



APPLICATION OF THE SECI KNOWLEDGE MODEL

By: Joe Schofield

Abstract: Ikujiro Nonaka and Hirotaka Takeuchi are not the first names you might associate with agile frameworks or Scrum in particular. Yet, as progenitors of the age of agility, perhaps they should be. Their proclivity to pioneer persisted with the knowledge generation model known as SECI. Evidence of its usefulness surrounds us, even in the realms of functional measurement, metrics, and benchmarking.

What came first: the chicken, or the egg? ⁰ Scrum or agile? Iterative and incremental development or adaptive software development? Daily stand-ups or user stories? Software functional measurement or software benchmarking?

Sometimes it all depends on who you ask. Best to ask someone who knows. If that's too much to pursue, then read on.

Scrum or agile? Many agilists might blurt out that Scrum, at least as described by Schwaber and Sutherland, predates agile for software with deference to the development of the agile

manifesto. Schwaber and Sutherland presented their novel approach to software development in Austin, Texas, in 1995.¹ And, it wasn't until early in 2001 that the development of the Agile Manifesto began.² But that does not settle the question. In 1986 the word "scrum" was used to describe a team moving an item forward in its initial development with specific application of the "rugby approach."³ Manufacturing was the subject of this innovation not software. Nonetheless consider the characteristics described by Nonaka and Takeuchi's paper when compared to Scrum and the agile principles that remain popular today.

	Nonaka & Takeuchi Characteristic ⁴	Parallel to agile principles (A,n) and Scrum (S)
1	Built-in instability	Welcome changing requirements; even late in development (A,2)
2	Self-organizing project teams	A Scrum foundation, trusted teams (A,5); best work products (A,11)
3	Overlapping development phases	The heart of iterative development
4	"Multilearning," Cross-fertilization	The essence of cross-functional Scrum teams ^{5,6}
5	Subtle control	Scrum's empirical process control (S); intervals of reflection (A,12)
6	Organizational transfer of learning	Implicit (A,12); Scrum of Scrums (S)

Regarding the fourth characteristic above referencing “cross-fertilization,” one could be accused of being remiss if foregoing the acknowledgement that the italicized text below sounds very similar to Alistair Cockburn’s “osmotic communication.”

“When all the team members are located in one large room, someone’s information becomes yours, without even trying. You then start thinking in terms of what’s best or second best for the group at large and not only about where you stand. If everyone understands the other person’s position, then each of us is more willing to give in, or at least to try to talk to each other. Initiatives emerge as a result.” ⁷

Given the similarities above, one could reasonably posit that the concepts of agile and the bedrock of scrum were inextricably portrayed together in *The New New Product Development Game*. Consider that half-a-century earlier Shewhart’s Plan-Do-Check-Act (PDCA) stipulated iteration⁸, as well as experimentation. But Scrum’s empirical process control is a far cry from Shewhart’s statistical process control (SPC) in the 1920’s.⁹ Nor does SPC align well with the Manifesto’s initial value statement of “individuals and interactions over processes and tools.” Regarding the sequence of Scrum or agile, perhaps the answer is merely “yes” with neither as a relevant predecessor of the other.

Measurement advocates may be pondering, “What does any of this have to do with software functional (or non-functional) size?” While software, hardware, and services products are developed today using agile techniques, many with user stories, the potential of aligning story points with function points is hardly novel.¹⁰ We turn again to Nonaka and Takeuchi who provided us with much more than scrum and agile concepts. Nonaka early on, and Takeuchi later, were also the developers of the SECI knowledge model wherein a broader more expansive learning awaits.

Socialization, Externalization, Combination, and Internalization¹¹ (SECI) Knowledge Model

As highlighted in the footnote¹² this abbreviated overview merely introduces the SECI model. The importance of this topic flows from its bridging—if only in part—between today’s agile software development (iterative development) and of IFPUG’s own expanding knowledge domains (incremental accretion). Perhaps a simpler and clearer description of the model was coaxed from ChatGPT;¹³

The SECI model of knowledge creation describes the process of how knowledge is created and shared within an organization. It consists of four main modes: Socialization, Externalization, Combination, and Internalization. The SECI model shows how tacit knowledge (personal knowledge that is difficult to formalize and communicate) is converted into explicit knowledge (codified and documented knowledge) and vice versa through socialization, externalization, combination, and internalization. Socialization refers to the

process of sharing tacit knowledge through direct interaction and observation, while externalization involves articulating tacit knowledge into a form that can be shared with others. Combination refers to the process of merging explicit knowledge from different sources, and internalization is the process of converting explicit knowledge into tacit knowledge through learning and application.

Figure 1 below identifies the four quadrants, or modes of knowledge conversion in the SECI. Each of these stages represents a type of knowledge contagion from person (individual) to person, from person to team (group), and from team to organization (a “container” of teams).

