

Applying the Dreyfus Model of Skill Acquisition to the Adoption of Kanban Systems at Software Engineering Professionals (SEP)

Christopher M. Shinkle
Software Engineering Professionals
cmshinkle@sep.com

Abstract

In 2004, SEP tried adopting Agile practices. However, Agile failed to have the desired lasting impact across the entire organization. Things changed in 2007, when SEP implemented a kanban system for the first time. Kanban created an effective means for institutionalization of Lean by providing a unique method of discovering and learning practices and principles. Upon examining multiple teams using kanban systems it was apparent that teams followed similar patterns of adoption and learning. They appeared to follow the Dreyfus Model for Skill Acquisition.

1. Organization Characteristics

SEP is a privately held software engineering company located near Indianapolis, IN. It was founded by engineers in 1988 and has grown to include 70+ employees. SEP collaborates with many Fortune 500 companies to develop new software systems and products for business, data and safety critical applications. They offer full lifecycle software solutions to clients in the medical, aerospace, healthcare and national defense markets.

Projects at SEP average four to six people in size and last an average of six months, with a high degree of variance in size and duration.

2. Early Agile Adoption

Early in 2004, SEP began experimenting with Agile practices – specifically Feature Driven Development (FDD). The approach was to try FDD on a project, evaluate the results, and then make a decision about how to proceed. The project was a huge success and a decision was made to teach other teams/projects one at a time about FDD and the specific practices being used. We hoped that we could institute FDD throughout the organization through this organic approach. In order to facilitate this learning, I began to coach project teams – first starting with the Agile

principles and then moving to practices. I hoped that by first teaching the principles, the adoption and retention rates would be higher. I was wrong.

FDD and Agile failed to have the lasting impact on the organization that I had hoped. As a company, SEP was Agile-friendly, but we weren't able to get buy-in from everyone. Widespread adoption of Agile practices was met with resistance and ultimately failed to gain traction.

Over time, experience with Agile practices within SEP grew, but only in limited pockets. This continued until early 2007 with the implementation of our first kanban system.

3. Kanban Adoption

After the initial implementation in 2007, it quickly became apparent that kanban would have different results. Kanban provided a more effective vehicle to introduce practices and principles. The resistance I had previously encountered seemed to be nonexistent. The culture on project teams began to change as they learned the system. And more importantly, the attitudes of team members changed.

Since 2007, I have implemented more than a dozen kanban systems at SEP. While coaching these project teams, an adoption and evolution pattern began to emerge. The maturation of a kanban team correlated strongly to the understanding of Lean principles.

I began to research different learning models and found the Dreyfus Model for Skill Acquisition. The model provided guidance for the pattern I saw emerging and why our previous attempts at introducing Agile and Lean principles failed.

4. The Dreyfus Model of Skill Acquisition

The Dreyfus Model of skill acquisition was developed in the early 80s by two brothers – Hubert and Stuart Dreyfus. They surmised that “in acquiring a skill by means of instruction and experience, the student normally passes through five developmental

stages.” [1] They labeled the stages as Novice, Advanced Beginner, Competent, Proficient, and Experts. “As the student becomes skilled he depends less on abstract principles and more on concrete experience.” [1] They concluded that any skill-training procedure must be based on some model of skill acquisition. [1]

The model was originally prepared for the Air Force Office of Scientific Research for the training of pilots, but has since been applied to nurses, chess players, and even software developers. The model has traditionally been applied to individuals; however, I used the model to evaluate software teams. Understanding the model allows us to address the appropriate issues at each stage in order to facilitate team advancement.

The Dreyfus skill model is summarized in Figure 1 [4].

Skill Level	Components	Perspective	Decision	Commitment
Novice	Context free	None	Analytic	Detached
Advanced Beginner	Context free and situational	None	Analytic	Detached
Competent	Context free and situational	Chosen	Analytic	Detached understanding and deciding
Proficient	Context free and situational	Experienced	Analytic	Involved understanding; detached deciding
Expert	Context free and situational	Experienced	Intuitive	Involved

Figure 1

At each stage, you don’t just “know more” or gain skill. Instead you experience fundamental differences in how you perceive the world, how you approach problem solving, and the mental models you form and use. [2] The different stages reflect changes in three general aspects of skilled performance: movements from abstract principles to concrete experience as paradigms, situations are not seen as equally relevant bits but as a complete whole, and movement from a detached observer to an involved performer. [3]

For each stage, I will identify the characteristics that define each stage and describe my observations from the teams I observed at SEP.

