Empirical Studies of Software Design: Implications for SEEs

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

Individual level

LIFT experiment
8 experienced programmers designing the control structure for a set of elevators during an intense 2 hr. session

Team level

Object server exp.
Videotaped team meetings from a 7 mo. effort to design and build a tool to support object oriented programming

Project level

Field study
Detailed interviews with key members of 18 large development projects to model their decision-making and communication process

Participant
Experimenter

Customer team

Observers

Project team

Field study team

Shareholder project member

Lockheed
Missiles & Space Company, Inc.
Software Technology Center
11759
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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Terminated</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Support Software</td>
</tr>
<tr>
<td>2</td>
<td>Development</td>
<td>24</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>Radio Control</td>
</tr>
<tr>
<td>3</td>
<td>Development</td>
<td>50</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>Process Control</td>
</tr>
<tr>
<td>4</td>
<td>Development</td>
<td>50</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>Operating System</td>
</tr>
<tr>
<td>5</td>
<td>Design</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CAD</td>
</tr>
<tr>
<td>6</td>
<td>Development</td>
<td>130</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>CAD</td>
</tr>
<tr>
<td>7</td>
<td>Development</td>
<td>150+</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Avionics</td>
</tr>
<tr>
<td>8</td>
<td>Maintenance</td>
<td>194</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>C³</td>
</tr>
<tr>
<td>9</td>
<td>Development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Compiler</td>
</tr>
<tr>
<td>10</td>
<td>Maintenance</td>
<td>250</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Run-time Library</td>
</tr>
<tr>
<td>11</td>
<td>Development</td>
<td>350+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Compiler</td>
</tr>
<tr>
<td>12</td>
<td>Maintenance</td>
<td>400</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Transaction Proc.</td>
</tr>
<tr>
<td>13</td>
<td>Design</td>
<td>500</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>Telephony</td>
</tr>
<tr>
<td>14</td>
<td>Maintenance</td>
<td>725</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>Operating System</td>
</tr>
<tr>
<td>15</td>
<td>Development</td>
<td>1000</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>Telephony</td>
</tr>
<tr>
<td>16</td>
<td>Maintenance</td>
<td>50k+</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Radar, C³</td>
</tr>
<tr>
<td>17</td>
<td>Requirements</td>
<td>100k+</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>C³, Life Support</td>
</tr>
</tbody>
</table>
Summary of Results from MCC Field Study*

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

* Paper appearing in this months CACM
Analysis of Three Significant Problems in Software Design for Large Systems

- Application Knowledge Acquisition
- Fluctuating and Conflicting Requirements
- Communications Breakdowns

Effect on productivity and quality through behavioral processes
Layered Behavioral Model of Software Processes

Business Milieu

Company

Project

Team

Individual

Cognition and Motivation
Group Dynamics
Organizational Behavior
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

---

Missiles & Space Company, Inc.
The Software Project as an Ecological System

- Changing World
- External Technology
- Project
- Internal Processes of Interest
  - Assimilation of knowledge
  - Communication and coordination
  - Managing change
  - Issue resolution and decision making
  - Technical design
  - Organizational bureaucracy
- Industry and Company
  - P&P, Standards, Etc.
- Application Knowledge
- Specific Needs of Customers
- Contracting Mechanisms

Missiles & Space Company, Inc.
Software Technology Center
11766
Five Crucial Problem Areas in Large Software Projects

- External Context
  - Change & Uncertainty
  - Communication
  - Technology Transfer

- Project
  - Communication and Coordination
  - Technology Transfer
  - Design Evolution and Analysis

* see STP-390-86p

Lockheed
Missiles & Space Company, Inc.
Software Technology Center
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 Management Task

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

Overall Conclusion
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

Problem Models:
- essential aspects

Exploration Process:
- commitments
- assumptions
- scenarios of use
- test cases
- simulations
- trade-off analysis

Solution Models:
- issues
- constraints
- evaluation criteria
- discarded alternatives
- focal points

Missiles & Space Company, Inc.
Software Technology Center 11770
Individual Differences in Software Design Strategies

- Exemplar driven (Experienced)
- Method (process) driven (Intermediate)
- Computational paradigm driven (Novice)
- Trial and error driven (Beginner)

General computation strategies
Results of the Team Design Study*

- Identification of conflict behavior 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
   - Commitment 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


Team Study Papers


Individual Study Papers


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:

<table>
<thead>
<tr>
<th></th>
<th>Japan</th>
<th>U.S.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bugs</td>
<td>.26 bugs</td>
<td>8.3 bugs</td>
</tr>
<tr>
<td>1000 SLOC</td>
<td>1000 SLOC</td>
<td></td>
</tr>
<tr>
<td>Projects</td>
<td>5% late</td>
<td>43% late</td>
</tr>
<tr>
<td>Reuse</td>
<td>34%</td>
<td>15%</td>
</tr>
</tbody>
</table>

Missiles & Space Company, Inc.
Software Technology Center