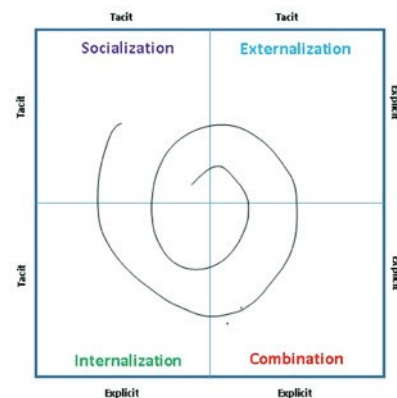


Figure 1:
SECI Knowledge Model



While software, hardware, and services products are developed today using agile techniques, many with user stories, the potential of aligning story points with function points is hardly novel.

A brief description of each of the four quadrants includes:

1. *Socialization* is the sharing of knowledge between (not among) individuals. It may include tutors or mentors, and often observation that becomes imitation. Labeled as tacit, it is typically spoken or verbal.
2. *Externalization* is the codification of knowledge that can be shared with others, as with a team or networking group. It may include images, text, and models that may be formally published or merely notes, checklists, diagrams, or cuneiforms. Described as explicit, it captures verbal expression in a written format.

3. *Combination* is the identification of connections among explicit knowledge domains such that comparisons, analysis, and processing of knowledge—within and beyond the organization itself—leads to new knowledge and understanding. Considers a collection of domain teams convening to solve a challenge to which each holds a piece of the puzzle. Also described as explicit, harvestings are written but not necessarily textually. See externalization for other retention techniques.

4. *Internalization* is the application of new knowledge due to combination, at the individual level, to identify patterns (CRM) while applying them to domains of non-origin. The application of observed events and outcomes is typically expressed first verbally; thus, the looping or spiraling effect back to tacit.

Selected comparisons and contrasts between the SECI and other similar concepts:

- The Personal Software Process (PSP), the Team Software Process (TSP), and the Capability Maturity Model Integration (CMMI) similarly focused on individuals, groups, and organizations, respectively.
- As attributed to George Box, “all models are wrong, but some are useful.”¹⁴ The SECI model has its distractors.¹⁵ While the model may not be precisely representative about all instances of knowledge sharing, its premise of knowledge spreading from individuals eventually to groups and the organization is both practical and empirical.
- The Four (learning) Stages of Competence in psychology¹⁶ differs from the SECI model as its focus is the staging of learning by an individual for a skillset from *unconscious incompetence* to *unconscious competence*. This model is individual-centric, though its application need not be limited to individuals.
- Malcolm Gladwell’s 10,000 hours of learning in pursuit of mastery¹⁷ of a skillset pertains to a given knowledge domain; it has little to do with group or organizational learning.
- The notion of the “spiral” in the SECI is reminiscent of those familiar with Barry Boehm’s work around software development and the spiral model introduction in 1986.¹⁸

The following table exemplifies the use of the SECI knowledge model with the introduction of function points almost 45 years ago and more recently, Software Non-functional Assessment Process (SNAP). The example suggests each of the SECI stages, the transitions among tacit and explicit, and the connections among individuals, teams, and organizations. Other potential IFPUG-relevant fields of knowledge that could have been used in the example include: measurement, metrics, Simple Function Points, and benchmarking.

An SECI Example—Knowledge Sharing Within IFPUG

ACTION	SECI STAGE	TACIT / EXPLICIT	INVOLVES	CONTENT APPLIED
A member shares an idea verbally with a fellow member	Socialization	Tacit to tacit	Individuals to individuals	Function Points
The idea is presented at a conference or in an article such as this	Externalization	Tacit to explicit	Individual to groups	Function Point Analysis
Other committees’ members apply past knowledge to become a standard like ISO	Combination	Explicit to explicit	Groups to organization	SNAP
IFPUG shares its new standard with members who understand how to apply it	Internalization	Explicit to tacit	Organization to individuals	Function Point Analysis and SNAP

Why does this matter and why should we care? The SECI knowledge model serves as a:

- reminder to the importance of writing, drawing, or otherwise recording knowledge,
- lens to better understand knowledge contagion, and perhaps how to stunt misleading information,
- taxonomy for determining the “stage” of knowledge sharing and to how to accelerate its internalization,
- pattern for successful introduction of emergent thinking and targeted communication channels.

The SECI model also helps us to recognize how interconnected technical and social skillsets can be assimilated for accelerated knowledge dissemination among individuals, groups, and organizations. The outcome of such learnings is of value to IFPUG, our own organizations, and ourselves.

But Wait, There’s More

For integrity’s sake, let’s return to the opening questions in this article and proffer some answers.

