Evaluating The Software Architecture Competence of Organizations

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

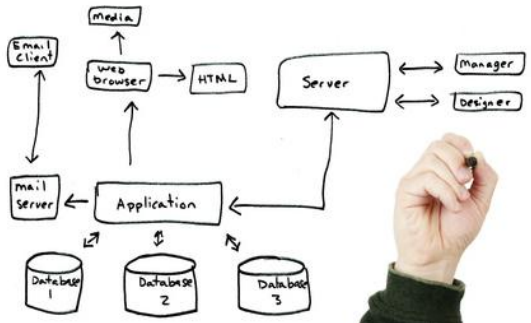
Paul Clements
5 November 2010

Software Engineering Institute | Carnegie Mellon

Improving Architecture Competence - 1

Most of the work in architecture to date has been technical

- Design and creation
- Documentation
- Evaluation and analysis of archit
- Styles and patterns
- Architectural reuse and software product lines
- Architectures for particular domains
- Architectural re-engineering and recovery



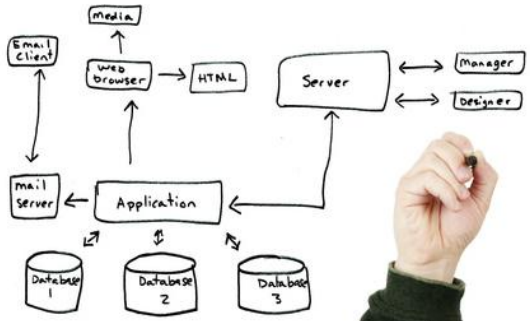
Improving Architecture Competence - 2

But architectures are created by *architects*…

- What does it mean for an architect to be competent?
- How can an architect improve his/her competence?

…working in *organizations*.

- How can we help an organization help their architects do their best work?
- What does it mean for an organization that produces architectures to be competent?
- How can an organization improve its competence in architecture?



Doing architecture well

- If architecture is worth “doing,” surely it’s worth doing well.
- Only if people and organizations are equipped to effectively carry out software-architecture-centric practices will organizations routinely produce high-quality architectures that are aligned with their business goals.
- An organization’s ability to do this well cannot be understood simply through examination of past architectures and measurement of their deficiencies. The root causes of those deficiencies need to be understood.

What Is Competence?

People-oriented or process-oriented definitions:

- Competent: having suitable or sufficient skill, knowledge, experience, etc., for some purpose; properly qualified
  - Random House Unabridged Dictionary©, (accessed via www.dictionary.com, October 19, 2006).
- Competent: Capable of performing an allotted or required function.
  - The Heritage® Stedman's Medical Dictionary, Houghton Mifflin Company, 2002.

Results-oriented definition:

- Competent people are those who can create valuable results without using excessively costly behavior.
  - Thomas F. Gilbert, *Human Competence – Engineering Worthy Performance*. HRD Press, Inc., 1978.

Architecture competence

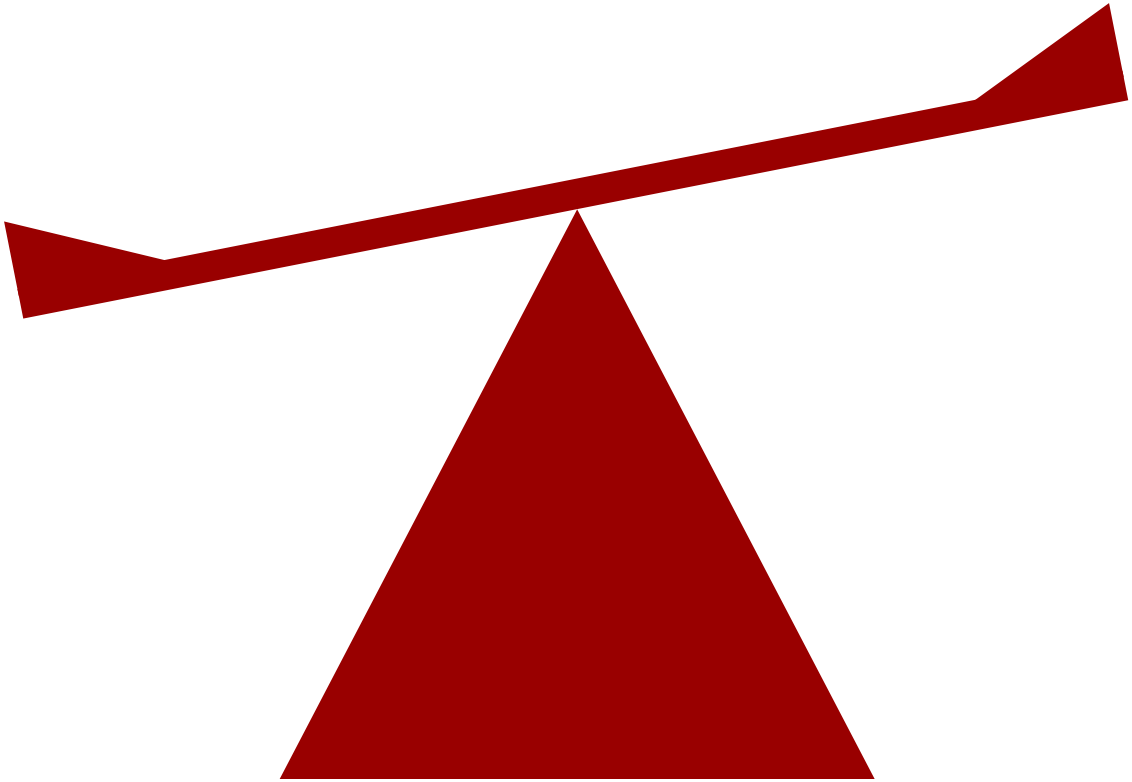
The architecture competence of an organization is the ability of that organization to grow, use, and sustain the skills and knowledge necessary to effectively carry out architecture-centric practices at the individual, team, and organizational levels to produce architectures with acceptable cost that lead to systems aligned with the organization's business goals.

