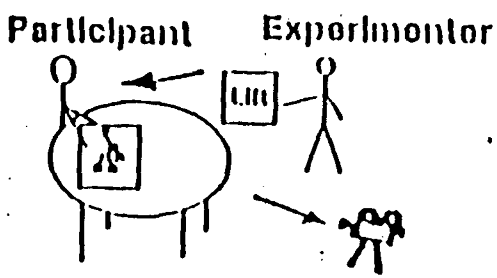
Empirical Studies of Software Design: Implications for SEEs

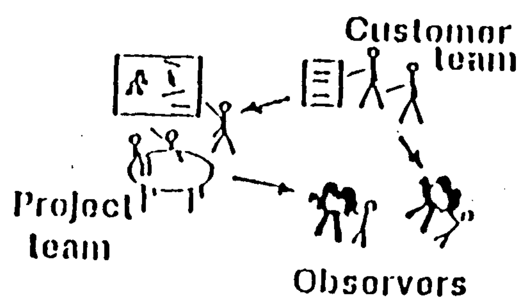
Herb Krasner
Manager, Software Process Research
Lockheed Software Technology Center
Austin, Texas

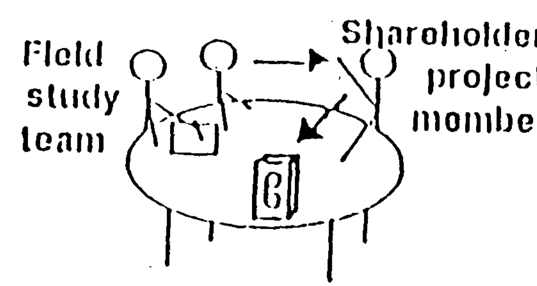
Lockheed
Missiles & Space Company, Inc.
Software Technology Center #11758

Empirical Research on the Software Process

Individual level	Team level	Project level
LIFT experiment	**Object server exp.**	**Field study**
8 experienced programmers designing the control structure for a set of elevators during an intense 2 hr. session	Videotaped team meetings from a 7 mo. effort to design and build a tool to support object oriented programming	Detailed interviews with key members of 18 large development projects to model their decision-making and communication process







Results of the Field Study

- Observations about commonality/difference of projects
- Identification of five areas of organizational breakdown (within that sixteen specific problems)
- Implications for process modeling
- Mapping of problems onto lower-level phenomena

"You need to understand, this project isn't the way we develop software at our company."