4.1. Stage 1 – Novice

At the novice stage, individuals have little or no previous experience. “The instruction process begins by decomposing the task environment into context-free features which the beginner can recognize without benefit of experience. The beginner is then given rules for determining action on the basis of these features.” [1] Individuals often exercise no discretionary judgment at this stage. Their focus is on accomplishing an immediate goal and they often have little desire to learn. Novices often are faced with the

question as to which rules to apply to a given situation. To improve, the novice needs monitoring and coaching to bring their behavior more into compliance with the rules.

Think about a typical person wanting to bake a cake. They go to the store, purchase a boxed cake mix, and follow the directions as described on the back of the box. They have little to no knowledge about how to alter the recipe nor do they have a desire to do so. Their goal is simply to bake a cake.

At this stage, teams at SEP most often viewed the kanban board as a task tracking system with a simple rule set such as: when work is completed move the sticky note to the next section on the board, use yellow sticky notes for scenarios, blue for bugs, pink for issues, etc. and prioritize the work to be started in the engineering ready queue. Teams rarely showed regard for Work-In-Progress (WIP) limits – without which the system isn’t a ‘pull’ system. Daily stand-up meetings were conducted in which the teams discussed only the state of the current work items.

Despite this very limited implementation of a kanban system, teams still experienced a significant culture change. They told me they had a better understanding of the state of the project. They had a much better understanding of the priority of work items. They knew what to work on next without having to ask or be told. The board provided a very easy way to get a sense for their progress. As one team member put it, “We never implemented limits. What kanban DID accomplish though was illustrate the development process to a team that had never thought about it much before. It basically exposed them to their selves.”

4.2. Stage 2 – Advanced Beginner

“As the novice gains experience actually coping with real situations and begins to develop an understanding of relevant context, he or she begins to note, or an instructor points out, perspicuous (clear) examples of meaningful additional aspects of the situation or domain.” [4] Advanced beginners start to break away from rule sets as they gain experience from seeing a sufficient number of examples. However, advanced beginners don’t experience a holistic understanding. They don’t want it and likely would dismiss it as irrelevant. At this stage, principles, based on experience start to be formulated.

Returning to the cake example, an advanced beginner understands how to apply some context to the instructions or rules on the back of the box. They can make minor adjustments for things like altitude, pan size, oven conditions, etc. They are still following the

basic recipe, but can make minor adjustments likely based on previous experiences.

The biggest change I noted from novice to advanced beginner was that the team attempted to limit the amount of work in progress. For the first time, they began to realize how too much work in progress negatively impacted the system. When WIP was too high, lead times grew and as a result so did the bugs and rework. The relationship between WIP and quality started to be grasped. Poor quality resulted in more rework, leading to higher WIP, and ultimately poorer quality. This basic principle encouraged teams to attempt to control WIP.

Another interesting thing began to occur. As a result of the blocked work items and fixed WIP limits, team members began to collaborate more with each other. If they were unable to complete their work item and couldn't start new work because of the WIP limits, they would help others complete their tasks. This wasn't always easy - they weren't used to collaborating on development tasks – but the payoff was huge. Team members began to learn how to coordinate work with each other in ways that would allow for lower WIP limits in the future.

Finally, as a result of the WIP limits, teams began to identify bottlenecks in their process. This led them to seek out areas for improvement to increase the flow of the system and reduce the opportunity for the system to become clogged. It was because of the WIP limits that teams first started thinking about continuous improvement.

Keith, a developer working on an SEP team realized the result of enforcing WIP limits all too well. “WIP limits seem to be the worst understood part of the kanban system. When used properly, it exposes bottlenecks and reduces lead time for individual work items. Used improperly, it can starve developers for work or result in too many people working on the same work items.”

4.3. Dreyfus, Deming, and Fear

The Dreyfus model suggests that taking responsibility is often a critical point for people when learning a new skill. Most people never become competent without a willingness to accept responsibility for their decisions.

