

Effective work practices for Free/Libre Open Source Software development



Kevin Crowston*, Hala Annabi^,

James Howison*, and Chengetai Masango*

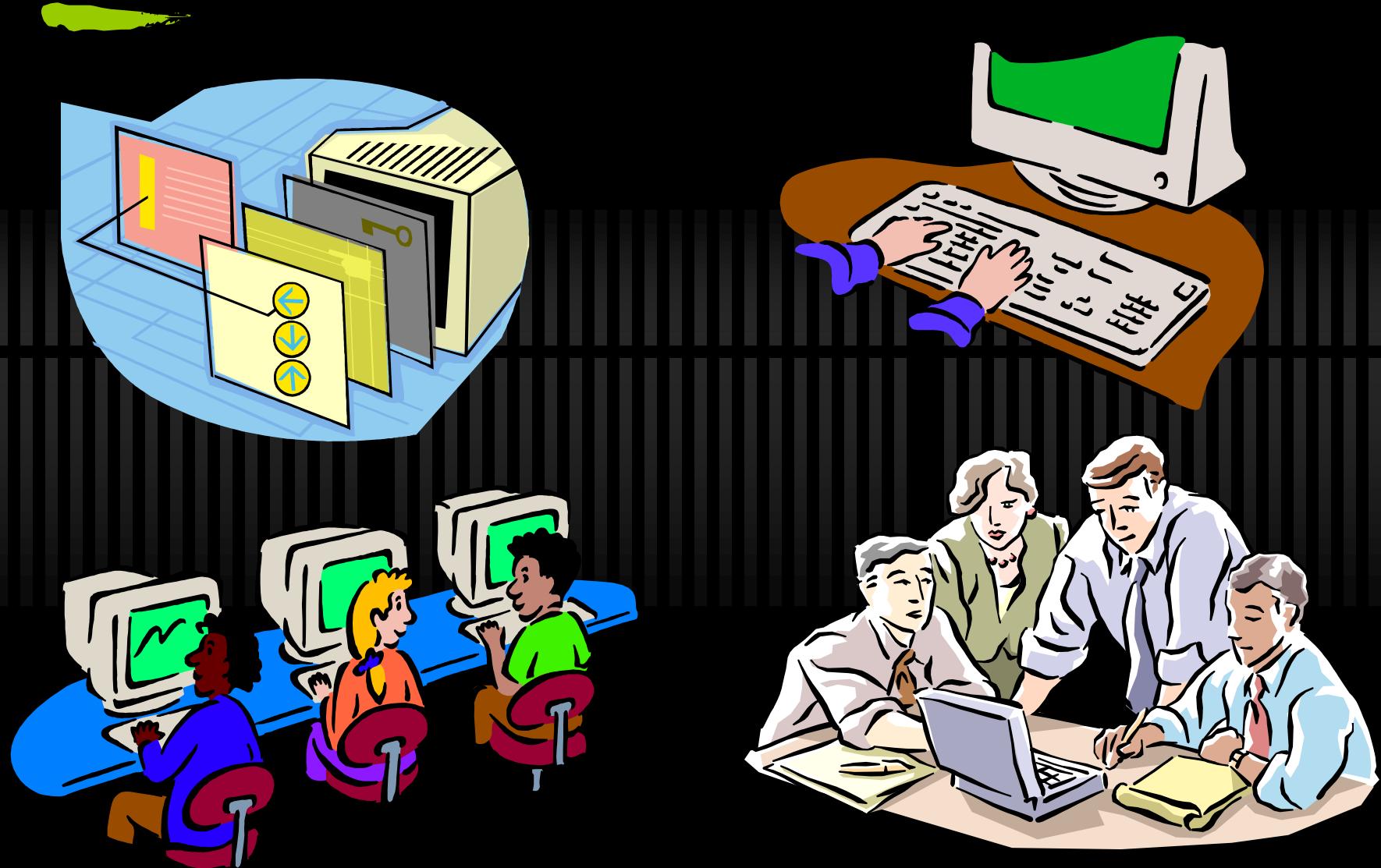
*Syracuse University
School of Information Studies
Syracuse NY USA