Characteristics of Projects Studied

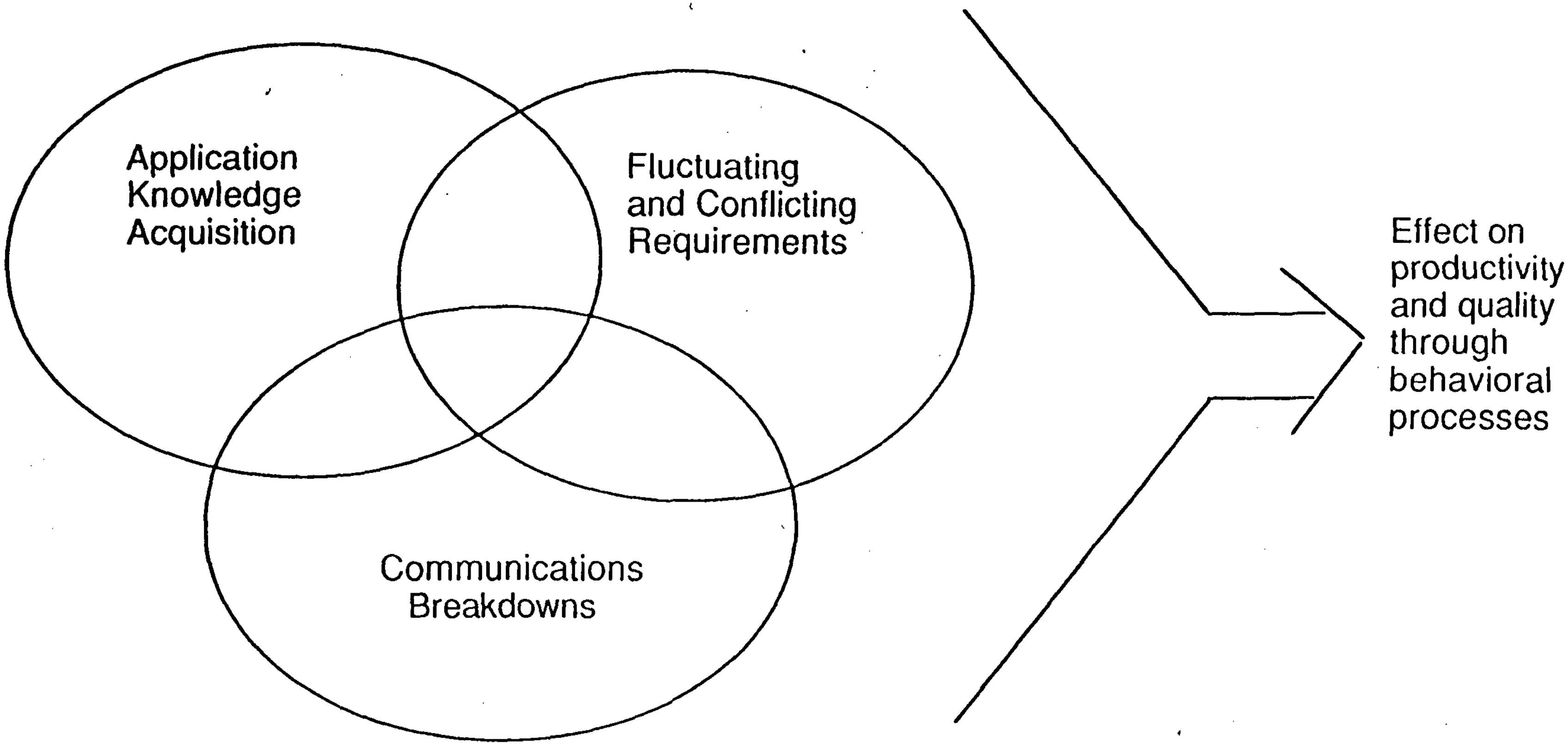
Project	Stage of Life Cycle	KLOC	Characteristics: Real time	Dist. Sys.	Emb. Sys.	Gov.	Application
1	Terminated	-					Support Software
2	Development	24	✔		✔		Radio Control
3	Development	50	✔	✔	✔		Process Control
4	Development	50	✔				Operating System
5	Design	70					CAD
6	Development	130				✔	CAD
7	Development	150+	✔		✔	✔	Avionics
8	Maintenance	194	✔			✔	C^3
9	Development						Compiler
10	Maintenance	250					Run-time Library
11	Development	350+					Compiler
12	Maintenance	400					Transaction Proc.
13	Design	500	✔	✔			Telephony
14	Maintenance	725					Operating System
15	Development	1000	✔	✔			Telephony
16	Maintenance	50k+	✔	✔	✔	✔	Radar, C^3
17	Requirements	100k+	✔	✔	✔	✔	C^3, Life Support