Individual vs. Organizational competence

Individual architects might be very competent, but their organization might not produce good architectures if the organization is not architecturally competent.

Individual architects might not be very competent, but their organization might still produce good architectures.

We need to consider both!



How can we model competence?

- Our research has uncovered four distinct models of organizational and human behavior that can be applied to software architecture.
- They can help us evaluate and improve how individuals and organizations produce good architectures.

DSK model

a model based on the duties, skills, and knowledge of software architects

Human Performance model

based on an organization's performance

Organizational coordination model

based upon how organizations internally interact and coordinate

Organizational learning model

how do organizations acquire, internalize, and use information

Duties, Skills, and Knowledge model

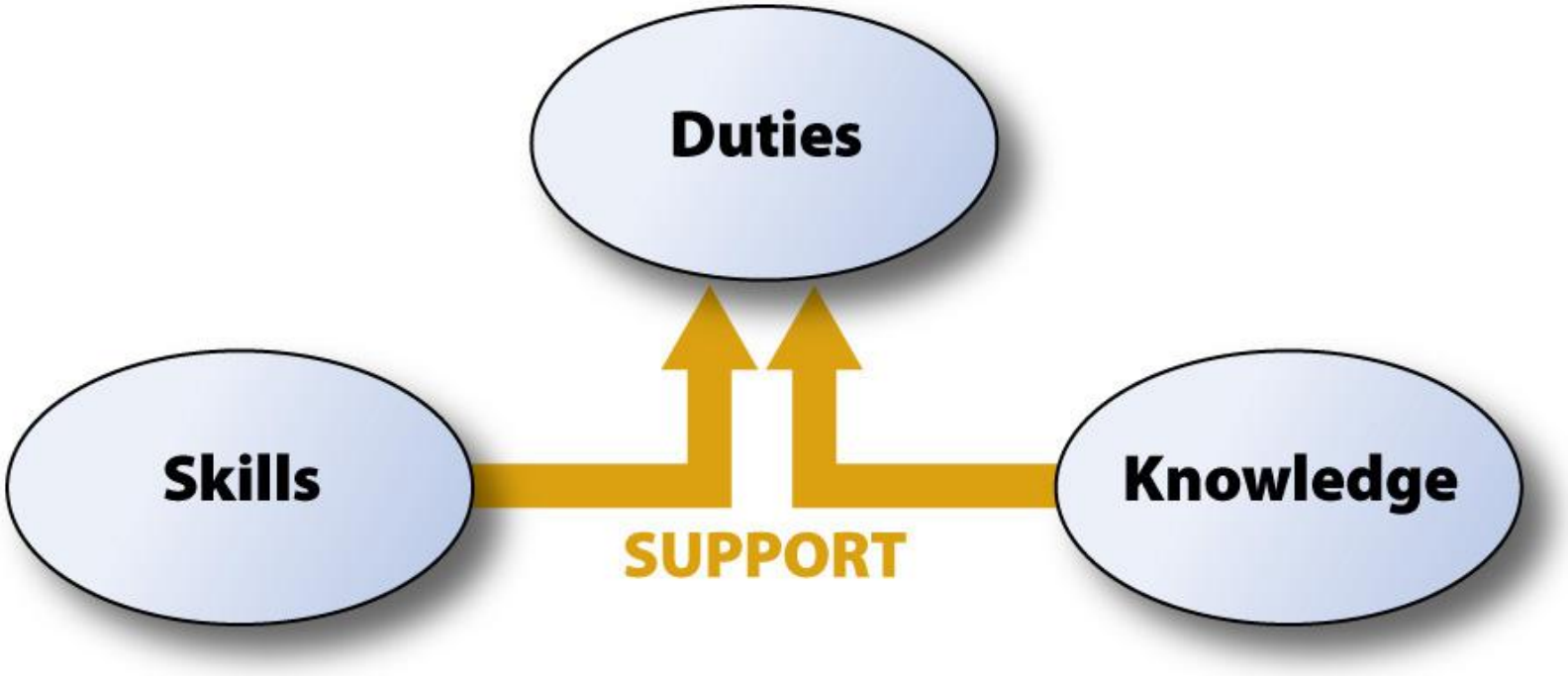
- Architects and architecture-producing organizations help us understand the tasks necessary to architecting.
- We can catalog what architects and organizations do and know
- We can build measures for how well they do and know it
- We can craft improvement strategies for their duties, skills, and knowledge.
- We surveyed approximately 200 sources of information targeted to professional architects – books, web sites, blogs, position descriptions, and more, and polled practicing architects.