- Adaptive software development techniques such as eXtreme Programming, Feature-Driven Development, Test-Driven Development, Crystal Clear Methodology, to name a few,

became popular in the 1990s. Barry Boehm's Spiral Model, arguably with iterative and incremental embedded, appeared in 1986. But some 30 years earlier, in 1957 Gerry Weinberg is quoted as using incremental development in Los Angeles, at IBM.¹⁹

- Daily stand-ups or user stories²⁰ Daily stand-ups were practiced in 1993 and included in conference proceedings in 1994. ²¹ Sutherland seems to have incorporated them into what was becoming Scrum for software. User stories were being used around the same time in Detroit by Kent Beck. Alistair Cockburn is credited with devising the phrase "user stories" and Mike Cohn with the "so that" or business purpose often used today.²²
- Software functional measurement²³ or software benchmarking.²⁴ While two references are provided, I leave deeper investigation to interested parties.

Closing

The array of learning, improvement, and communication models, many of which are referenced in this article, fosters innovation within technology and society. Mastery of all of them isn't necessary for us to grow from any one of them. We see the benefits of their usage, intended or otherwise, in our daily engagements and work products. This holds true in our own areas of special interests in IFPUG. Recognizing those patterns may help us to accelerate reaching desired outcomes, in work as in life. 🍷

REFERENCES:

⁰Genesis 1:21 describes the creation of animals, including winged birds, on the 5th day; no eggs were mentioned.

¹Business Object Design and Implementation Workshop held as part of Object-Oriented Programming, Systems, Languages & Applications '95; (OOPSLA '95) in Austin, Texas

²<https://www.agilealliance.org/agile101/the-agile-manifesto/>

³The New New Product Development Game; Nonaka and Takeuchi; Harvard Business Review; 1986

⁴Ibid, pg. 138

⁵The Scrum Guide; November, 2020; Schwaber & Sutherland; page 5

⁶Scrum Body of Knowledge; SCRUMstudy, Edition 4, 2022; pages 21, 23, 47, 57, 116, et al

⁷The New New Product Development Game; Nonaka and Takeuchi; Harvard Business Review; 1986; pg. 140

⁸<https://en.wikipedia.org/wiki/PDCA>; retrieved 3/20/2023

⁹https://en.wikipedia.org/wiki/Statistical_process_control; retrieved 3/19/2023

¹⁰Function Points, Use Case Points, Story Points: Observations from a Case Study; CrossTalk; May / June, 2013

^{11,12}This short space does not allow, nor I am capable of detailing the intricacies of Nonaka's (with refinements later from Takeuchi)

SECI knowledge model. The serious scholar is encouraged to execute a more thorough search of the Model, its uses, and criticisms to extricate salient gleanings.

¹³ChatGPT unique rendering of the SECI knowledge model 3/22/2023; <https://chat.openai.com/chat>

¹⁴Box, George E. P.; 1976; "Science and statistics" (PDF), Journal of the American Statistical Association; 71 (356): 791-799,

¹⁵Gourlay, Stephen; 2006; "Conceptualizing Knowledge Creation: A Critique of Nonaka's Theory"; Journal of Management Studies; 43 (7): 1415-1416; 1421.

¹⁶https://en.wikipedia.org/wiki/Four_stages_of_competence; retrieved 3/22/2023

¹⁷<https://www.businessinsider.com/malcolm-gladwell-explains-the-10000-hour-rule-2014-6?op=1>; retrieved 3/22/2023

¹⁸Boehm, B.; August, 1986; "A Spiral Model of Software Development and Enhancement"; ACM SIGSOFT Software Engineering Notes; 11 (4): 14-24

¹⁹Iterative and Incremental Development: A Brief History; Larman, Craig; Basili, Victor R.; 2003; Computer 36 (6): 47-56.

²⁰<https://www.linkedin.com/pulse/20140926150354-136414-the-origin-of-the-daily-stand-up/>

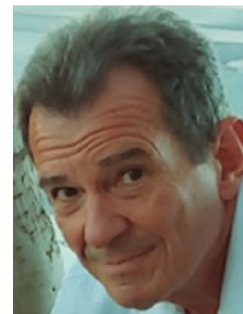
²¹Borland Software Craftmanship: A New Look at Process, Quality and Productivity; James O. Coplien, AT&T Bell Laboratories; Proceedings of the 5th Annual Borland International Conference; Orlando, Florida; 5 June 1994

²²https://en.wikipedia.org/wiki/User_story

²³https://en.wikipedia.org/wiki/Software_measurement

²⁴<https://www.castsoftware.com/pulse/software-benchmarks-and-benchmarking>

ABOUT THE AUTHOR



Joe Schofield, SCT, SCAC, SSMC, SSPOC, SMC, SPOC, SDC, SAMC, CSQA, CSMS, SA IFPUG Past President