Summary of Results from MCC Field Study*

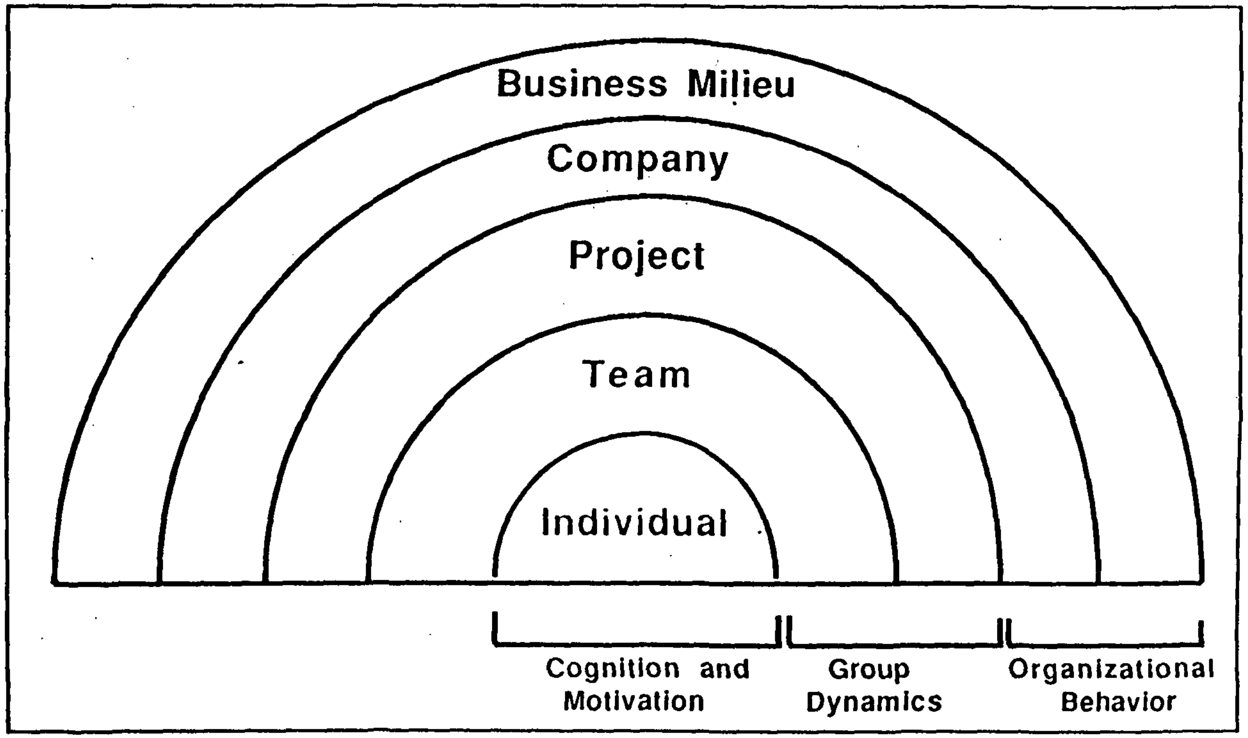
- Analysis of three significant problems
- Layered behaviorial model of software processes
- Conclusions and implications

