Design of Information Systems: Simulating the Effectiveness of Knowledge Transfer Throughout the System Analysis Phase.

Peter Otto
Salvatore Belardo
University at Albany, State University of New York
School of Business
Agenda

- Challenges
- Problem Statement
- Definitions
- Analysis
- Dynamic Hypothesis
- Model Structure and Feedback Loops
- Policy Simulation
- Future Research
Challenges

• One of the critical issues determining the successful development of information systems is what might be described as the communication gap between the user group and the IS development group that occurs during the system analysis phase.
Problem Statement

• The bigger the knowledge gap between the two groups, the less efficient the systems analysis team will be.

Explicit knowledge:
- User: Vocabulary (A-Z)
- Developer: Vocabulary (E-Q)

Tacit knowledge:
- User: Maturity
- Developer: Rookie
Definitions

**Tacit knowledge:** entails “stuff” that is difficult to express, formalize, or share, i.e. business acumen.

**Explicit knowledge:** what a person can articulate, such as the personnel policy of a firm.
Definitions

• **Information System Analysis:** The early steps in the systems development process, to define the requirements of the proposed system and determine its feasibility.
Analysis

• The quality of communication between the user and the developer group determines the quality of the information systems analysis, which determines the quality of the IT system.
Depending upon one’s mental model, the explicit knowledge provided by one party could be interpreted by another as intended or could be badly misinterpreted, as determined by the receiving party’s tacit knowledge.
Knowledge Transfer

Developer Team:
- Tacit knowledge
- Explicit knowledge

Transfer of knowledge

User Team:
- Tacit knowledge
- Explicit knowledge

Quality of System:
- User satisfaction
- No cost overrun

Effectiveness of Team

E

t
Knowledge Transfer

Developer Team:  
- Tacit knowledge  
- Explicit knowledge

High tacit knowledge

Effectiveness of Team

User Team:  
- Tacit knowledge  
- Explicit knowledge

Quality of System:  
- User satisfaction  
- No cost overrun
Knowledge Transfer

Developer Team:
- Tacit knowledge
- Explicit knowledge

User Team:
- Tacit knowledge
- Explicit knowledge

Effectiveness of Team

Quality of System:
- User satisfaction
- No cost overrun

Low tacit knowledge
“Users Become Smarter” Loop

- Initial user knowledge
- Domain skill
- Initial developer knowledge
- Developer become smarter
- Knowledge acquisition rate of developer

- Transfer of IS skill
- IS skill
- Task specific developer knowledge

- Task specific user knowledge
- Transfer of domain skill
- Time to complete project
- Effectiveness of team
- Need for training
- Burn-out rate
- Burn-out
- Experience gained
- Quality of system analysis

- Confidence in team ability
- Task accomplishment rate

- Motivation for cooperation
- Cost
- Pressure on team
- End user satisfaction

- Knowledge acquisition rate of user
- User become smarter

- Effectiveness
- Motivation for cooperation
- Need for training
- Burn-out
- Pressure on team
- End user satisfaction

- Experience gained
- Quality of system analysis
- Burn-out rate
- Need for training
- Burn-out
- Experience gained
- Confidence in team ability

- Knowledge acquisition rate of user
- User become smarter

- Motivation for cooperation
- Need for training
- Burn-out
- Pressure on team
- End user satisfaction

- Experience gained
- Quality of system analysis
- Burn-out rate
- Need for training
- Burn-out
- Experience gained
- Confidence in team ability

- Knowledge acquisition rate of user
- User become smarter

- Motivation for cooperation
- Need for training
- Burn-out
- Pressure on team
- End user satisfaction

- Experience gained
- Quality of system analysis
- Burn-out rate
- Need for training
- Burn-out
- Experience gained
- Confidence in team ability
Dynamic Hypothesis

- The effectiveness of a system design team is related not only to the initial explicit knowledge that the team will bring to the project but also to the tacit knowledge possessed by the members of the team.
**Dynamic Hypothesis cont.**

**Hope:** System analysis team has the necessary level of tacit and explicit knowledge to finish the project (tasks) in the desired time ($t_d$).
Hope: System analysis team has the necessary level of tacit and explicit knowledge to finish the project (tasks) in the desired time ($t_d$).

**Fear ($F_1$):** Initial low levels of tacit and explicit knowledge levels, causes low effectiveness.
Dynamic Hypothesis cont.

Hope: System analysis team has the necessary level of tacit and explicit knowledge To finish the project (tasks) in the desired time ($t_d$).