^The Information School
University of Washington
Seattle, WA, USA

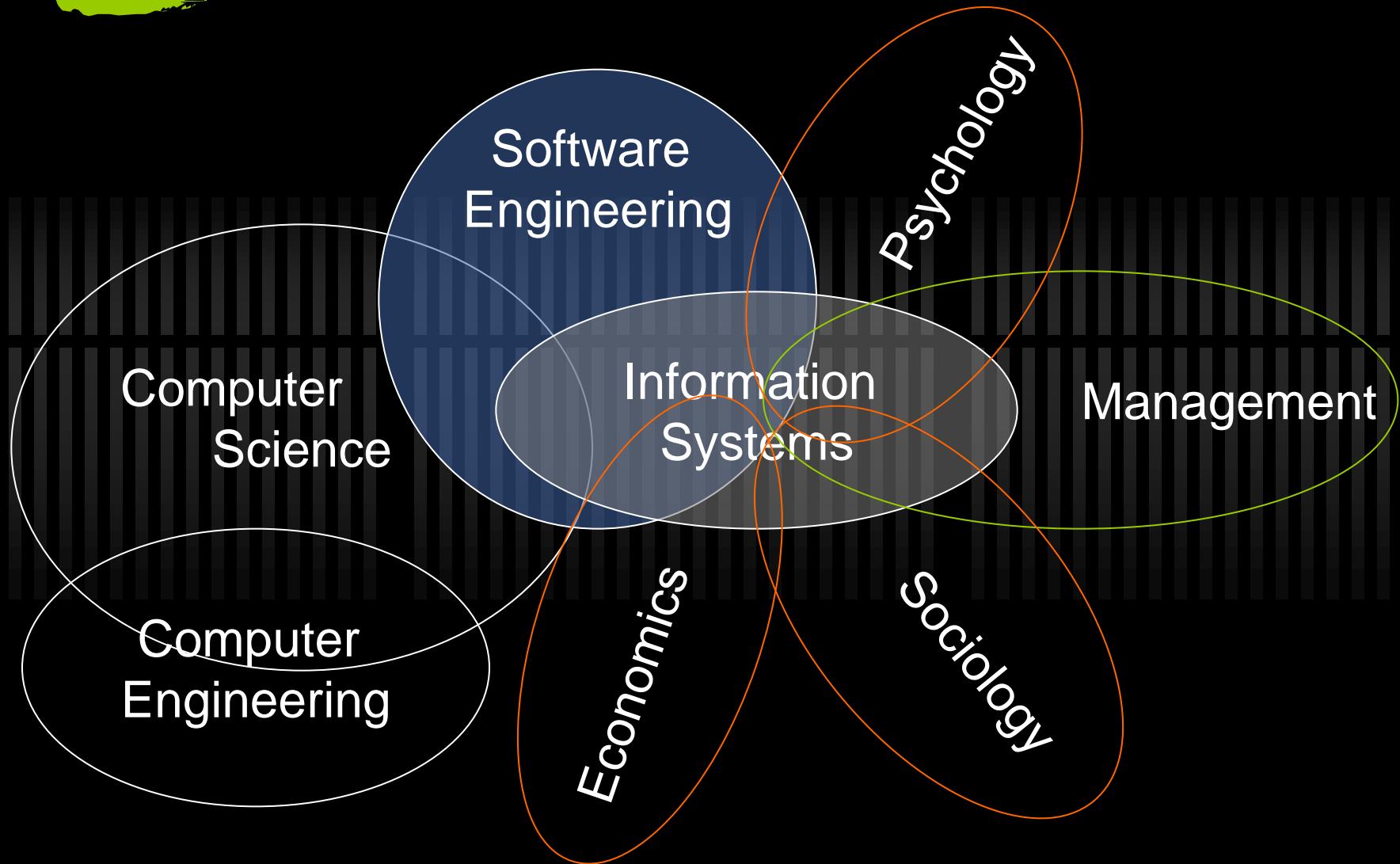
Overview of talk

- 
- ✓ Interdisciplinary software engineering
 - ✓ Study of work practices for OSS
 - ✓ Research questions
 - ✓ Theories
 - ✓ Study design

Domain of software engineering



Software engineering and related disciplines



What is FLOSS?