* Paper appearing in this months CACM

Analysis of Three Significant Problems in Software Design for Large Systems



Layered Behavorial Model of Software Processes



Implications of Field Study Results

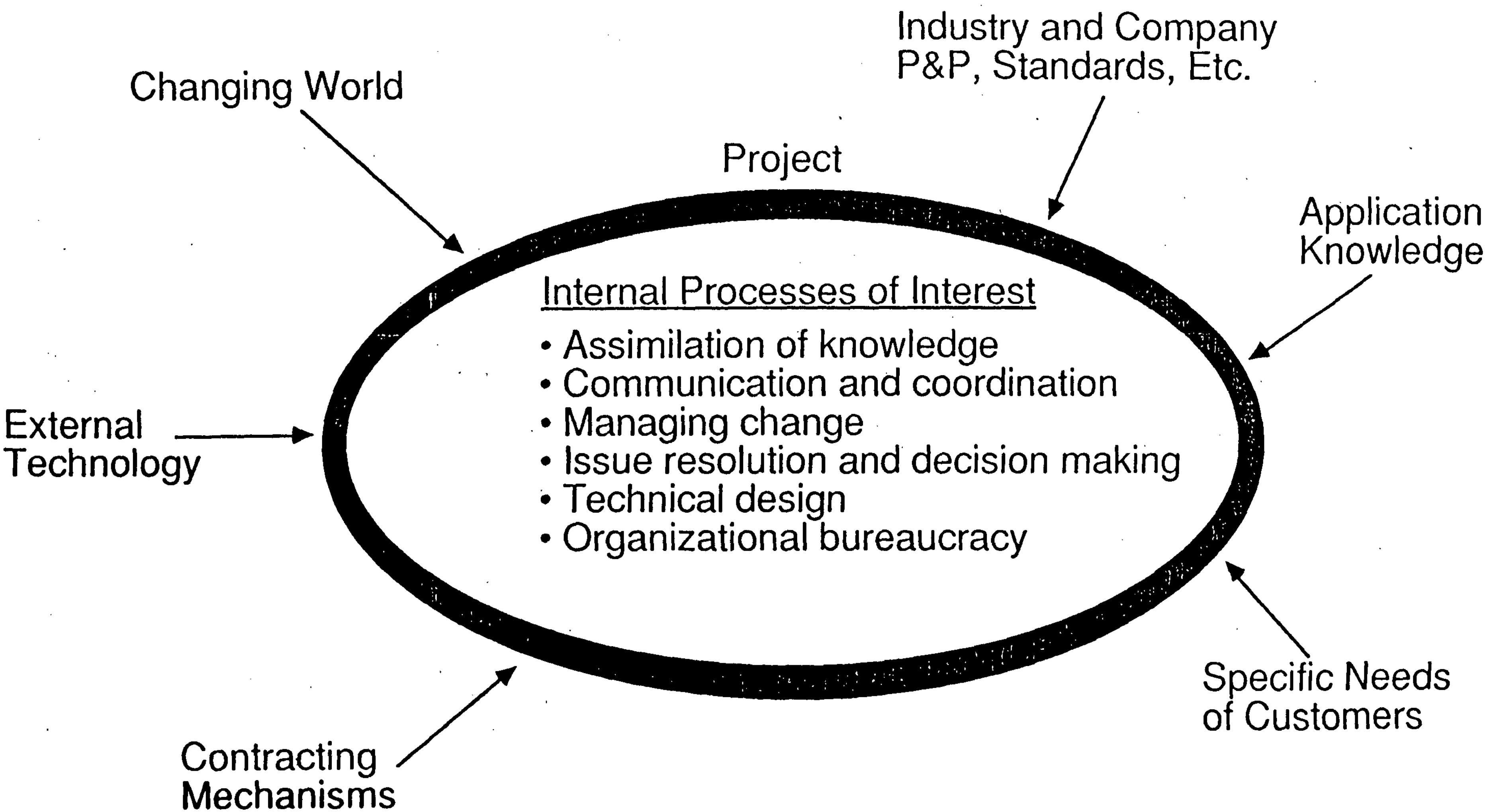
- For Software Technology
  - Environment support needed for:
    - = Knowledge integration
    - = Change facilitation
    - = Broad communication and coordination
  - Beginnings of an empirical model to measure improvement for a tool/practice

- For Project Management
  - Expertise is the primary determinant, new ways of effectively organizing should be pursued
  - Key role players identified and described:
    superconceptualizer, diagnostician, gatekeeper, boundary spanner
  - Coordination by shared model of process, product