Duties, skills, and knowledge

- Individual architects have to
  - *do* things (duties)
  - *be good* at things (skills)
  - *know* things (knowledge)
- We can measure these things. Examples:
  - "design the architecture" is a duty,
  - "ability to think abstractly" is a skill, and
  - "patterns, styles, and tactics" is a part of the body of knowledge.

Duties, skills, and knowledge

Skills and knowledge help an architect perform his or her duties.



Survey results: Duties of architects

- Architecting
  - Create an architecture
  - Evaluate and analyze an architecture
  - Document an architecture
  - Work with and transform existing system(s)
  - Other architecting duties
- Duties concerned with life-cycle phases other than architecture
  - Requirements
  - Coding
  - Testing
  - Future technologies
  - Tools and technology selection
- Interacting with stakeholders
  - Interact with Clients
  - Interact with Developers
  - Interact with other stakeholders
- Management
  - Project management
  - People management
  - Support for management
- Organization and business related duties
- Leadership and team building

Survey results: Skills of architects

- Communication skills
  - Outward
  - Communication skills in general
  - Inward (listening)
- Work skills
  - Leadership
  - Effectively managing workload
  - Skills to excel in corporate environment
  - Skills for handling information
- Interpersonal skills
  - Within team
  - With other people
- Personal skills
  - Personal qualities
  - Skills for handling unknown
  - Skills for handling unexpected
  - Learning

Survey results: Knowledge of architects

- Computer science knowledge
  - Knowledge of architecture concepts
  - Knowledge of software engineering
  - Design knowledge
  - Programming knowledge
- Knowledge of technologies and platforms
  - Specific technologies and platforms
  - General knowledge of technologies and platforms
- Knowledge about the organization's context and management
  - Domain knowledge
  - Industry knowledge
  - Enterprise knowledge
  - Leadership and management techniques and experience

Improvement strategies

- If you want to improve your individual architectural competence, you should:
  - **Gain experience carrying out the duties:** Apprenticeship is a productive path to achieving experience.
  - **Improve your nontechnical skills:** This dimension of improvement involves taking professional development courses, for example, in leadership or time management.
  - **Master the body of knowledge:** Taking courses, becoming certified, reading books and journals, visiting websites and portals, reading blogs, attending architecture-oriented conferences, joining a professional societies, and meeting with other architects are all useful ways to improve knowledge.

Architecture duties of an organization -1

- Hire talented architects.
- Establish a career track for architects.
- Make the position of architect highly regarded through visibility, reward, and prestige.
- Establish a clear statement of responsibilities and authority for architects.
- Establish a mentoring program for architects.
- Establish an architecture training and education program.
- Track how architects spend their time.
- Establish an architect certification program.
- Have architects receive external architect certifications.
- Measure architects' performance.
- Establish a forum for architects to communicate and share information and experience.
- Establish a repository of reusable architectures and architecture-based artifacts.
- Develop reusable reference architectures.

