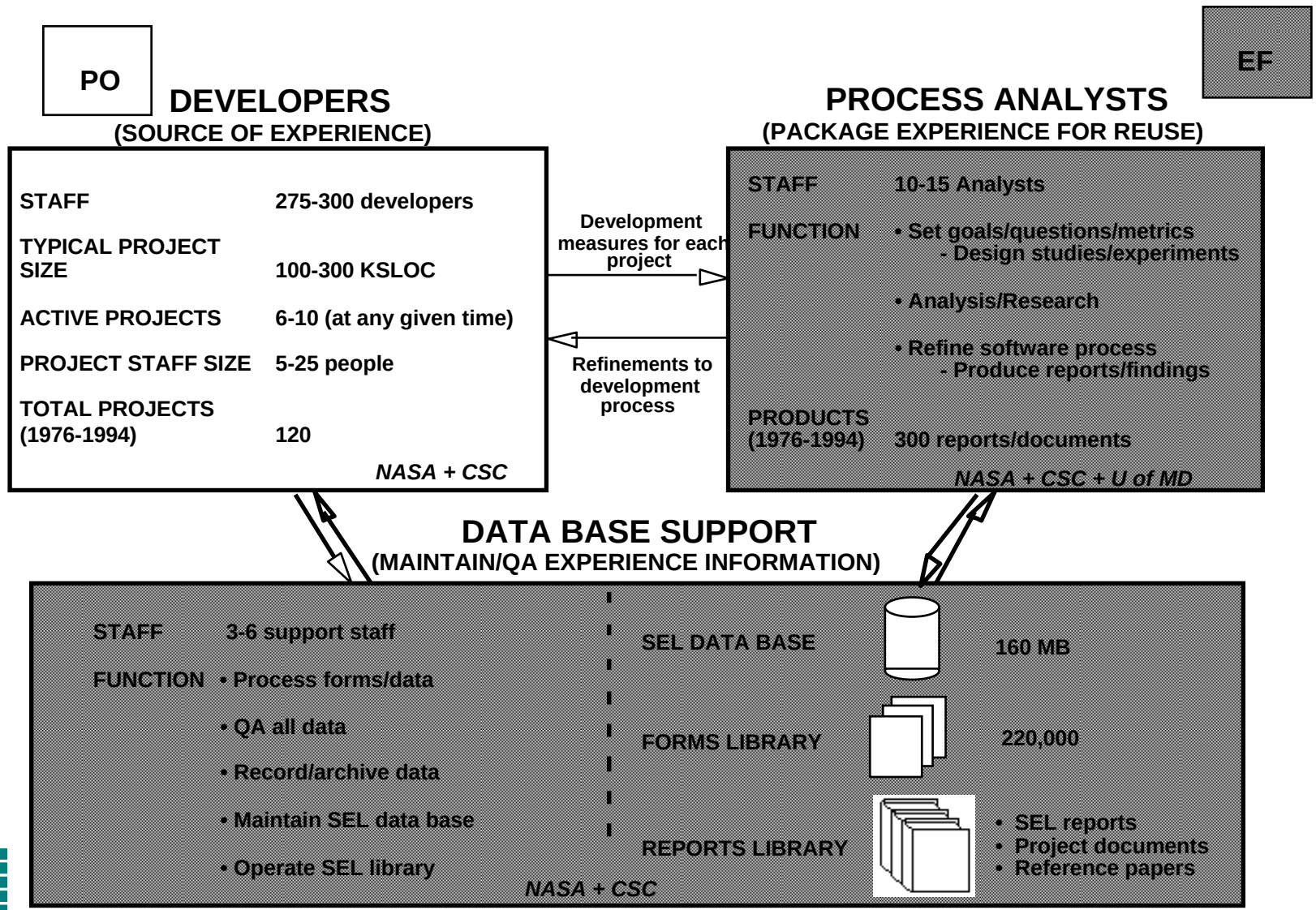


The Experience Factory Organization
A Different Paradigm

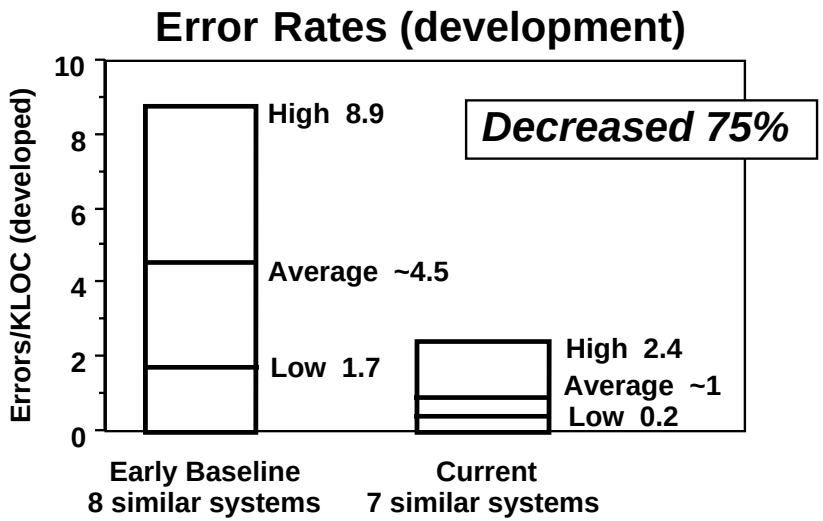
Project Organization **Problem Solving**	Experience Factory **Experience Packaging**
Decomposition of a problem into simpler ones	Unification of different solutions and re-definition of the problem
Instantiation	Generalization, Formalization
Design/Implementation process	Analysis/Synthesis process
Validation and Verification	Experimentation
Product Delivery within Schedule and Cost	**Experience / Recommendations Delivery to Project**