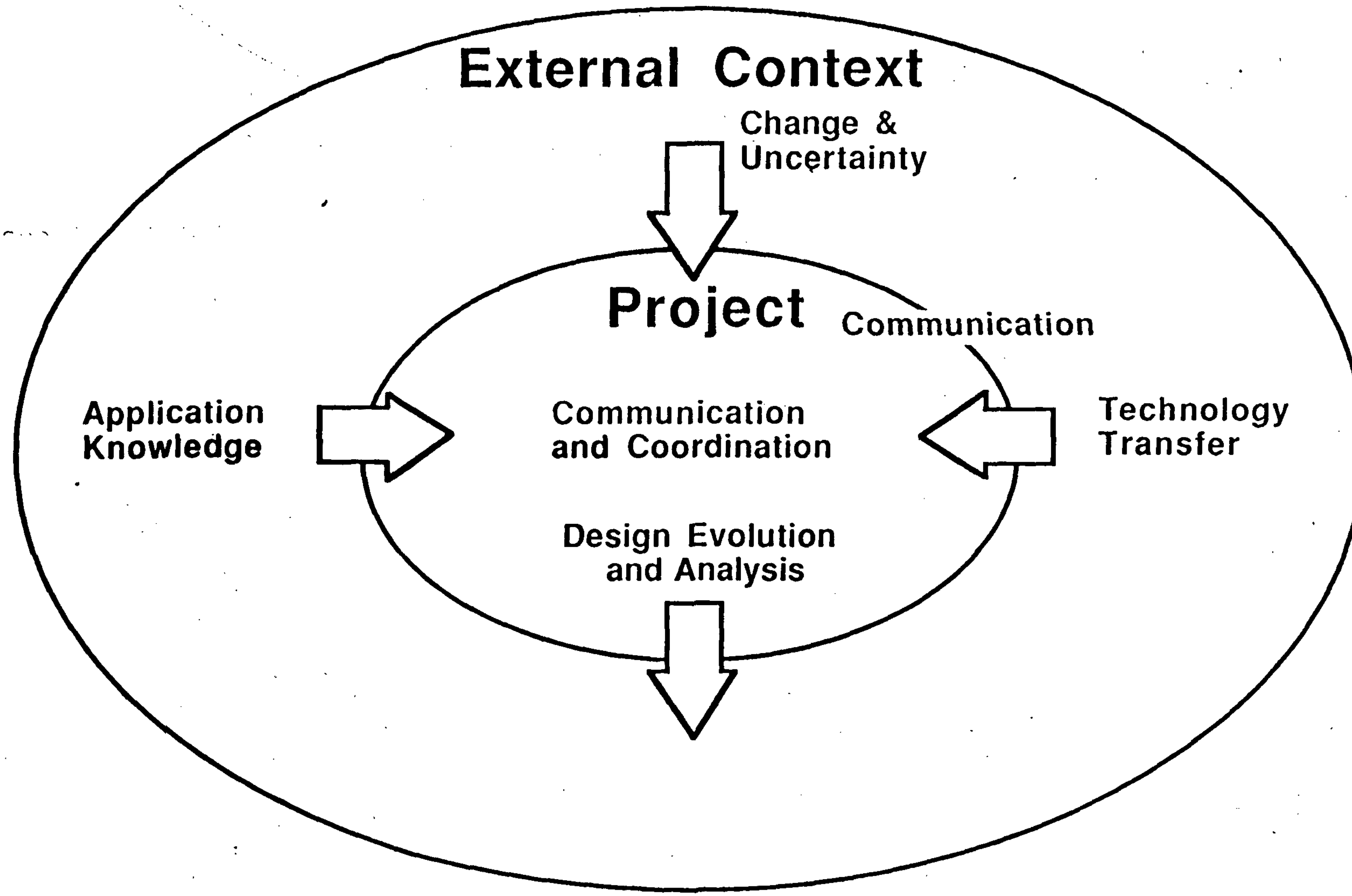
- For Software Process Models
  - Difference between prescriptive and actual processes
  - Current process models do not reflect:
    learning, technical communication, requirements negotiation, and customer interaction
  - Framework for an "ideal" process model emerging

- For Further Empirical Research on Professional Software Engineering
  - Much more to do
  - Focus on "variation" and its effect on the difference in productivity and quality outcomes among people, situations, and their interaction

The Software Project as an Ecological System



Five Crucial Problem Areas in Large Software Projects*



* see STP-390-86p

Overall Conclusion

The Greatest Leverage Is in Supporting the Intersection of:

The Technical Task

- Assessing customer needs
- Assimilating application knowledge
- Negotiating requirements, technology, and resources
- Identifying and exploring design assumptions/alternatives
- Decomposing and recomposing functionality
- Defining and controlling component interfaces

