Dreyfus's ladder of competence[1]			
LEVEL	**Description**	**Description Learning Modes (% Embodiment)***	**Software Engineering Examples**
Novice (beginner)	Just getting started in the domain. All action appears to be governed by rules defining allowable moves and strategies. Common situations are unfamiliar and are described by more rules.	Memorization, drill, and simple practice. Demonstrations of play. Practice in simple situations (0%).	Starting programmer. Focuses on syntax, compilation, simple debugging. Basic concepts of objects. Basic algorithms. Basic program design, software methods.
Advanced Beginner (rookie)	Recognizes common situations that help in recalling which rules should be exercised. Most action is deliberate application of rules or conscious recall of prior actions in the familiar situations. Can perform simple actions for customers; needs supervision for more complex tasks.	Problem-solving and practice with rules and strategies. Play in realistic situations with supervision. Repeated practice with common situations (30%).	Comfortable with syntax. Composes basic programs to solve problems up to several pages and tens of modules. Can write simple programs for customers. Works well with direction.
Professional (competent)	Carries out standard actions without causing breakdowns. Can fulfill standard promises to customers satisfactorily without supervision. Performs most standard actions without conscious application of rules. When faced with a new situation, works out appropriate actions by application of rules.	Advanced problem-solving, coaching on problem-solving and projects. Extensive practice in both common and exceptional situations. Apprenticeship to more advanced professionals and teams. Membership in professional networks (60%).	Skilled in multiple languages. Deals with programs of hundreds of modules. Designs systems and test protocols, integrates components. Helps customers solve system design and configuration problems. Can work on teams and with customers. May be a team leader.

* Percentages are suggestive, not quantitative

[1] **Peter J. Denning** The Profession of IT, Career Redux, How can one design a career when career as an institution is dead? Entrepreneurs have an answerCOMMUNICATIONS OF THE ACM (pág. 21 a 26) , September 2002/Vol.45,No. 9.

Proficient Professional (star)	Deals with complex situations effortlessly. Seldom thinks in terms of rules and may have some difficulty telling others what rules he or she works with. Appropriate action appears to come from experience and intuition, and is deliberately chosen. Individual performance is a benchmark for others. Considerable experience and practice across a wide range of situations over years of work.	Apprenticeship to experts.Coaching. Putting self into wide range of situations. Membership and contribution to professional networks. Teaches others (80%).	Highly productive. Designs and manages complex systems. Ingenious solutions. Clear code. Excellent problem-solver. Productivity much higher than average. Receives positive assessments from customers and other professionals.
Expert (virtuoso)	Consistently inspiring and excellent performances. Appears to solve difficult, complex problems effortlessly. Enormous breadth and depth of knowledge. Acts appropriately without thought or conscious choice of actions. Routinely forms and leads high-performance teams; admired by others as a benchmark of team performance. Performance standards are well beyond those of most practitioners.	Apprenticeship to masters. Advanced coaching, development of breadth, focus on observing and adopting style of the teacher. Teaches others. Years or decades of practice (95%).	Extensive experience with large systems. Anticipates subtle and indirect design issues. Anticipates and responds to customer concerns. Leads teams well. High productivity. Solves difficult configuration and performance problems quickly.
Master	Capacity for long-range strategic thinking and action. Sees historical drifts and shifting clearings. Has studied with many different teachers and has developed own distinctive style. Has produced innovations in the standard practices of others, altered the course of history in the field, and knows how to do this again. Teaches others to be experts and masters.	Learning continues by working with other masters as teachers. Creates and leads professional networks. Teaches others (100%).	Develops new methods and practices for the field. Admired for long, historical perspectives and strategies.
Legend	Has attained high public standing with almost mythical status as a master and performer. Leverages public standing to achieve results only public figures could attain. Work has widely accepted impact.	Same as for master with emphasis on public appearance (100%).	Widely admired software engineer who publicly set the pace for every-one else. His or her articulations shape the direction of the field.