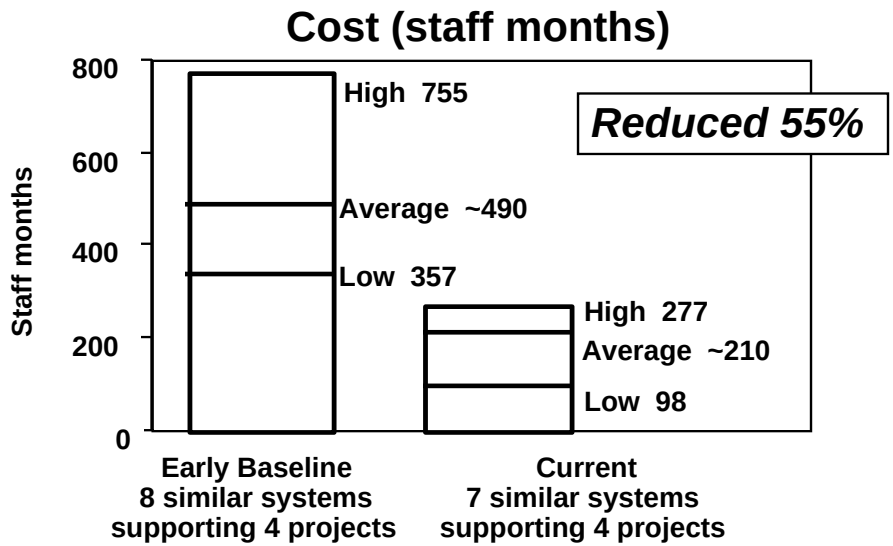


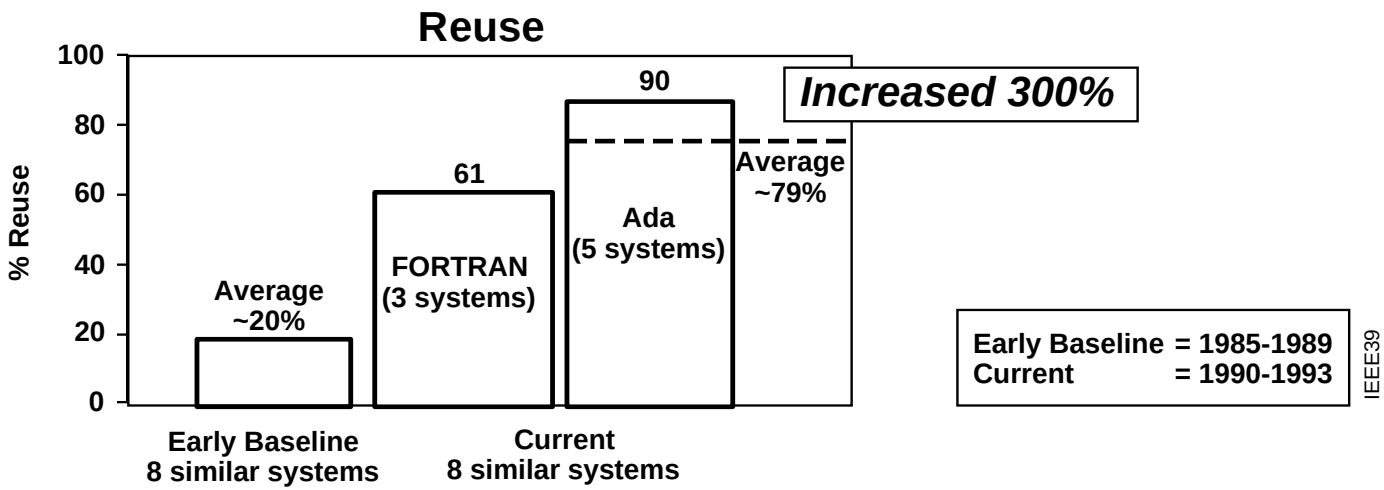
SEL: Example Experience Factory Structure



Using Baselines to Show Improvement 1987 vs. 1991







The SEL Empirical Approach Baselines: 1987, 1991, 1995

CeBASE

Continuous Improvement in the SEL

Decreased **Development Defect rates** by
75% (87 - 91) **37%** (91 - 95)
Reduced **Cost** by
55% (87 - 91) **42%** (91 - 95)
Improved **Reuse** by
300% (87 - 91) **8%** (91 - 95)
Increased **Functionality** five-fold (76 - 92)

CSC

officially assessed as CMM level 5 and ISO certified (1998), starting with SEL organizational elements and activities

Fraunhofer Center

for Experimental Software Engineering - Maryland created 1998

CeBASE

Center for Empirically-Based Software Engineering created 2000

Building a local experience base

for evolving software acquisition processes

Characterize the acquiring and vendor organizations

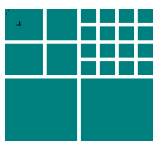
Set goals for successful acquisition and improvement

Select the appropriate **processes** for the goals in the context

Observe and measure the activities

Analyze and synthesize what has been learned into sets of local best practices recognizing what has been effective and under what circumstances allowing for tailoring based upon context variables

Package results for use in a local **experience base** and feed back what has been learned to improve the practices within the organization