W. Edwards Deming gives us a clue as to why people are unwilling to accept responsibility – fear. Deming tells us that fear is a barrier to improvement. “No one can put in his best performance unless he feels secure. *Se* comes from the Latin, meaning without, *cure* means fear or care. *Secure* means without fear, not afraid to express ideas, not afraid to ask questions. Fear takes on many faces. A common denominator of

fear in any form, anywhere, is loss from impaired performance and padded figures.” [5] We should strive to “drive out fear” from the organization. Teams operating in an environment where they fear making decisions will paradoxically *not* improve or reach their potential – they will avoid new ideas and changes because of their inherent risk and potential consequences. As long as fear exists in the organization, it is very unlikely individuals will be willing to accept responsibility for their decisions.

Fear leads to a widespread resistance to knowledge. [5] Kanban provides the opportunity for team members to make changes confidently and with less fear by providing mechanisms for monitoring the effects of those changes. There is no need to fear knowledge when one can see how the knowledge is being used. Through this in-depth knowledge of the system, kanban helps teams realize and understand variation in the system. As a result, kanban systems work to drive out fear.

4.4. Stage 3 – Competent

“Competence comes only after considerable experience actually coping with real situations...” [1] From these situations, patterns and principles begin to emerge. The previous “rules” become “rules of thumb” and guidelines begin to be established. “The competent practitioner will seek out and solve problems; their work is based more on deliberate planning and past experience. Without more experience, they’ll still have trouble trying to determine which details to focus on when problem solving” [2]

On competent teams at SEP, all team members began to participate in making process changes. The team members began to feel a sense of ownership, and as a result, started seeking alternate practices to optimize the process. People are often resistive to new practices being pushed on them. However, at this stage, teams pursued the new practices themselves and were thus far more open to accepting them. The practices were being introduced to solve specific issues. As teams adopted new practices and improved the process they experienced a genuine sense of gratification. The team witnessed items flowing through the system more smoothly than before. They made process changes and observed the effects of those changes.

Competent teams began to see and understand the concepts of flow, value, and pull. They saw these things at work every day. Using a kanban systems and associated practices, teams began to derive Lean principles by themselves. The teams started to make changes and solve problems using Lean principles

without knowing what Lean was! One developer, Julie, remarked, “It was like switching on a light, no joke. Suddenly, everyone was able to offer ideas on how to improve our processes. It really motivated everyone.”

From the beginning, there appears to be a relationship between teams using a kanban system and having a working knowledge of Lean principles. What I saw led me to believe that the maturation of a kanban team caused teams to learn and understand Lean principles.

4.5. Stage 4 – Proficient

Proficient students begin to seek out and want to understand the “big picture”. They start to see the system holistically. The system is perceived as a complete system instead of a set of individual parts.

“Proficient practitioners make a major breakthrough on the Dreyfus model; they can correct previous poor task performance. They can reflect on how they’ve done and revise their approach to perform better the next time. Up until this stage, that sort of self-improvement is simply not available.” [2] Additionally, they can learn from the experience of others and take full advantage of reflection and feedback.

While moving from competent to proficient, changes to practices on teams at SEP were less prominent. However, one very noticeable change occurred around planning. Teams made throughput and reducing cycle-time its primary focus and goal. In order to accomplish this, teams began to focus on optimizing the whole and reducing the cost of delay. Instead of just following WIP limits, they were actively trying to reduce their WIP. Stand-ups were more focused on improving the system rather than tracking progress. Kaizen moments became more commonplace. Matt, a developer, pointed out that, “The entire picture allows me to work with full knowledge of our progress in relation to our goals. The limits in each section force me to work in the areas that most need help – in order for the board to keep flowing.”

4.6. Stage 5 – Expert

“The proficient performer, immersed in the world of his or her skillful activity, sees what needs to be done but decides how to do it. The expert not only sees what needs to be achieved; thanks to his or her vast repertoire of situational discriminations, he or she also sees immediately how to achieve this goal.” [4] The expert no longer relies on rules, guidelines, or maxims, but work primarily from intuition. The decisions made

stop being conscious decisions and are based solely on instinct. An expert has enough experience in an area that each situation encountered dictates an intuitively appropriate response. The expert can distinguish between very important details and those considered irrelevant. They know which details can be safely ignored.

This expert stage was not something I was able to observe at SEP. Some estimate it can take as many as 10-15 years in a particular area to reach expert. Only time will tell what the expert stage might look like for a kanban team.

5. Common Challenges Observed

“I have not failed. I’ve just found 10,000 ways that won’t work.” – Thomas Edison

5.1. Coaching Teams

I failed to coach the first team using a kanban system at SEP with a consideration for their maturity. Without an understanding of the Dreyfus Model and how teams acquired skills I started with principles, few rules, and firm WIP limits. After struggling for a little while, I changed my approach. I began with a small set of rules and practices. From there we progressed at a pace that felt comfortable to the team. The results were much better and followed a pattern consistent to the model.