- ✓ FLOSS = Free/Libre Open Source Software
- ✓ Software distributed under license that allows inspection, modification and redistribution of the source code
 - ✓ AKA free or libre software
 - ✓ “Free as in speech” vs. “free as in beer”
- ✓ Examples: Linux, Apache, gcc, sendmail, X-windows, GNOME, GAIM, OpenOffice, etc.
 - ✓ ...as well as many lesser-known projects

Why FLOSS is interesting for this workshop



- ✓ Mostly developed by distributed teams of volunteers coordinated via the Internet
- ✓ Conway's law: Structure of the software reflects the structure of the team that develops it
 - ✓ Implies that distributed teams should have trouble creating integrated software
- ✓ Successful FLOSS teams somehow overcome problems of distributed software development

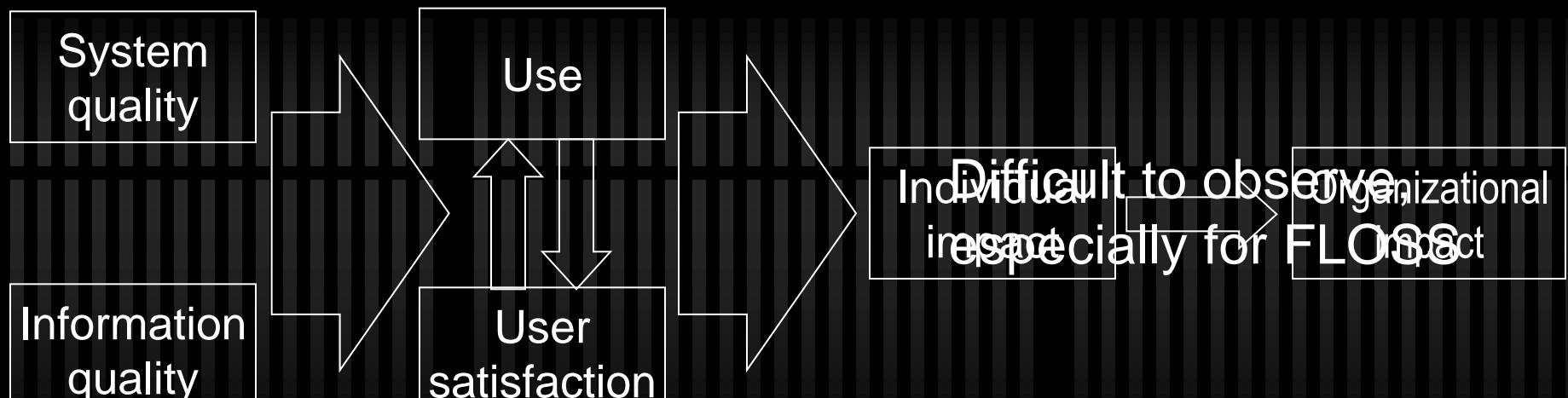
Overall research question



- ✓ What work practices make some FLOSS teams more effective than others?
- ✓ Issues
 - ✓ What do we mean by effective?
 - ✓ What practices should we look for?

Effectiveness: Success measures in IS

DeLone & McLean (1992):