Maturing Software Acquisition Models and Measures

CeBASE

Characterize

Describe and differentiate acquisition processes

Build descriptive models and baselines

Understand

Explain associations/dependencies between processes and effects

Discover causal relationships

Analyze models

Evaluate

Assess the achievement of quality goals

Assess the impact of various acquisition processes

Compare models

Predict

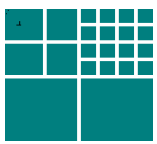
Estimate expected product quality and process resource consumption

Build predictive models

Motivate

Describe what we need to do to manage the contractor

Build prescriptive models



Software Acquisition Models and Measures

CeBASE

Resource Models and Baselines,

e.g., cost models, resource allocation models

Change and Defect Baselines and Models,

e.g., defect/quality prediction models

Product Models and Baselines,

e.g., progress measurement, technical performance measures

Process Definitions and Models,

e.g., acquisition lifecycle models for large and small acquisitions, COTS evaluation models

Method and Technique Evaluations,

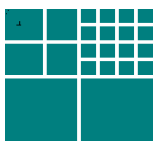
e.g., acquisition risk management methods, contract management methods

Quality Models,

e.g., reliability models, ease of change maintenance, availability models

Lessons Learned,

e.g., risks associated with a performance-based acquisition

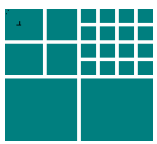


Software Acquisition Methods and Practices

CeBASE

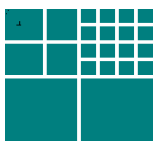
- Example Practices that are defined and evolved
  - Institutionalization Features

  and
  - Software Acquisition Planning
  - Solicitation
  - Contract Tracking & Oversight
  - Requirements Development & Management
  - Project Management
  - Evaluation
  - Transition To Support



Software Acquisition Institutionalized Features

- **Commitment**
  - "actions that the organization must take to establish the process and ensure that it can endure, … typically involves establishing organizational policies and management sponsorship"
- **Ability**
  - "preconditions that must exist in the project or organization to implement the software acquisition process competently"
- **Measurement and analysis**
  - "to determine the status and effectiveness of the activities performed"
- **Verifying implementation**
  - "the steps to ensure that the activities are performed in compliance with the process"



Sharing empirical evidence

for evolving software acquisition processes

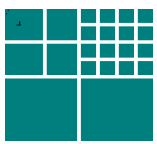
Interact with various industrial, government and academic organizations to open up the domain for learning, e.g., use and contribute to cebase.org, get involved in the Clearing House experience base

Partner with other organizations to expand the potential competencies

Observe and gather as much information as possible

Analyze and synthesize what has been learned into sets of best practices recognizing what has been effective and under what circumstances allowing for tailoring based upon context variables

Package results for use and feed back what has been learned to improve the practices



CeBASE
Center for Empirically Based Software Engineering

The **CeBASE** project was created to support the symbiotic relationship between research and development, and make empirical results sharable by a variety of organizations

Virtual Research Center

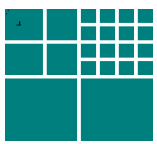
Created by the NSF Information Technology Research Program

Co-Directors: Victor Basili (UMD), Barry Boehm (USC)

Initial technology focus: Defect reduction techniques, COTS based development

CeBASE Framework

Experience Factory, Goal/Question/Metric Approach, Spiral Model extensions, MBASE, WinWin Negotiations, Electronic Process Guide, eWorkshop collaboration, COCOMO cost family, EMS Experience Base, VQI (Virtual Query Interface)



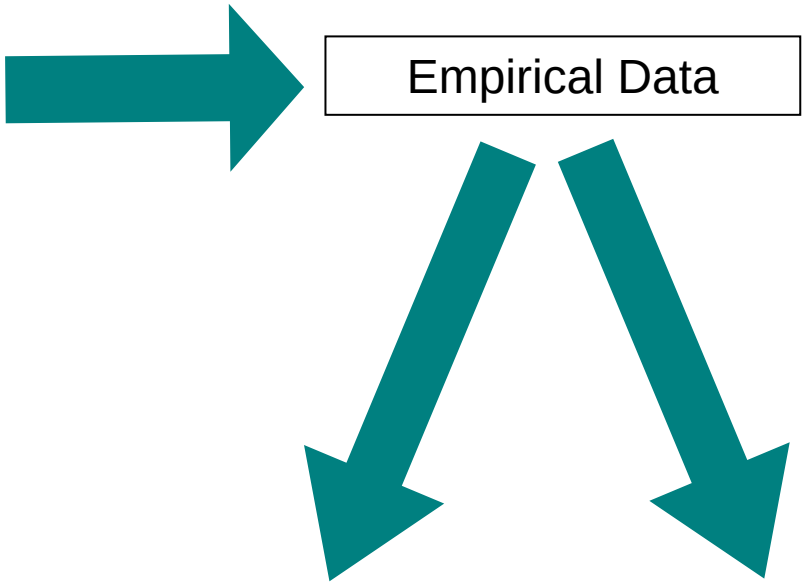
CeBASE
Center for Empirically Based Software Engineering

CeBASE Project Goal: Enable a **decision framework and experience** base that forms a basis and an infrastructure for research and education in empirical methods and software engineering

CeBASE Research Goal: Create and evolve an **empirical research engine** for evaluating and choosing among software development technologies

CeBASE Approach

Observation and Evaluation Studies of Development Technologies and Techniques



Predictive Models

(Quantitative Guidance)