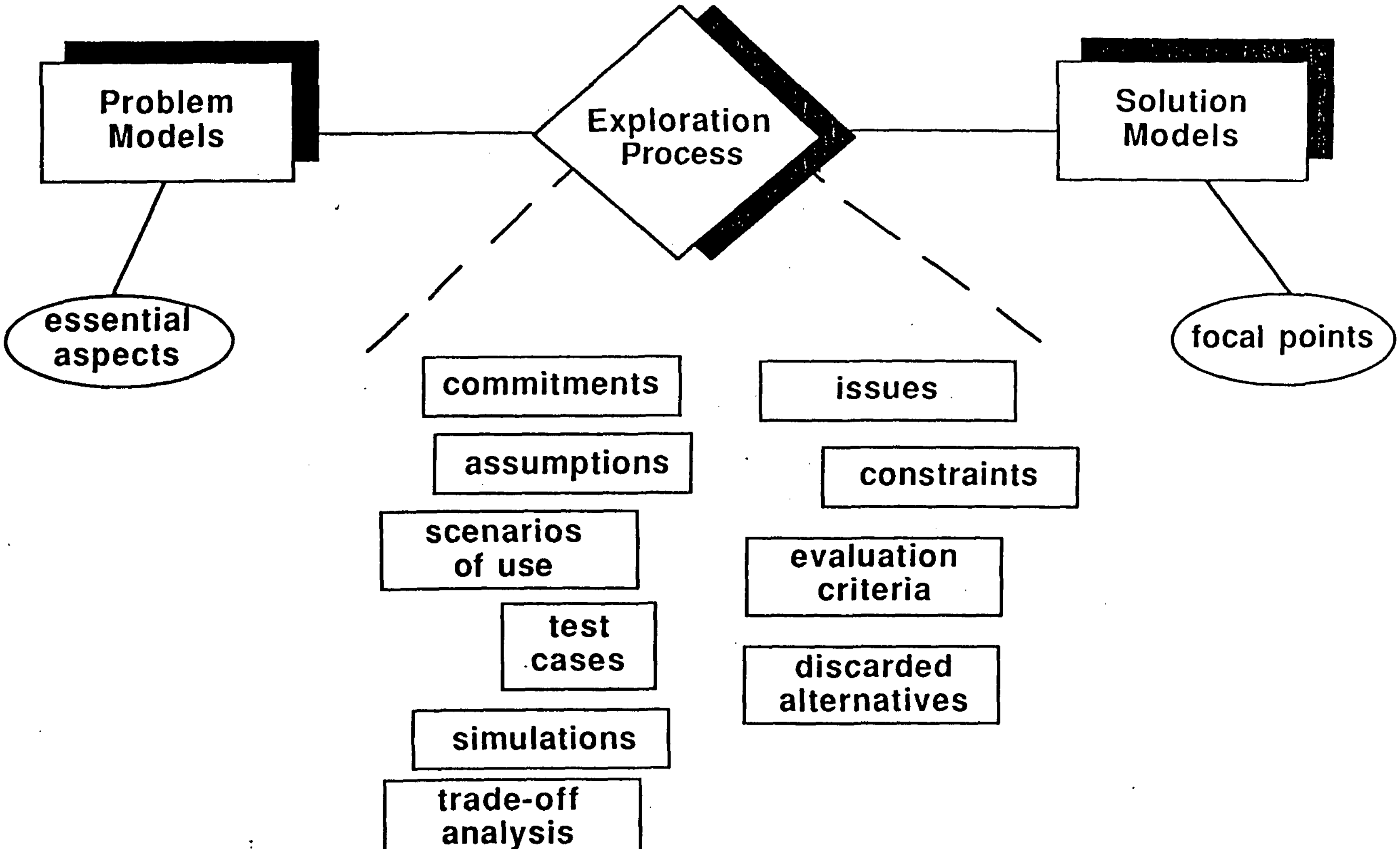
The ManagementTask

- Strategically managing system features and attributes
- Assessing and controlling risks
- Ensuring developers work from the same models

