Information Management and Knowledge Management (IMKM)

Exercise 7

Basics of Knowledge Management

Case BMW

TUM Chair for Information Systems

Constantin Lutz, David Buchmann, Emir Beglerovic, Maren Käfer, Yimeng Yuan

IMKM WS 20/21

Lehrstuhl für Wirtschaftsinformatik

The Knowledge-Based Theory of the Firm

Theories of the Firm - General:

- Conceptualizations and models **predicting structure** and **behavior** of organizations
  → There is not one universal theory explaining everything, rather many competing ones covering different topics

Theories of the Firm - Knowledge Based:

- The knowledge based theory of the firm is an **outgrowth** of the **resource based** view
  → It focuses on the most strategically important resource of a firm, **knowledge**

Forms of Knowledge:

- Knowing about facts = explicit
  Revealed by **communication**; can be **acquired** for basically **no cost**; can be **stored** in one place and **distributed**
- Knowing how = tacit
  Only revealed by **application**; can only be acquired through practice - slow and **costly**; **scattered** across individuals

Knowledge in Production

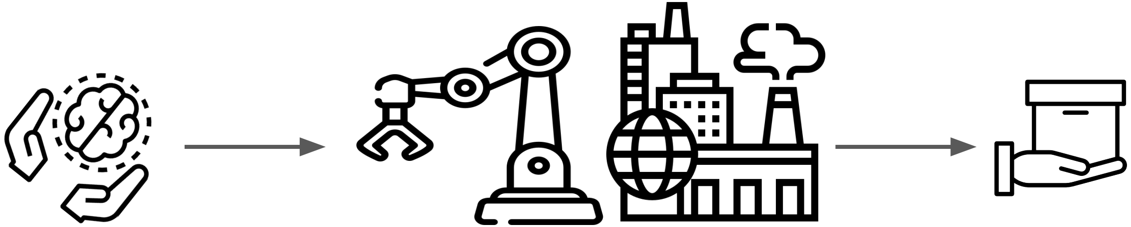
Knowledge Creation:

- Knowledge creation is an individual activity, the firm can only learn through:
  - Learning of its employees
  - Hiring of employees with new knowledge

→ The role of the firm is **not** knowledge **creation**, but the **application** of the individuals' knowledge for **production**

Production:

- A fundamental assumption of the knowledge based theory is that the most **critical input** and primary value source for production, is **knowledge**



Knowledge Integration

Mechanisms for efficient knowledge integration:
Efficient knowledge integration means organizing production in a way that requires as **little knowledge transfer** as possible. Grant proposes 4 mechanisms for that:

1. *Rules and directives*
   Skilled engineer should **rather set up** a set of **rules** and procedures, **instead of teaching** every production worker her knowledge

2. *Sequencing*
   Production process should be designed in a time-patterned **sequence** such that each **specialist's input** occurs **independently** in separate time slot

3. *Routines*
   Complex **patterns of interaction** when rules and directives are absent. E.g. operating fast food restaurant, pit stops in Formula 1 etc. Can also have a repertoire of responses for situation variations

4. *Group problem solving*
   While organizations should minimize communication and knowledge transfer, if **task is unusual, complex** and important, it should be **solved by group** problem solving, involving lots of tacit knowledge communication

Organizational Structure

Implications for hierarchy:

- Simon (1981) argues that hierarchy is a general feature of complex systems emerging because of its evolutionary and problem-solving advantages

  → Companies: 'composed' of interrelated **sub-systems**

- Near decomposability: for most aspects of their functioning each unit may be viewed as **operating autonomously**
- When managers know only a fraction of what their subordinates know and tacit knowledge cannot be transferred upwards, then coordination by hierarchy is inefficient
- 'higher-level decisions' are **dependent** upon immobile 'lower-level' knowledge, hierarchy impoverishes the quality of higher-level decisions

Implications for decision-making:

- The conventional basis for the analysis of decision making is **delegation**

  → employees own the bulk of the firm's resources
  → the quality of decisions depends upon their being based upon relevant knowledge

- Decisions based upon such tacit and idiosyncratic knowledge are **decentralized**, while decisions requiring statistical knowledge are **centralized**