E.g. COCOTS excerpt:

Cost of COTS tailoring = f(# parameters initialized, complexity of script writing, security/access requirements, …**)**

General Heuristics

(Qualitative Guidance)

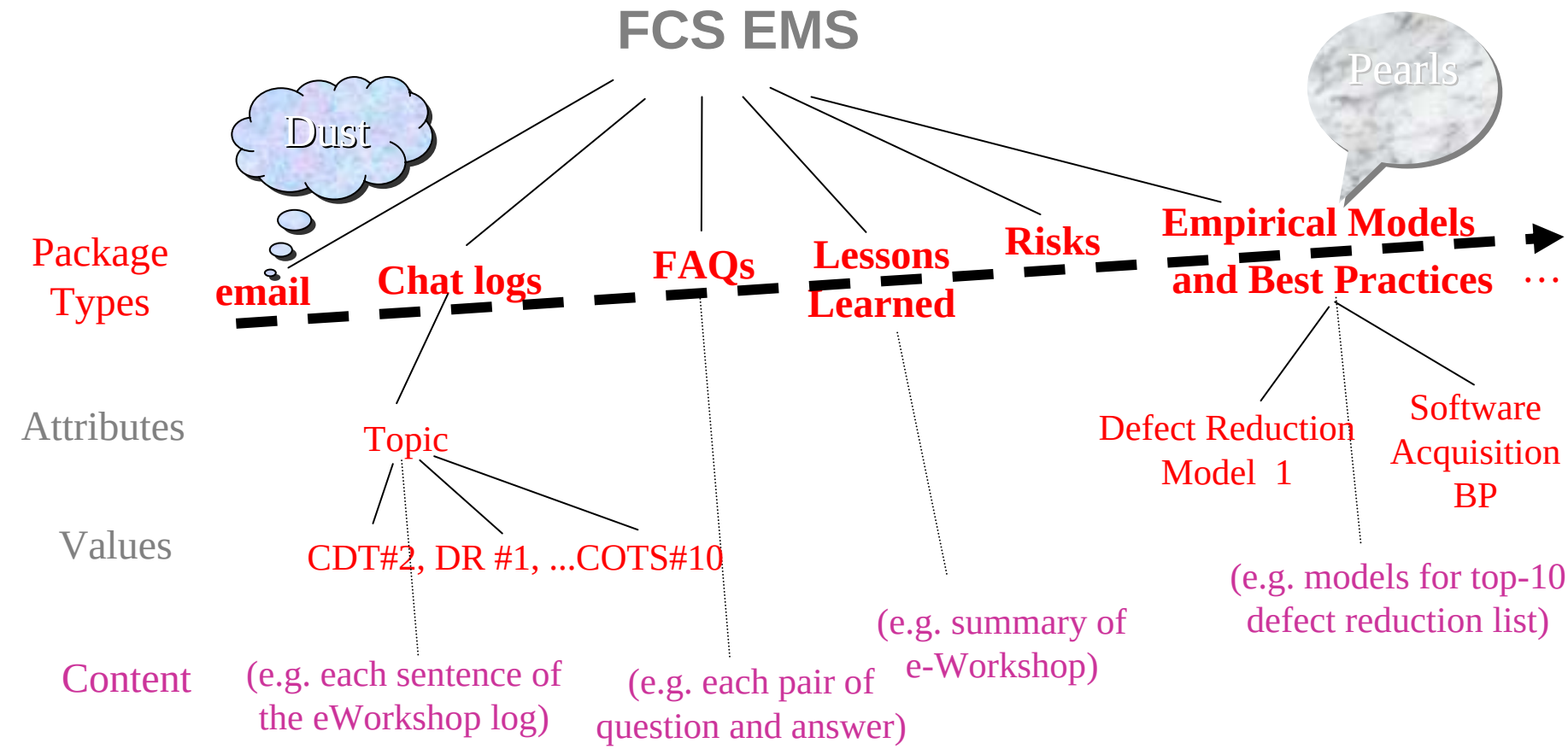
E.g. Process customization Heuristic:

For projects < 10 months, < $50K, < 10 people, have stable requirements, and use a known technology, a minimum process is acceptable.

The Dust-to-Pearls Approach

- Focuses on what people do anyway,
  - Collects that data, analyses, evolves and refines it
- Encourages experts to share by quickly giving value back
  - Instant feedback loop
- Does not add significant work to already busy experts
- Allows the EF Group to analyze data over time
- Allows for organic growth of the EB, according to needs

CeBASE Experience Management System From Dust-to-Pearls

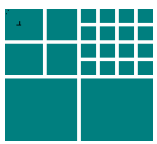


Example COTS Acquisition Lessons Learned

- Capture experience and knowledge for use in COTS acquisition for Complex System of Systems
- Avoid errors and build on strengths
- Support future acquisitions through Office of the Under Secretary of Defense (OSD)

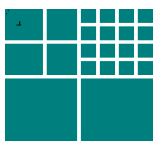
Example Topics:

- How do you pick the right suppliers?
- How do you organize the work?
- How do you make sure that
  - Each supplier builds “the right component?”
  - Each component integrates well?