Architecture duties of an organization -2

- Establish organization-wide architecture practices.
- Establish an architecture review board.
- Measure the quality of architectures produced.
- Provide a centralized resource to analyze and help with architecture tools.
- Hold an organization-wide architecture conference.
- Initiate software process improvement or software quality improvement practices.
- Have architects join professional organizations.
- Bring in outside expert consultants on architecture.
- Include architecture milestones in project plans.
- Have architects provide input into product definition.
- Have architects advise on the development team structure.
- Give architects influence throughout the entire project life cycle.
- Reward/penalize architects based on project success.

Human Performance Engineering model

- Based on the human performance engineering work of Thomas Gilbert (1927-1995).
- Competent individuals in any profession are the ones who produce the most valuable results at a reasonable cost.
- This model relies on measuring the worth and cost of the outputs of architecture efforts, finding areas where that ratio can be improved, and crafting improvement strategies based on environmental and behavioral factors.

Human Performance Engineering

- Gilbert strongly advocated measuring performance, not knowledge or behavior or motivation or skills or....
  - "If I want to know if people are competent, I have to observe how they behave, don't I? My answer to such questions is a firm 'No!'"
  - Worth = Value of result / Cost to achieve it.
  - W = V / C
  - Egyptian pyramids are "monuments to useless knowledge"
  - Arabic alphabet was a much more "worthy" achievement

- Thomas F. Gilbert, *Human Competence – Engineering Worthy Performance.* HRD Press, Inc., 1996 "Tribute Edition." Book originally published 1978.

Competence model based on Human Performance Engineering

1. Identify what “worthy performance” means for each task involved in architecture.
2. Identify what costs are involved for each task involved in architecture.
3. Identify performance-related measures of each
4. Identify an exemplary measure – the best we could hope for – of each
5. Build an assessment instrument that will gather measurements in an organization, compare them to exemplar in each category, and identify best potential areas for performance improvement.
6. Suggest specific improvement strategies.

Measuring Worthy Performance: W = V / C

Performance (or the worth of the result) has the following dimensions or "requirements":

Quality

- **Accuracy**: Degree to which accomplishment matches a model, without errors of omission or commission.
- **Class**: Comparative superiority of an accomplishment beyond mere accuracy. Possible measures include market value, judgment points (as for show dogs), physical measures (such as number of mfg. flaws), opinion ratings (Oscars, "MVP")
- **Novelty**: An engine that gets 100mpg is novel. For artistic novelty we probably resort to judgmental points or opinion rating.

Quantity (or Productivity)

- **Rate**: Applies when bulk is time-sensitive; pieces produced per hour; time to completion
- **Timeliness**: Time, not bulk, is key: letter mailed by sundown, Cinderella home by midnight
- **Volume**: Bulk is important, but not time-sensitive. "How many fish did you catch?"

Cost

- **Labor** (behavior repositories): Includes direct overhead, benefits, wages, insurance, taxes
- **Material** (environmental support): Includes supplies, tools, space, energy
- **Management**: Supervision, its supports, public taxes, internal allocations of admin costs.

Example of applying Steps 1, 2, and 3

TASK: CREATING THE ARCHITECTURE

Quality

- Accuracy: Is the architecture the right one for the task at hand?
  Measure: Total cost of changes (= revisiting decisions) to the architecture during development [accounts for lots of small changes as well as number of big ones]. Cost means cost of making change in the architecture AND cost of downstream resulting changes. Measure as % of total cost of system, to (a) find exemplar; and (b) compare systems.
  Comments: Doesn't help for changes that were too expensive to address. Alternative measure is to capture satisfaction of important requirements (e.g., QA scenarios) and test fulfillment (e.g., ATAM-style walkthroughs).
- Class: How many architectures were influenced by this one? Whole thing? Pieces? Ideas?
- Novelty: N/A

Quantity (or Productivity)

- Rate: Time to completion.
- Timeliness: Deadlines met.
- Volume: Size of system.

Cost

- Labor (behavior repositories): Count staff hours for architects
- Material (environmental support): Staff hours for consultants; costs of tools used by architect. Travel costs. Communication costs.
- Management: Count staff hours for managers

Possible dimensions of architecture value

- Predictability in cost and schedule and quality
  - Measure: Variance between predictions and actual
  - Hypothesis: Architecture practices lead to lower variances

- Ability to achieve system that meets its requirements (which presumably reflect business goals)
  - Measure: Does it or doesn't it? What percentage of requirements are met? What percentage of high-priority requirements are met?
  - Hypothesis: Architecture practices lead to higher achievement

An assessment instrument: Gilbert's Performance Audit

First measurements (Institution's accomplishments)