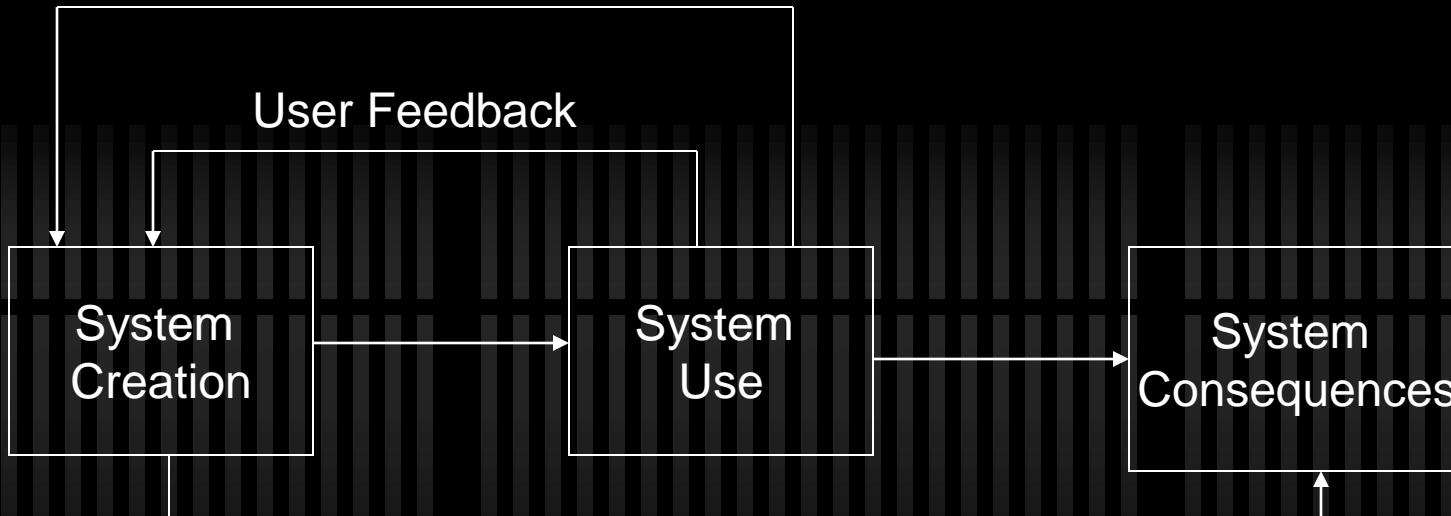


Seddon (1997): system quality, information quality, perceived usefulness, user satisfaction, and IS use