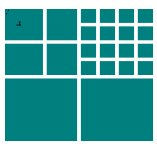
COTS Acquisition LL Example (1/2)

- *Type*: Good practice
- *Statement*: For large, multi vendor solicitations: hold pre-award hearings so that each vendor will have an opportunity to ask questions and all vendors will hear the same response
- *Issue/Risk factor*: Vendor protest situation
- *Recommended action*: Hold pre-award hearings so that each vendor will have an opportunity to ask questions and all vendors will hear the same response
- *Comments*: With RFP on street without a pre-award hearing, one vendor submitted 5 pages of technical question irrelevant to the solicitation. When we refused to answer all questions and explained irrelevance, we learned that the vendor intended to protest award if not awarded contract. To avoid protest, we pulled RFP.



COTS Acquisition LL Example (1/2)

- *Aspect*: Managerial
- *Object*: Vendor
- *Life-cycle Phase*: Solicitation, acquisition
- *Recommended audience*: Program manager
- *Type of system*: ERP
- *Type of company*: Unknown
- *Number of COTS per project:* 1
- *Type of COTS*: Unknown
- *Type of data*: Qualitative

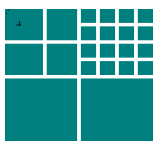


How Do We Share Experiences Across Organizations?

- Through the Best Practices Clearinghouse
  - Promote and assist in the adoption and effective utilization of "best practices"
  - Provide a centralized repository of validated, actionable practice information as well as a gateway to other sources of practices
  - Target the needs of the Department of Defense software acquisition and development community

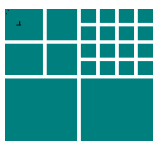
Software Acquisition Manager Needs

- 48 senior SA, SW managers recently surveyed at the SIS Acquisition conference support the use of best practices, but
- Those surveyed can't find best practices
  - Don't exist (need to create a CH)
  - Don't know BPs exist or where they are (need to promote the CH)
  - Not easily accessible (need to make the CH available on the web)
- When best practices are found, information is missing
  - The cost and benefits are not clear (need to make C&B explicit)
  - The effect in specific contexts is not clear (need to make context explicit)
  - Lack of evidence that BPs will work (need to provide empirical evidence)
  - Lack of detail to apply (need to provide general guides, links to specifics)

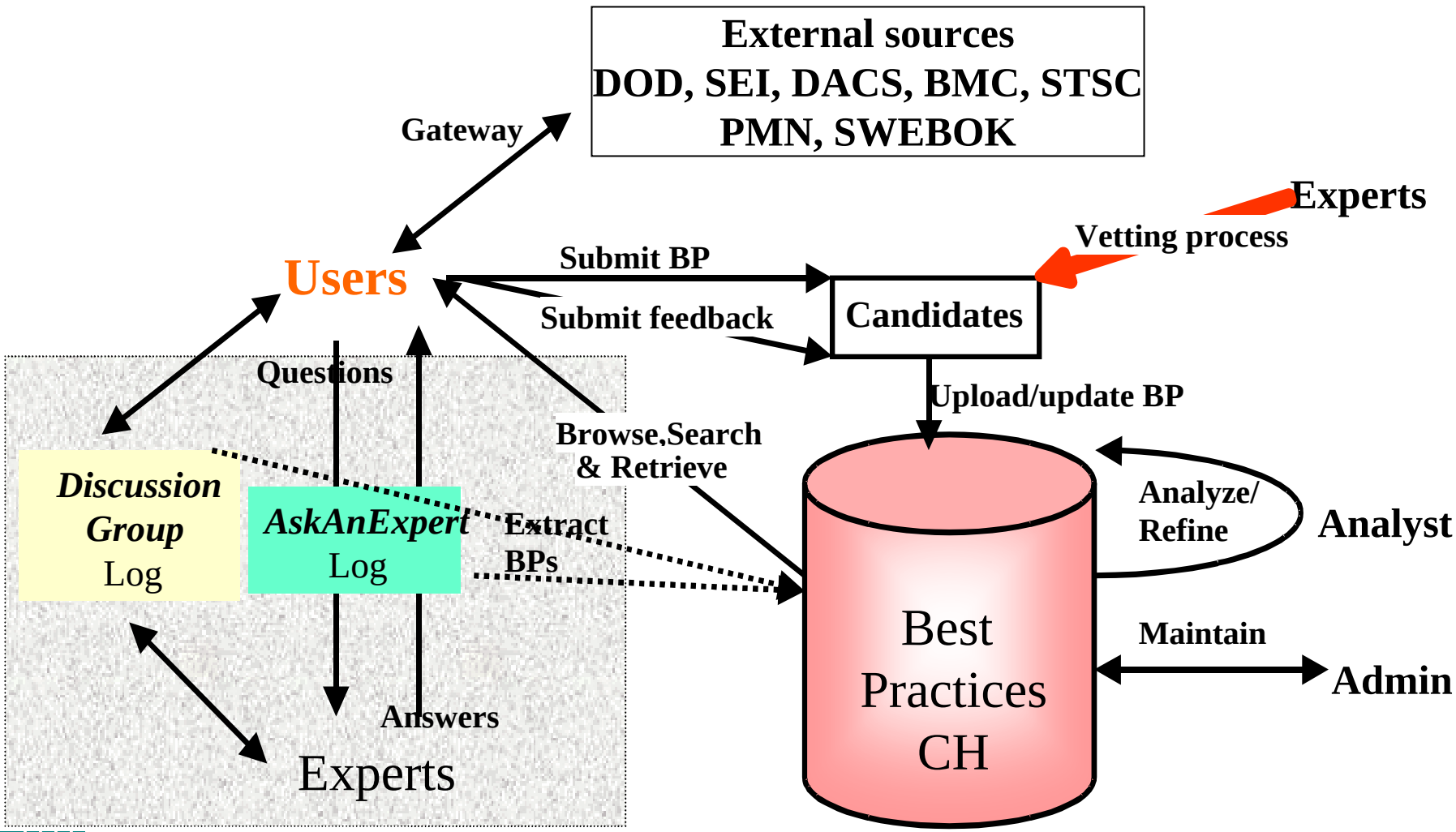


Some Strategies to Meet the Needs

- An experience base
  - User-focused design
  - Empirically based information
  - A set of stories are synthesized into a profile
  - Details of the practice are provided on demand
  - A color code indicates robust practices
- Expert Advice
  - Frequently asked questions
  - Discussion Groups