Second measurements (Job accomplishments)

Third measurements (Task accomplishments)

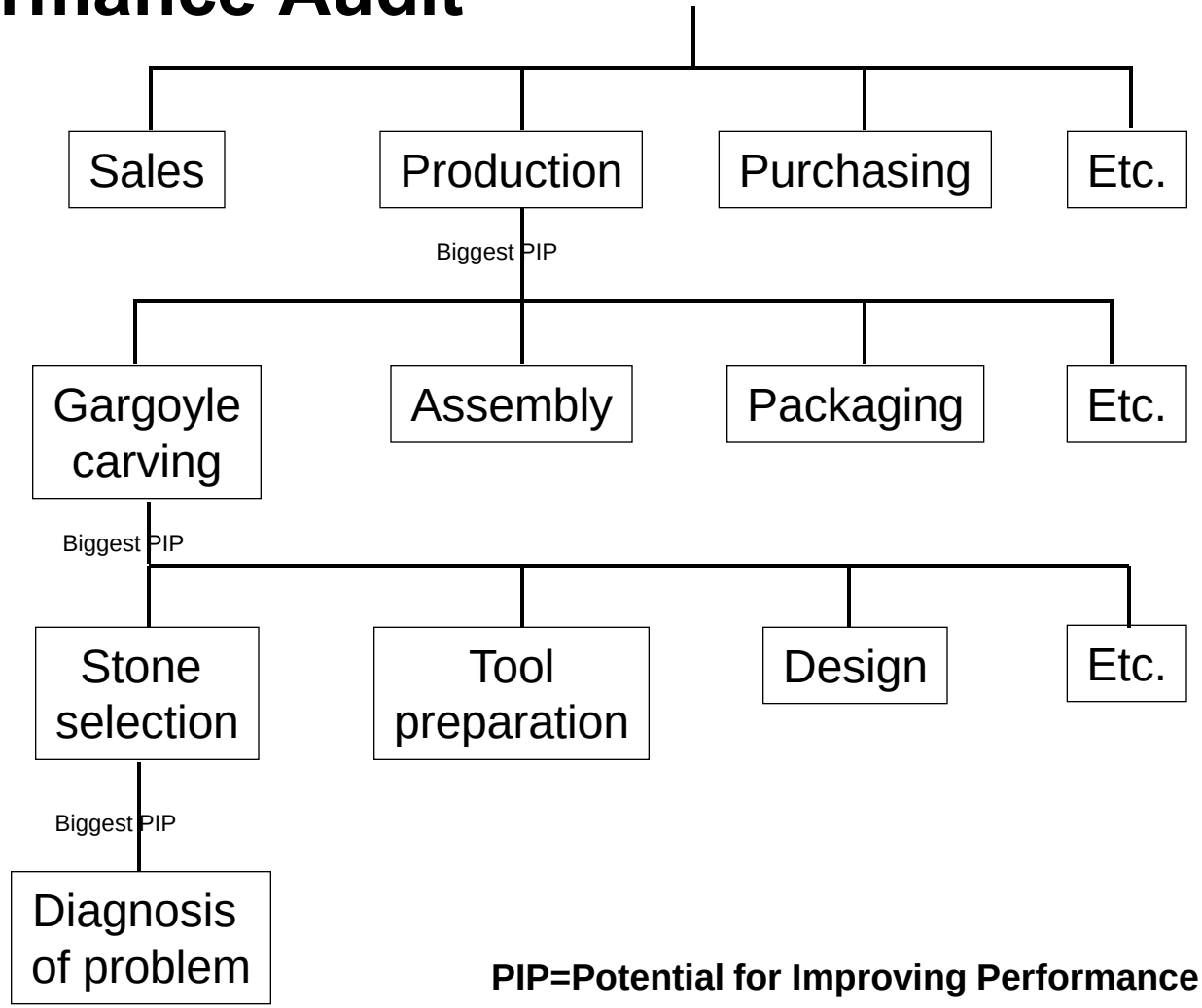


Figure 2.2, p. 60

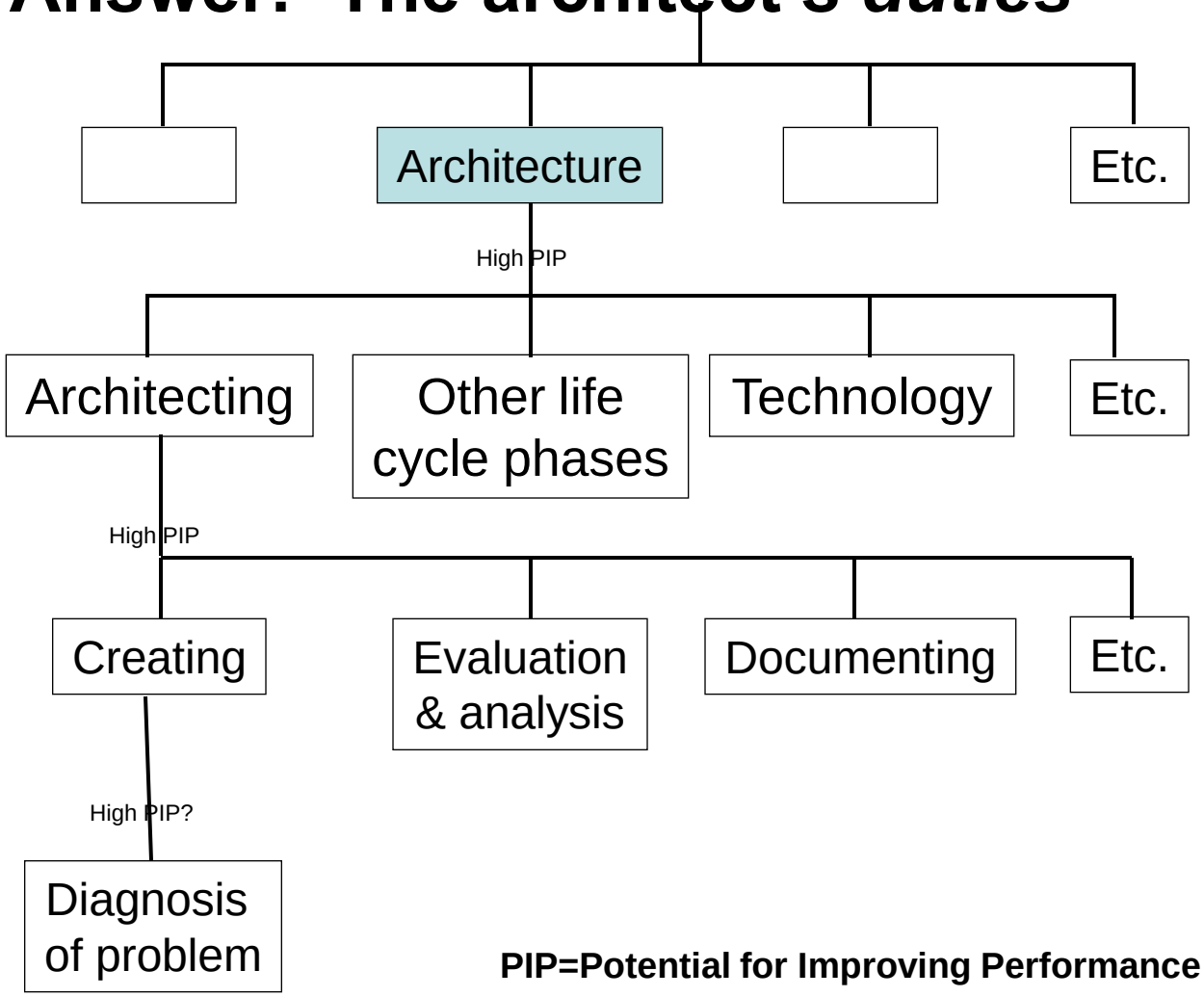
PIP=Potential for Improving Performance

What tasks go in the boxes for an architecture organization? Answer: The architect's *duties*

First measurements (Institution's accomplishments)

Second measurements (Job accomplishments)

Third measurements (Task accomplishments)



PIP=Potential for Improving Performance

Organizational Coordination model -1

- Research is related to multisite development of software.
- The architecture for the product requires teams to coordinate during the realization or refinement of various architectural decisions.
- The organizational structure, practices, and tool environment of the teams allow for particular types of coordination.
- The coordination model of competence will compare the requirements for coordination that the architecture induces with the bandwidth for coordination supported by the organizational structure, practices, and tool environment.