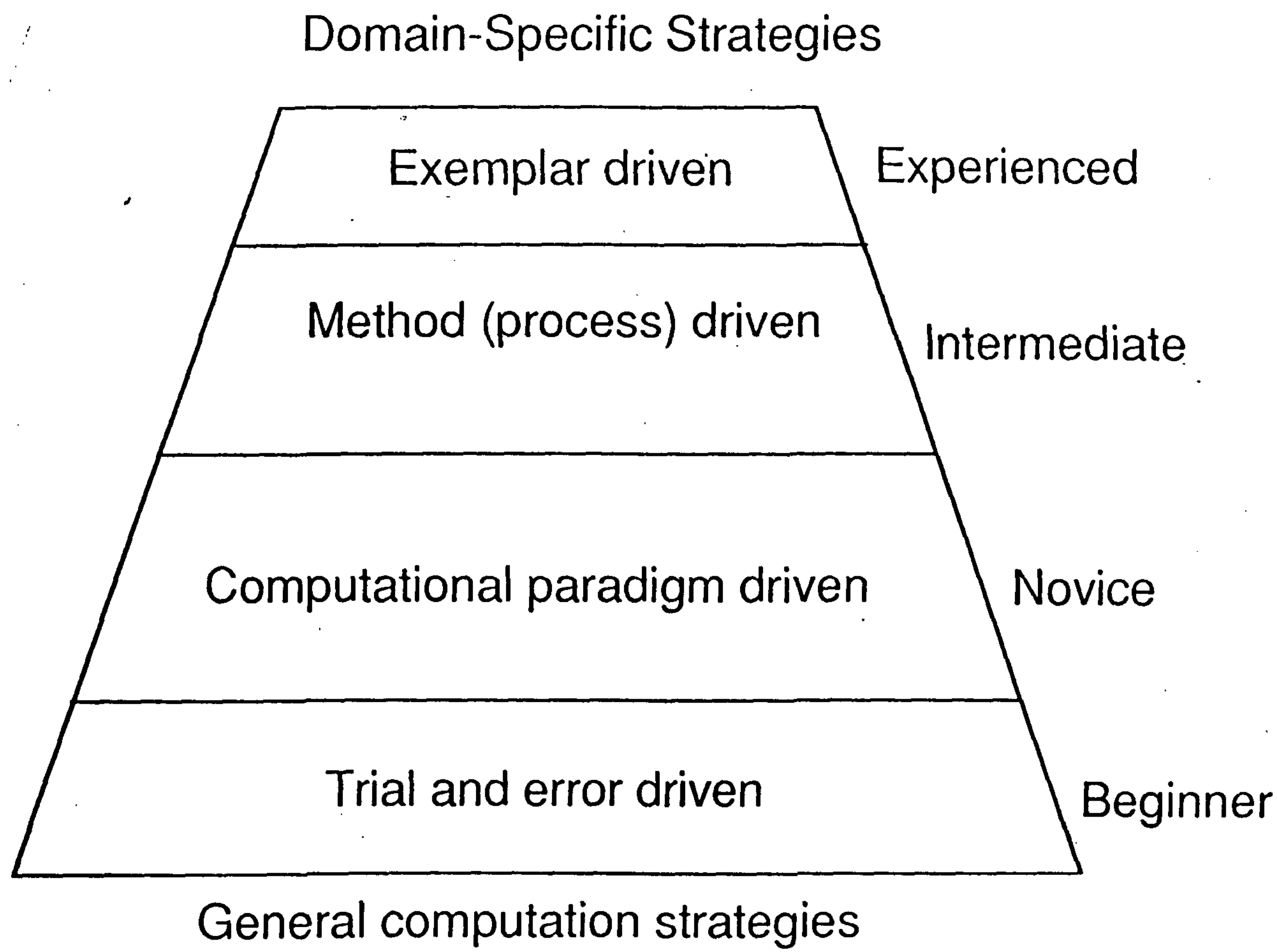
Results of the "LIFT" Study

- Observations on relative effort distribution
- Observations about individual differences
- Identification of six process breakdowns
- A cognitive model of design problem solving