Clearinghouse Key Concepts

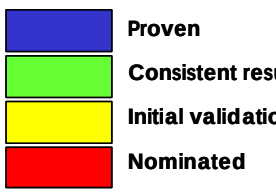


Best Practices Vetting Process

Each cycle allows more experience to be gathered and processed, leading to better characterization of the practice, improved recommendations, and more dependable implementation guidance.

Practice/packaging maturation cycle

Identification	Characterization	Analysis & Synthesis	Validation	Packaging &Dissemination
Inputs: Leads to practices **Activities:** •Collect •Categorize •Filter •Synthesize •Prioritize **Outputs:** Candidate set of practices	**Inputs:** Set of candidate practices and rationale for consideration **Activities:** •Gather/research characteristics about the practice including context (project, etc.), evidence of use, lessons learned •Complete "story" profile **Outputs:** More detailed set of candidate practices with "stories"	**Inputs:** Detailed set of candidate practices **Activities:** •Aggregate stories, create profile of practice •Populate the repository •Identify/define Interrelationships **Outputs:** Single profile for each best practice, associated artifacts, and confidence levels	**Inputs:** Sets of practice data; validation criteria **Activities:** •Check outputs from previous phases •Color Code practices •Approve practices via panel of experts **Outputs:** Validated practices	**Inputs:** Sets of practice data; validation criteria **Activities:** •Packaging •Publishing •Promoting •Providing user help •Discussions **Outputs:** •Repository update •Papers & conference presentations •Course materials/updates

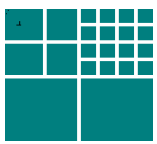


Possible practice validation coding

Objectives for Characterization, Analysis & Synthesis Approach

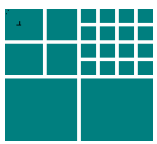
- Populate an empirically-based profile for each practice
- Define profile context and impact attributes
- Create a traceable characterization method
  - Make links to underlying empirical evidence explicit
- Define a repeatable model-based process
  - Enable different people to create profiles consistently
  - Allow for integration of new evidence

Model integration is researched in CeBASE



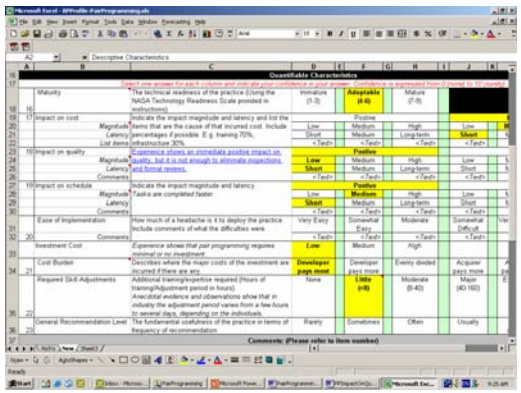
Process for Populating the Repository

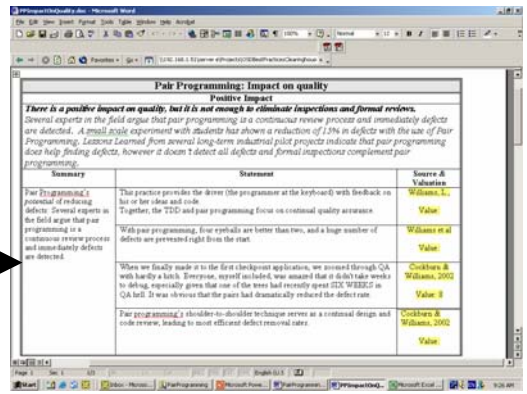
1. Select practice
2. Collect empirical evidence (stories)
3. Organize evidence according to attributes
4. Assign a value to each evidence
5. Characterize each attribute
6. Fill out profile – link to evidence

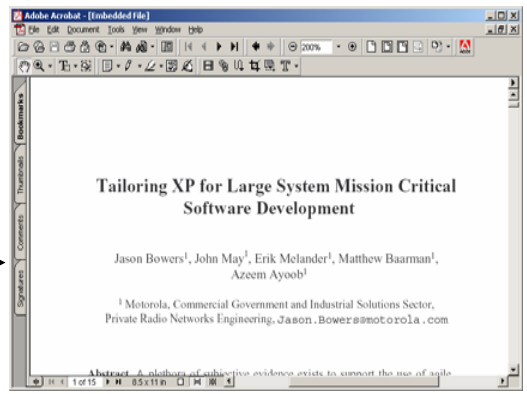


CH Core: Empirically-Based Practices

- **Profile**
  - Attributes, Values, Brief justification, links to
- **Empirical evidence**
  - Justification, Summary, Statement, Source, Valuation, links to
- **Sources**
  - (Full report/paper, Summary/Story)

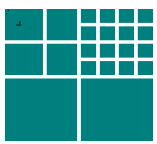






Models Needed

- **Model for judging maturity of Best Practices**
- **Model for Valuating/Weighting Empirical Evidence**
  - **Based on scale, application, and context**