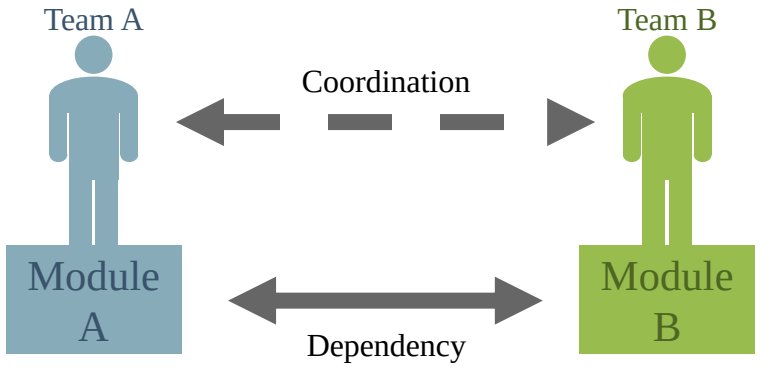
Organizational Coordination model -2

Teams developing a product must cooperate.

This cooperation has, as its external manifestation, the coordination of activities.

The goal of this model is to help us understand:

- what coordination activities are caused by particular architectural decisions, and
- the effectiveness of specific coordination mechanisms



Coordination Mechanisms

- Co-locating representatives of one team with another team
- Using tools such as telephone or computer supported chat for synchronous coordination
- Using tools such as e-mail, discussion boards, or documents for asynchronous coordination
- Maintaining system knowledge within a core team (the architecture team) by requiring that all communication be mediated by that team

Organizational Learning model -1

- Organizations, not just individuals, can learn.
- Organizational learning is a change in the organization that occurs as a function of experience. This change can occur in the organization's cognitions or knowledge, its routines or practices, or its performance.
- To measure organizational learning, we can
    1. measure knowledge directly though questionnaires, interviews, and verbal protocols;
    2. treat changes in routines and practices as indicators of changes in knowledge; or
    3. view changes in organizational performance indicators associated with experience as reflecting changes in knowledge

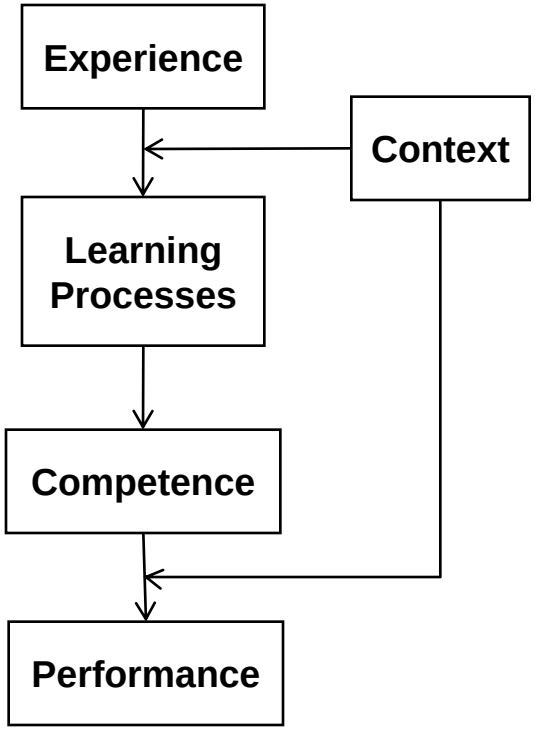
Organizational Learning

Organizational learning is defined as a change in an organization that occurs as a function of experience.

Manifested as:

- changes to the organization's knowledge,
- its routines, or
- its performance

Learning processes transform experience into knowledge, moderated by context.



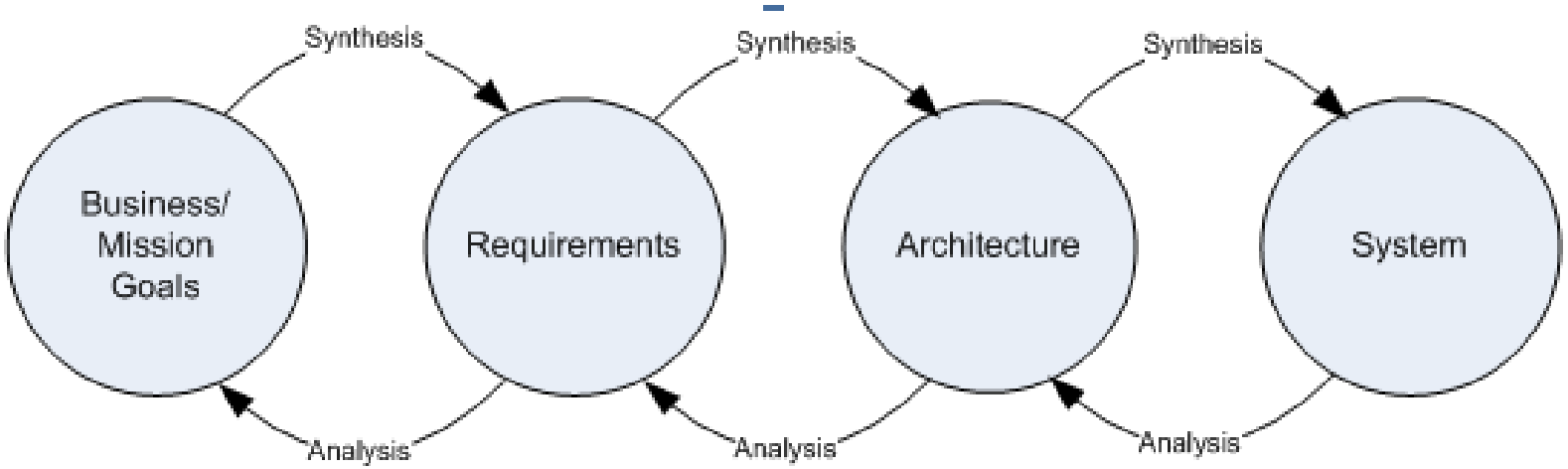
Implications for Architecture Competence

