



Developer's output is knowledge

...and not code

The New New Product Development Game

by **Hiroataka Takeuchi** and **Ikujiro Nonaka**

FROM THE JANUARY 1986 ISSUE

Moving the scrum downfield

From interviews with organization members from the CEO to young engineers, we learned that leading companies show six characteristics in managing their new product development processes:

- 1 Built-in instability
- 2 Self-organizing project teams
- 3 Overlapping development phases
- 4 "Multilearning"
- 5 Subtle control
- 6 Organizational transfer of learning



Contract game problem is essentially misunderstood product development



**Rewriting an existing product is
much cheaper than coding a new one**



**If code is output, then you will want to
measure the amount of code/features/
jiras/.... always**

**“We need to estimate and predict when
it will be finished”**



**LLM's don't learn and it will not happen
any time soon**



Routine expertise

- Sushi chef
- Java developer
- UI designer
- Limited group of component / microservices expertise

Adaptive expertise

- Problem solver
- Innovative
- Customer and product perspective
- High skill-acquisition efficiency

Hatano, G. and K. Inagaki (1986). "Two courses of expertise." *Child development and education in Japan*: 262–272.



Testing automation is about retaining knowledge in the long term



Team cognitive load (from Team Topologies) is a harmful concept since it limits to learning

Organization has to be designed for learning or knowledge will not grow



Knowledge and LeSS



Product increment is the ultimate measure of progress



**LeSS is designed to maximize
learning and knowledge**



Measure individual capability (knowledge and skills) and not performance



How does a better team look like?

How do we measure that?



Remote working limits and slows down learning