Information Model of Design Exploration



Individual Differences in Software Design Strategies



Results of the Team Design Study*

- Identification of <u>conflict behavior</u> as key to achieving shared models
- Observations on the limitations of "documents"
- Observation of ombudsman to facilitate communication between customer and design teams
- Observations on the effect of *midnight prototype* creation
- Videotape identified as history capture mechanism

* being completed at U.T. - D. Walz, 1988

Future SSEs Should Contain Facilities For

1) Focus on Productivity and Quality
   - Statistical QC
   - Reduce waste and redundancy
   - Institutionalized reuse process yields component parts (via standards)

2) Process Engineering
   - Introduction of good practices, tools, etc.
   - Process definition, tailoring, monitoring, analysis, and improvement
   - Embodiment in education programs

3) Process Efficiency through Teamwork and Communication
   - Revocation of Brook's Law
   - High performance teamwork
   - "Groupware"

4) Flexible Organization Evolution
   - Coordinated technology, policy and organizational structure around process management concerns
   - Committment to improve (facilitation of change)
   - Capture of corporate domain knowledge (via issue-oriented domain analysis)
   - Negotiation-based requirements technology

5) Liveware Support
   - Variety of "experts" (stakeholders)
   - Significant variation in abilities

PUBLICATIONS

Field Study Papers

Curtis, B., Krasner,. H., and Iscoe, N. (1988), A Field Study of the Software Design Process for Large Systems, in Communications of the ACM, Vol. 31, No. 11, November, 1988

Krasner, H., Curtis, B. and Iscoe, N. (1987) Communications Breakdowns and Boundary Spanning Activites on Large Software Projects, In Proceedings of the Second Annual Conference on Empirical Studies of Programmers, Chapter 4, Ablex, Inc., Norwood, NJ.

Curtis, B., Krasner,. H., Shen, V. and Iscoe, N. (1987), On Building Process Models Under the Lamppost,In the Proceedings of the Ninth International Conference on Software Engineering, Washington, DC: IEEE Computer Society, 1987, 96-103.

Krasner, H., Shen, V., Curtis, B. and Iscoe, N. (1986) Preliminary Observations from the MCC Field Study of Large Software Projects, MCC Technical Report Number STP-390-86P.

Shen, V., Krasner, H., Curtis, B. (1986) A Field Study Plan for Developing Models of the Design Process, MCC Technical Report Number STP-115-86P.

Team Study Papers

Elam, J., Walz, D., Krasner, H., Curtis, B. (1987), A Methodology for Studying Software Design Teams: An Investigation of Conflict Behaviors in the Requirements Definition Phase,In Proceedings of the Second Annual Workshop on Empirical Studies of Programmers, Chapter 6, Ablex, Inc., Norwoood, NJ.

Walz, D. (1988), Phd Dissertation, U. of Texas, to appear

Individual Study Papers

Guindon, R., Krasner, H., Curtis, B. (1987) Breakdowns and Processes During the Early Activities of Software Design by Professionals, In Proceedings of the Second Annual Workshop on Empirical Studies of Programmers, Chapter 5, Ablex, Inc., Norwoood, NJ.

Guindon, R., Krasner, H., Curtis, B.(1987b) A Model of Cognitive Processes in Software Design: An Analysis of Breakdowns in Early Design Activities by Individuals, MCC Technical Report Number STP-283-87.

Motivational Slide for this Morning

In a study of 38 U.S. and Japanese Companies a wide variety of software management strategies were observed (Cusumano, 1987). It was concluded that Japanese firms are significantly ahead in applying a disciplined and flexible factory approach, as evidenced by:

Japan	$\frac{.26 \text{ bugs}}{1000 \text{ SLOC}}$	5% projects late	34% reuse
U.S.	$\frac{8.3 \text{ bugs}}{1000 \text{ SLOC}}$	43% projects late	15% reuse