Boundaries of the Firm

Knowledge transfer inefficiently

Prod A → Prod B

Knowledge transfer efficiently

Prod A → Prod B

linked by market interface

- firms are characterized both as **product domains** and **knowledge domains**

  → perfect **congruence** does not exist

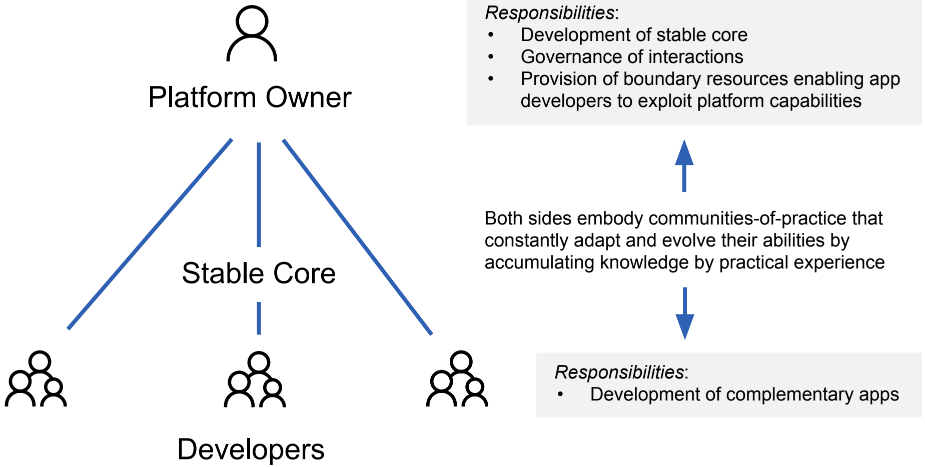
  → creates opportunities for **knowledge trading** to achieve fuller utilization of knowledge

KBV’s Implications for Platforms

A digital platform is “*a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components. It incorporates a central core surrounded by multiple actors in its digital platform ecosystem*”

Hierarchies

Usual case

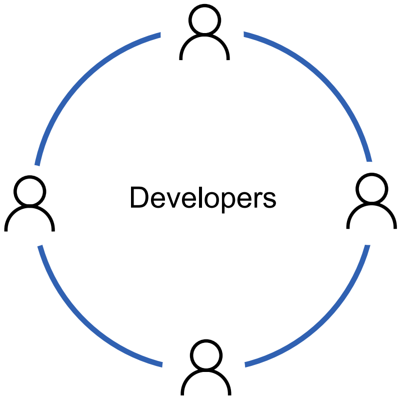


Responsibilities:
- Development of stable core
- Governance of interactions
- Provision of boundary resources enabling app developers to exploit platform capabilities

Both sides embody communities-of-practice that constantly adapt and evolve their abilities by accumulating knowledge by practical experience

Responsibilities:
- Development of complementary apps

In special cases, the platform is not orchestrated by a **single owner** but driven by a **developer community**, which is globally distributed



KBV’s Implications for Platforms

Decision-Rights

Platform Governance

“Partitioning of the decision-making authority between platform owners and app developers, control mechanisms and pricing and pie-sharing structures”

Usual case

- Covers **tactical decisions between platform owner and app developers** - decisions usually transmitted and enforced by resources that enable app developers in their activities
- Beside technical resources, **platform owner provides knowledge on app development as boundary resource**

➡ Multiple levels of knowledge transfer through boundary resources: From owner to app developers via multiple channels - app developers can choose most appropriate for individual requirements

Design of boundary resources is constantly shaped and evolved in a common refining process of platform owner and app developers

→ Platform boundary resources correspond to the platform owner’s decisions on platform governance, thus refinements on platform boundary resources mirror adjustments of platform governance decisions

Special case

- Even though these platforms also require governance, different members of the community consolidate certain decisions
- E.g. app developers are able to get involved as the platform owner and contribute to platform core by submitting pull requests

KBV’s Implications for Platforms

Boundaries

→ The accessibility of the provided platform resources is defined by the configuration of the platform’s **vertical openness**

Vertical Openness

- Defines **degree of accessibility and transparency** of platform boundary resources for external actors
- Platform owner can decide to **limit access** to certain platform assets for a defined group of users or restrict the usage of specific resources
  → decides on the **potential of knowledge transfer** from the platform owner to app developers
- Alternatively, platform owner can **pull external innovation** back into the core of its platform
  → **potential danger** of getting replaced by a platform feature may discourage app developers from entering a digital platform ecosystem, **BUT** overall digital platform ecosystem mostly benefits from improvements of the platform core through coring

Horizontal Openness

- **Interoperability** with other platforms and willingness of **sharing the platform ownership** with others