5.2. Using Little’s Law to Set WIP Limits

Little’s Law states: $\text{Throughput (TP)} = \frac{\text{Work-In-Progress (WIP)}}{\text{Cycle Time (CT)}}$. Because we work in a contracted outsourcing environment, desired throughput can often be calculated from the contract or statement of work. Using TP and the initial WIP limits, we calculated the desired cycle time for work items that would lead to successfully meeting budget and schedule goals. Cycle time could then be used as a leading indicator to monitor our progress. However, there was an underlying assumption that WIP would remain constant – a false assumption. When work in progress dropped below the limits, the team could continue to hit cycle times, but would fall short of the total throughput number – the number that mattered most from a contractual and reputation perspective. Therefore, when using Little’s Law, it is important that the formula be adjusted periodically as WIP limits change.

5.3. WIP Limits Too Low For a New Team

After having initiated several kanban systems at SEP, I setup a new team with what seemed like reasonable WIP limits. They were a little lower than I had used on previous teams, but I thought they would encourage the correct habits from the start. They didn't. With the WIP limits set too low, the team just struggled. When a work item became blocked, they didn't have the maturity or experience to collaborate and team up to complete current work in progress. Therefore, team members would try to start new work items. When no new work could be started, they were left trying to figure out what to do. Team members were frustrated and felt unproductive. Depending upon the specific context and situation, we address this today by taking a conservative approach with our WIP limits.

Corey Ladas, in his Lean Software Engineering blog, suggest three ways to set queue sizes: (1) Start every limit at one. Add tokens one at a time until one person is always busy, then apply the Theory of Constraints. (2) Start every limit at an arbitrarily large value, like 10. Subtract tokens 1 at a time until flow is observed. Then start looking for a way to remove 1 more. (3) Create a Value Stream Map and measure the time-on-task distribution of each activity. Use Little's Law to calculate the corresponding queue sizes. [6]

5.4. Blocked Issues Resulting from Clients

Blocked work items create real problems and struggles for the development team. Most of the time they were resolved by applying team resources to the issue. However, when the blocked work item resulted from a client, a different problem emerged. The blocked items piled up and stopped flow. Team members were left wondering what to work on. They had learned that exceeding WIP limits often resulted in more rework and longer lead times.

Unfortunately, we have not figured out a good answer to this problem. One team took blocked work items and placed them in a holding area. The team monitored lead times and quality metrics for leading indicators of potential problems. They proceeded in this way throughout the life of the project. This particular project was a verification project; I'm not confident it this would work for a software development project, but it was successful in this particular situation.

Ideally, I would like to use the information gained through monitoring lead times and throughput in conjunction with blocked work items to educate our clients on how blocked work affects their projects.

6. A Project Success Story: How Kanban Saved EHMS

I would like to share how one particular project at SEP was impacted by using a kanban system. Pay close attention to how the team matured in accordance with the Dreyfus Model.

Prior to starting the EHMS project, there was a long standing, small (4 people), successful team in place. They had been developing similar web applications for several years and well versed in how they did things. The team rarely worked in a strict deadline environment. They were very comfortable with their approach to this type of system and our client was very happy as well.

At the start of the EHMS project, the team experienced significant changes. The size of the team ballooned to three times its original size. The technology platform the team was accustomed to changed. For the first time, there was a high pressure, politically charged deadline. In spite of all these changes, the team did not try to change the process they were familiar with. They continued to work in the same manner as before.

A couple of months into the project, things were not proceeding well. The lead did not have a feel for what people were working on. It was unclear to anyone how the project was progressing. Discontent and frustration was felt throughout the team. There was lots of negative emotion. No one was stepping up to fix the bigger problems the team was encountering – only heroes dealing with the symptoms.

Midway through the project, a kanban system is introduced. At first, the team experienced minor confusion but quickly aligned with some coaching. Immediately, the kanban system highlighted a major issue: they had an enormous amount of work in progress. Even though the team had not implemented WIP limits yet, they realized very quickly that WIP needed to be reduced. The kanban system had highlighted the current state of project. Even though it wasn't ideal, team members started to experience a little hope.