Model for Evaluating Maturity of BP

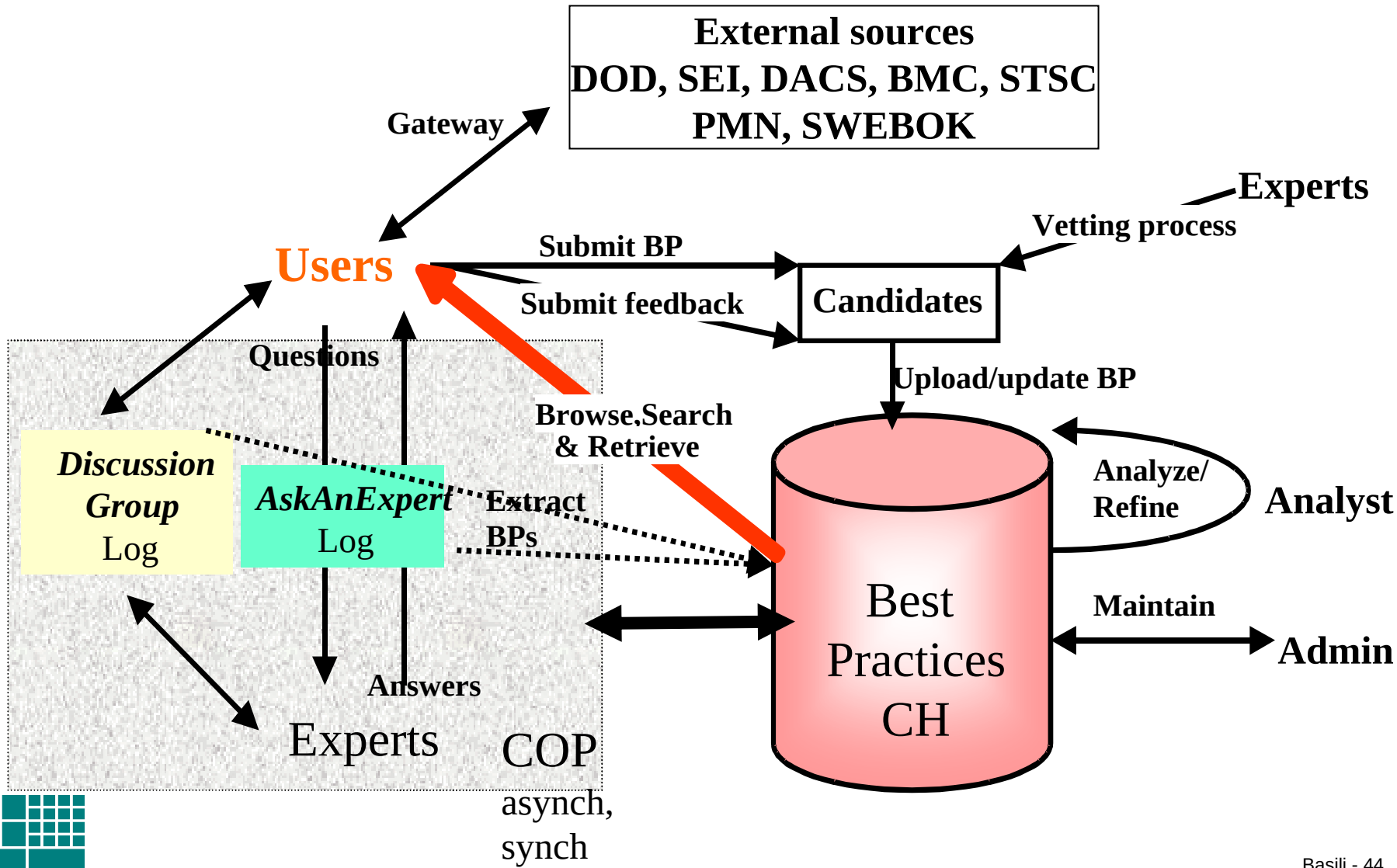
Model for valuation of the maturity[1] of a practice		
Attribute	**Descriptive Value**	**Numerical Value**
How long the practice has been around	Less than 1 year	1
	Less than 5 years	2
	More than 5 years	3
Magnitude of problem to which the practice has been applied (Pick "best" value)	Unclear	0
	Problem that took 40 hours (one person week worth of effort) or less per person to solve	1
	Problem that took more than 40 hours (one person week worth of effort) per person to solve	2
	Problem that took more than 176 hours (one person month worth of effort) per person to solve	3
	Problem that took more than 1760 hours (one person year worth of effort) per person to solve	4

Inspired by NASA Technology Readiness Scale and adapted to Best Practices

Model for Weighting Empirical Evidence

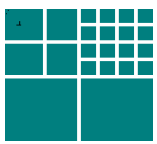
Model for valuation of empirical evidence			
Attribute	**Descriptive Value**		**Numerical Value**
Person(s) who applied the practice	Unclear		1
	Student(s)		1
	Practitioner(s)		2
How the practice was applied	Unclear		1
	One small scale experiment		1
	One large scale experiment		2
	One industrial pilot project		2
	One industrial production project		3
	A series of small scale experiments		2
	A series of large scale experiments		3
	A series of pilot projects		4
	A series of production projects		5
Quality of experience report	No report		0
	Report not published		1
	eWorkshop statement		1
	Unpublished classified internal company report of	Low quality	1
		Medium quality	2
		High quality	3
	Workshop publication		1
	Conference publication		2
	Journal publication		3
Person who conveyed the evidence	Unclear		1
	Student		1
	Researcher		2
	Practitioner		3
	Expert in the field		3