⚠ Platform with **insufficient level of openness and inadequate transparency or accessibility** to app developers may have the potential to discourage app developers and aggravate the chicken–egg problem. An excessive level of openness on the side may involve the danger of losing control

BMW’s Platform for Automotive Onboard Apps

start in 2016

enables modular deployment of apps in the car

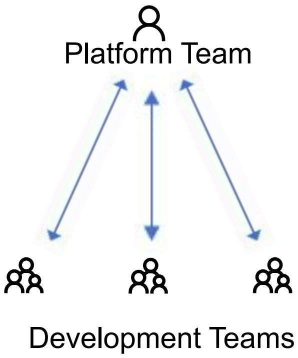
platform managed by a development department = platform team

platform used by app development teams from multiple different departments, distributed all over the organization

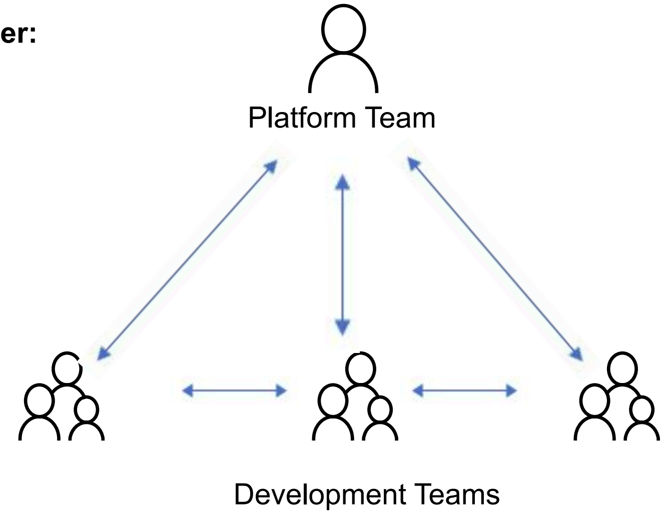
BMW’s Platform for Automotive Onboard Apps

Hierarchies

Initial case:



Later:



- **Control function**
- Setting up **rules and guidelines**
- More approachable

Knowledge exchange trough

- **Community meetings**
- **Developer Portal**

Knowledge exchange through

- Work in **cross-functional teams** (Starter App)

BMW’s Platform for Automotive Onboard Apps

Decision-Making

Initial case:

- Decisions made by platform team (based on knowledge of platform team) → no specialist knowledge from developers leading to less quality in decisions

Later:

- Decisions still made by platform team BUT
- **More attention towards developers’ opinion** (Thread on BMW Answer for new inputs and refinements) → integration of their knowledge into decisions
- Platform team **taking perspective of a developer** (to understand their logic) → increased interaction between platform team and development and knowledge exchange → developers’ knowledge included in decision process

BMW’s Platform for Automotive Onboard Apps

Vertical Openness

- The **accessibility** of the platform development repository enabled app developers to add functionality and merge their change into the platform
- However, the platform team realized that rules for the transfer of artifacts from app developers to the platform were needed to **avoid integration of malicious or insufficient artifacts**
- Hence, it decided to create contribution guidelines to define **clear rules** for external contributions

Horizontal Openness

- “We as platform developer **appreciate external input**. However, we recognize that we need clear rules for that. Otherwise, **we spend too much efforts** in discussions and in general communication with app developers that want to contribute to the platform.”
~Platform team member

Questions

- What was a situation in an internship / job where communication cost was too high due to tacit knowledge and you/your team implemented rules and procedures to decrease the communication costs?
- What could be an applicable way of connecting the sub-systems as described in Simon (1981) within firms?
- Can you think of examples, where platforms failed due to too much/too little “openness”?
- Are you aware of a situation where a bad decision was made because no expert knowledge was considered in the decision process?
- If you are an aspiring application developer in an up-and-coming platform would you be willing to contribute to/code an API for a platform?

Resources

Weiß, N., Wiesche, M., Schreieck, M., Krcmar, H. (2020). Learning to be a Platform Owner: How BMW Enhances App Development for Cars. *IEEE Transactions on Engineering Management, (Early Access)*. doi: https://doi.org/10.1109/TEM.2020.3017051

Grant, R. (1996). Knowledge and the Firm: An Overview. *Strategic Management Journal*, 17(S2), 109-122. doi: https://doi.org/10.1002/smj.4250171103