Fear ($F_1$): Initial low levels of tacit and explicit knowledge levels, causes low effectiveness

Fear ($F_2$): Mid-term corrections, by either adding more skilled people to the team or by sending team through training.
Key Variables

- Initial tacit and explicit knowledge of developer and user (how can we measure the knowledge levels?)
- Interactions within team
- Task specific team knowledge
- Effectiveness of team
Bloom‘s Taxonomy

Measuring knowledge in the domain of IS Prototyping

• **Vocabulary:** „I have heard the term before. Isn‘t that a method for developing an application in stages, where at each stage the user works with the application and suggests additional functionality“

• **Comprehension:** „Prototyping addresses the situation where users aren't able to specify what they want a system to do. You give them a simple version with only a few features and they learn more about what they want by using the system.“

• **Application:** „Last semester I completed a system for a hospital that followed the prototyping approach. It took 12 versions, but the hospital administrator was pleased.“
Measuring knowledge in the domain of IS Prototyping

- **Analysis:** „The learn by using method is really quite powerful. While it was originally intended as a learning devise for users, it also works as a learning tool for the developer. I have found that I learn as much about the functional area as the user learns about the system’s functionality.“

- **Synthesis:** „I see a good application of prototyping in the design phase. By creating several system designs, in phases much like prototyping, we can cut design time and cost“.

- **Evaluation:** „I think we should measure the characteristics of the system to be developed and based on those characteristics decide which SDM to use.“
Model Structure

- **User explicit knowledge**
  - Initial user explicit knowledge
  - Effect of tacit kn. on explicit kn. gain
  - User exp. kn. gain per task
  - User gaining explicit knowledge
  - User exp. know. gain from working on project
  - User exp. know. gain from training
  - User explicit kn. gain per task
  - Normal experience per task
  - Normal tacit kn. gain per week
  - Effect from user tacit kn. F

- **User tacit knowledge**
  - Initial user tacit knowledge
  - Losing user tacit kn.
  - Losing user explicit kn.
  - Fractional user tacit kn. decay rate
  - User gaining experience
  - Normal user training
  - User explicit kn. decay rate
  - User tacit knowledge

- **Team training based on anticipated schedule overrun**
Model Structure

Effectiveness gains per unit of kn. per week

Increase eff. from gaining knowledge

Task specific user knowledge

User knowledge acquisition rate

User knowledge gain from interaction

Time to gain kn. from training

User kn. gain from interaction

User knowledge gain from interaction

<Team training based on anticipated schedule overrun>

<Time to lose initial kn.>

<Time to lose mature kn.>

<Size of team>

Effect from number of tasks F

Time to gain kn. from training

Dev. kn. gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Team training based on anticipated schedule overrun>

<Time to lose initial kn.>

<Time to lose mature kn.>

<Interactions based on stage of project>

Effect from team size

Normal effect

<Transfer of IS knowledge>

<Developer knowledge>

<Fraction of tasks remaining>

<Time to gain kn. from training>

Dev. kn. gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>

<Transfer of domain knowledge>

TDOK F

<Time to lose initial kn.>

<Time to lose mature kn.>

Dev. knowledge gain from interaction

Dev. normal task specific kn. gain per week

Dev. knowledge gain from interaction

<Interactions within team>
High Initial Knowledge Levels: Base Run

Knowledge and Effectiveness

Task specific developer knowledge : base
Task specific user knowledge : base
Effectiveness of team : base

Time (Week)
Low Levels of Developer Tacit Knowledge

Knowledge and Effectiveness

- Schedule overrun triggers training...
- ...which will compensate for developer kn.

Task specific developer knowledge: low dev tacit
Task specific user knowledge: low dev tacit
Effectiveness of team: low dev tacit
Pressure from schedule overrun: low dev tacit
Comparing the Effect from Low Tacit Knowledge

Developers are able to gain knowledge faster than users!

Task specific team knowledge: base
Task specific team knowledge: low dev tacit
Task specific team knowledge: low user tacit
Low Levels of User versus Developer Knowledge

Effectiveness of Team

Initial low knowledge levels of users cannot be compensated with training
Insights from the Model

• To achieve an efficient knowledge transfer between the developer and user group, it is imperative to have a certain level of tacit knowledge on either side.

• The initial level of user knowledge determines the effectiveness of the team more than the knowledge level of the developer.
Critical Issues

• Because knowledge is an abstract concept, future research should investigate the parameter values that we have used in the model to better quantify the ideas and concepts in the proposed system dynamics model.