Clearinghouse Key Concepts

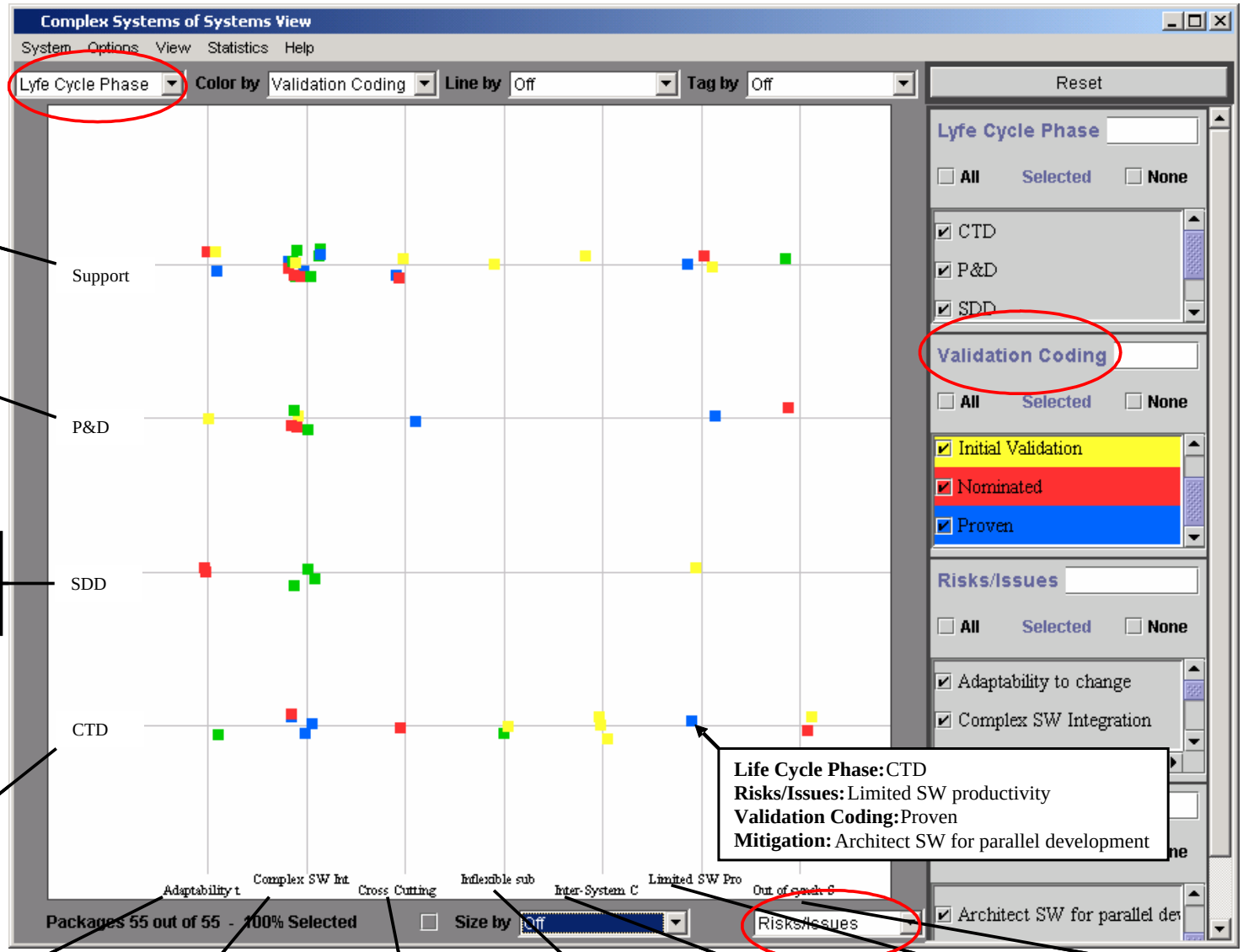


CH Provides Ways for user to look for Practices (Pull)

- User describes the characteristics of his program
  - Example result: Similar programs, recommended BPs
- User describes problems he wants to avoid
  - Example result: Recommended BPs to avoid such problems
- User drills down through some topology
  - Example result: Categories of BPs related to that topology
- User searches the repository on his own
  - Example result: BP Information related to that search



Visual BP Exploration



Support

Production & Deployment

System Development & Demonstration

Concept & Technology Development

Adaptability to change

Complex SW integration

Cross cutting performance trade-offs

Inflexible subcontracting

Inter-systems compound issues

Limited SW productivity

Out of synch SW upgrades

CH provides user with "automate" practices (Push)

- 10 practices to implement
- 10 practices/situations to avoid
- Practice of the day/month
- New and Updated practices

- Potentially based on the profile of the user,project, context
  - Notice that you ask questions, provide relevant information
  - Other users were also interested in…..

Summary

- Build a learning organization to support and improve software acquisition within your own organization and as a shared activity; e.g.,
  - **use and contribute to cebase.org and get involved in the Clearing House experience base**

- Software acquirers need to know what works and under what circumstances
- They need empirical evidence where possible, but any kind of evidence where possible

- We need
  - to continue to collect and share empirical evidence
  - analyze and synthesize the data into models and theories
  - Collaborate to evolve software acquisition processes and models