A short time later, the team hit the first delivery milestone. However, it had sacrificed quality to do so. After the initial delivery, strict WIP limits were enforced and prompted some immediate changes.

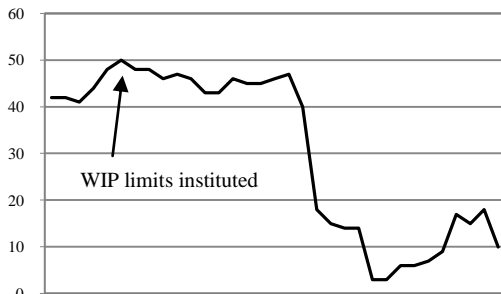


Figure 2 - Work in Progress

As a result, the team “took the pain” and fixed the quality defects along with the process issues. People who were normally quiet began to point out issues and come up with different solutions. The team began to take ownership of the process and began down the road of continuous improvement. As the work in progress was reduced, the lead times for work items was also significantly reduced.

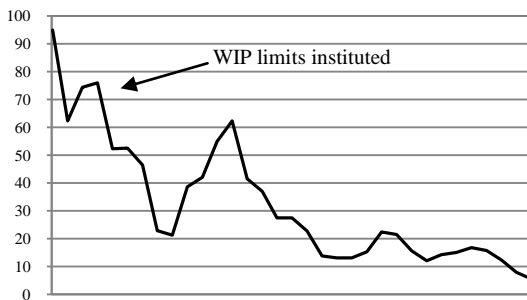


Figure 3 - Lead Time

Additionally, the number of bugs identified went down as work in progress and lead time were reduced.

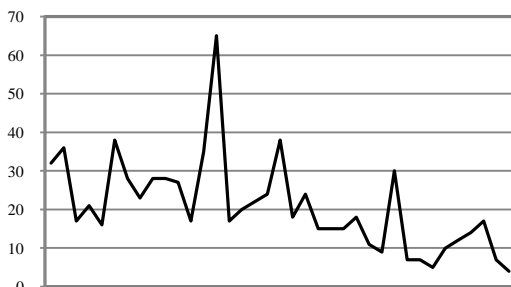


Figure 4 - Bug Injection Rate

Six months later, the team enforces good behavior in their day to day activities. The kanban board has changed multiple times and fits the team’s current situation well. Overall, there is a sense of progress felt throughout the team which has led to constructive attitudes. The team has become a bunch of “true believers”. They have started to really look at the data,

see what they can learn from it, and begin to manage objectively. Kyle, the EHMS project manager, said, “I wish we’d started kanban six months earlier.”

7. Conclusion

The Dreyfus model provides a useful topology for developing skills. It can be effectively applied to teams using a kanban system as a way to measure team maturity.

Everyone fundamentally understands that having the knowledge of an activity doesn’t mean you have the skill necessary to perform the activity. The Dreyfus model helps us to understand how to acquire the skills necessary to accomplish a goal. One of the fundamental concepts in the model is the placement of practices before principles.

Lean principles are best understood by applying the principles through practice. Kanban proved to be an effective tool for teaching these principles. It provided a way for teams to discover and practice the principles through application.

Finally, the model defines multiple stages of maturity. In each stage, certain behaviors guide your focus for improving. This information is invaluable for coaches and trainers as they work with teams using kanban systems. The knowledge of how a team matures helps to guide their efforts and training to help teams mature in the most efficient way.

8. Acknowledgements

9. References

- [1] Stuart E. Dreyfus, Hubert L. Dreyfus, “A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition”, Operations Research Center, University of California, Berkeley, February 1980, pp. 1, 7-8
- [2] Andy Hunt, *Pragmatic Thinking and Learning: Refactor Your “Wetware”*, Pragmatic Bookshelf, October 2008
- [3] P. Benner, *From Novice to Expert: Excellence and Power in Clinical Nursing Practice*, Menlo Park: Addison-Wesley, pp. 13-34
- [4] Stuart E. Dreyfus, “The Five-Stage Model of Adult Skill Acquisition”, *Bulletin of Science Technology and Society Vol. 24 No. 3*, SAGE, June 2004, pp. 177-181
- [5] W. Edwards Deming, *Out of the Crisis*, Massachusetts Institute of Technology Center for Advanced Engineering Study, Cambridge, Massachusetts, 1982
- [6] Corey Ladas, “Three Ways to Go About Queue Sizing” <http://leansoftwareengineering.com>, April 9, 2009