An architecturally competent organization will understand:

- *the organizational learning opportunities presented by the **experiences** in performing architecture-centric practices.*
- *which types of **learning processes** are best suited for different types of experiences.*
- *how various types of learning **context** affect the transformation of experience into **knowledge stock**.*

An Activity-Based View – Applying the models to Architecture-centric Engineering

The SEI's own work on architecture is based on a number of activities we believe are central to architecture-based development.



Competence is the ability to do these things well. Competent organizations need to "grow, use and sustain the skills and knowledge necessary to effectively carry out [these] architecture-centric practices at the individual, team, and organizational levels."

A Framework Informed by Four Models

- The **activity model** guides us on what architecture-specific activities to probe.

- The **four models** guide us on what questions to ask about each activity:
  - Gilbert: How do you measure the value? What is the cost?
  - DSK: What do your architects do and know? E.g., are they given time to mentor others?
  - OL: How is information internalized and nurtured?
  - OC: What coordination mechanisms exist to let parts of the organization carry out these activities effectively?

- The **definition** tells us to ask not only about what the organization does today, but what it does to *grow* and *sustain* its ability in the future.

A framework for organizational competence in architecture

Software Engineering Practice Areas

- Quality Attribute Elicitation Practices
- Tools and Technology Selection
- Modeling and Prototyping Practices
- Architecture Design Practices
- Architecture Description Practices
- Architecture Evaluation Practices
- System Implementation Practices
- Software Design Practices (Design conforms to architecture)
- Software Coding Practices (Code conforms to design and architecture)
- Software Verification Practices
- Proving Properties of the Software
- Software Testing
- Architecture Reconstruction Practices

A framework for organizational competence in architecture

Technical Management Practice Areas

- Business/Mission Goals
- Setting goals
- Measuring achievement of organization's goals
- Performance-based compensation
- Product/System Definition
- Setting functional requirements
- Allocating Resources
- Setting architect's workload and schedule
- Funding stakeholder involvement
- Project Management (Project plan structure aligned with architecture structure. Adequate time planned for architecture evaluation.)
- Process Discipline
- Establish organization-wide architecture
- Process monitoring and improvement
- Promote reuse
- Collaboration with manager
- Architects advise managers
- Architects support managers

A framework for organizational competence in architecture

Organizational Management Practice Areas

- Hire talented architects
- Establish a career track for Architects
- Leadership roles for architects
- Succession planning
- Professional Development Practices
- Ongoing training
- Creating and sustaining internal community of architects
- Support participation in external communities
- Organizational Planning Practices
- Technology Planning and Forecasting

Using the framework

To assess an organization for competence, we use the framework to ask questions about an organization's practices.

We can also ask about recent architecture successes and failures, and investigate the causes of each.

Process and outputs

- Pre-interview questionnaires – architects and stakeholders
- In-person interviews – approx. 3 days on-site
- Evaluation of past architectures
- Assessment Report – Competence at organization, team, and individual level
- Improvement recommendations based on assessment results – tied to underlying competence models

Summary

If architecture is worth practicing, it's worth doing well.

"Doing it well" requires competence from organizations as well as individuals.

We can codify what it means for an organization to be competent.

Architecture competence results in an architecture in conformance with a performing system that meets the organization's business goals for that system.

Competence applies to building present systems, but also in sustaining that capability as people come and go, as technology evolves, and as the competitive climate changes.

This continues the maturation of architecture into a repeatable discipline.

Questions?

Contact information:

Paul C. Clements
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
clements@sei.cmu.edu

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.