Effectiveness II: Our success model



User & Co-developers Contribution

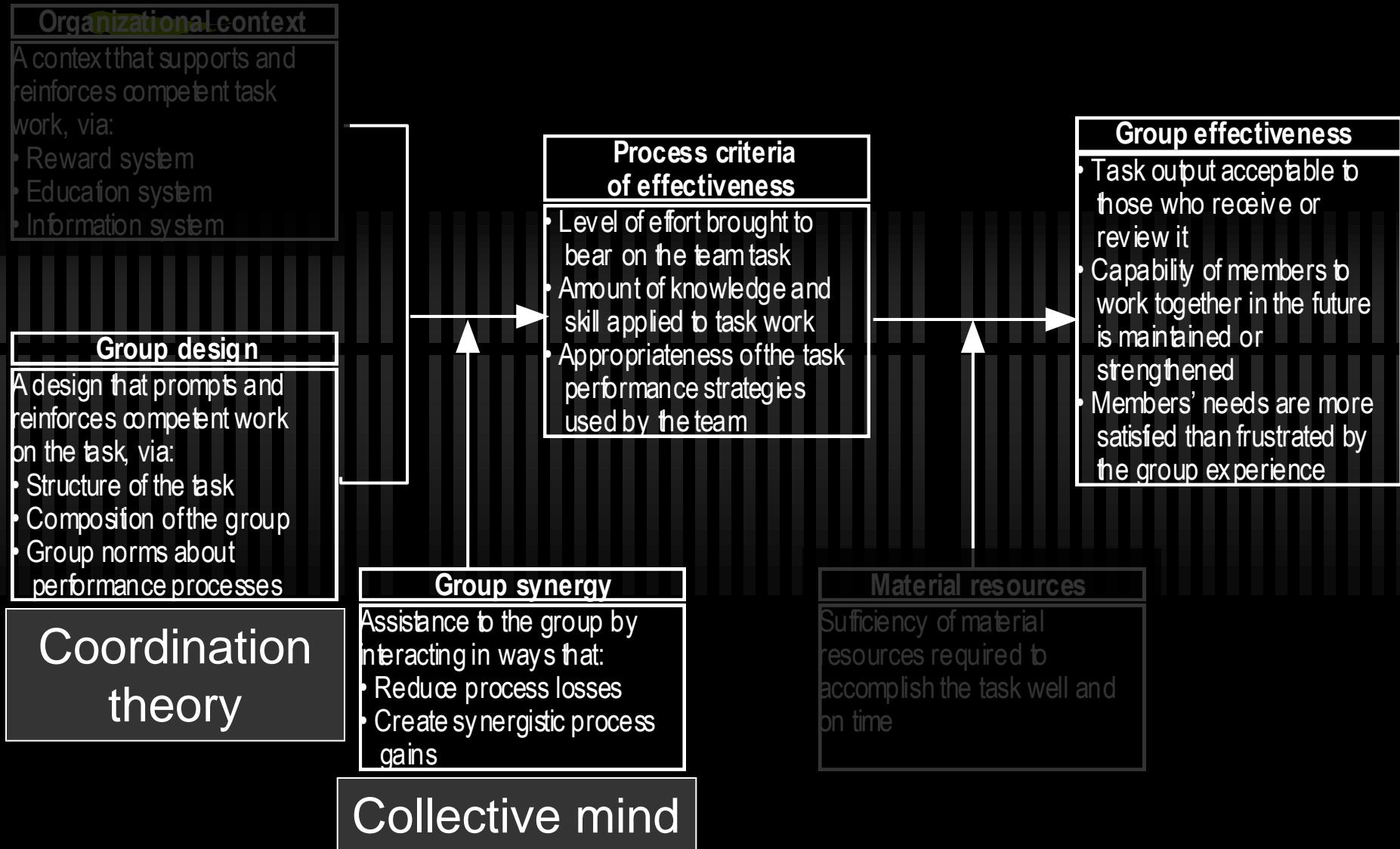


- One-off event vs. “often and early”
- Completion vs. Progress of process
- Number of developers

- User-base
- Downloads

- Developer Satisfaction (developers are often users)
- Meets development goals

Effectiveness III: Hackman's Team Effectiveness Model



Practices of interest



- ✓ Coordination of task
- ✓ Social structures of communication and development
- ✓ Member recruitment
- ✓ Development of norms (e.g., through socialization)
- ✓ Development of collective mind

Practices I

Task Structure: coordination theory

- ✓ Task structure as key input
- ✓ Malone and Crowston
 - ✓ actors in organizations face coordination problems arising from interdependencies that constrain how tasks can be performed

Proposition: Teams with task structures and practices that minimize dependencies will be more effective.

Proposition: Teams with coordination practices to manage dependencies will be more effective.

Practices II

Team synergy: Collective Mind

✓ Addressing Team Synergy through
“Collective Mind”

- ✓ Subordination (Alignment)
- ✓ Contribution
- ✓ Representation

Proposition: Teams with more highly developed shared mental models will be more effective.

Proposition: Teams which are able to align individual goals and team goals will be more effective.

Practices III

Socialization: Participant Observation

- ✓ In depth participant observation study of Plone, a content management system
 - ✓ Importance of IRC, conferences and “sprints”
 - ✓ Core team referred to as authority
 - ✓ Those with aligned commercial purposes (eg web designers) move quickest to centre
 - ✓ Socialization through rich references to geek culture (Star Wars, Ghostbusters, Snowcrash ...)

Proposition: Teams with higher levels of socialization, conversation and narration will display more highly developed shared mental models.

Expanding the WISER framework



- ✓ Information Systems as a column
 - ✓ Process modeling and coordination theory for “manageable processes”
 - ✓ Alignment of Communication/Management and artifact/core structures
- ✓ Consider “open systems” as issue/problem row
 - ✓ Project management of open source and “inner source”
 - ✓ Attracting and retaining quality developers
 - ✓ Managing/motivating non-employees
 - ✓ Managing Intellectual Property risks