**The Principles of the Kanban Method**

Posted on December 10, 2010 by David J Anderson

**Start with what you do now**  
**Agree to pursue incremental, evolutionary change**  
**Respect the current process, roles, responsibilities & titles**

Then (use the 5 Core Properties) ...

**1. Visualize the workflow  
2. Limit WIP  
3. Manage Flow  
4. Make Process Policies Explicit  
5. Improve Collaboratively (using models & the scientific method)**

Let's examine these elements one-by-one

**Start with what you do now**  
The Kanban Method does not ask you to change your process. It is based on the concept that you evolve your current process. There is no sweeping, engineered change to a new process definition or style of working. There is no such thing as the Kanban Software Development Process or the Kanban Project Management Method.

**Agree to pursue incremental, evolutionary change**  
The organization (or team) must agree that their current circumstances warrant a gentle, evolutionary approach to improvement. Perhaps a sweeping engineered change has recently failed due to resistance from team members, or perhaps the politics of the organization make it too risky for managers to propose and implement sweeping changes? Without agreement that a slow, gentle, evolutionary, incremental approach is the right way forward then there won't be the right environment or management support for a Kanban initiative.

**Respect the current process, roles, responsibilities and job titles**  
It is likely that, what the organization currently does, has some elements that work acceptably and are worth preserving. We must also seek to drive out fear in order to facilitate future change. By agreeing to respect current roles, responsibilities and job titles we eliminate initial fears. This should enable us to gain broader support for our Kanban initiative. Perhaps presenting Kanban against an alternative more sweeping approach that would lead to changes in titles, roles, responsibilities and perhaps the wholesale removal of certain positions will help individuals to realize the benefits.

**Visualize the Workflow**  
Luckily in my book, I chose not to use the Lean term Value Stream. It seems this term carries with it a metaphor that some have objected to in software development. There has been an argument that Value Creation Network is a more appropriate term. Janice Linden-Reed and I in our DZone RefCard used this term. I've come to realize, however, that it only adds to the confusion and the complexity. The Value Creation Network is really the interactions of all the people involved in creating software or solving IT Operations problems, whereas, what I am interested in with the Kanban Method is actually the information arrival process related to the work itself.

So typically, we are looking for state changes in the work, that generally reflect changes in the activity used to generate new information about that work, for example, analysis (an activity) generates information, and when it reaches a point of diminishing returns, we tend to refer to the work as "analyzed" and change to a different activity to generate further information such as design or test development. It is this process of punctuated information arrival that we seek to model when I ask us to Visualize the Workflow.

**Limit WIP**  
Limiting WIP implies that we implement a pull system on part or all of the workflow. The pull system can be a kanban system, a [CONWIP](http://en.wikipedia.org/wiki/CONWIP) system, a [DBR system](http://en.wikipedia.org/wiki/Theory_of_Constraints), or some other variant. The critical elements are that work-in-progress at each state in the workflow is limited and that new work in "pulled" into the new information discovery activity when there is available capacity within the local WIP limit.

**Manage Flow**  
The flow of work items through each state in the workflow should be monitored and reported - often referred to as Measuring Flow. By flow we mean movement. We are interested in the speed of movement and the smoothness of that movement. Ideally we want fast smooth flow. Fast smooth flow means our system is both creating value quickly, which is minimizing risk and avoiding (opportunity) cost of delay, and is also doing so in a predictable fashion.

**Make Process Policies Explicit**  
Until the mechanism of software development or IT operations process is made explicit it is often hard or impossible to hold a discussion about improving it. Without an explicit understanding of how things work and how work is actually done, any discussion of problems tends to be emotional, anecdotal and subjective. With an explicit understanding it is possible to move to a more rational, empirical, objective discussion of issues. This is more likely to facilitate consensus around improvement suggestions.

**Improve Collaboratively (using models/scientific method)**  
It is the WIP limit that ultimately stimulates conversations about process problems. Things which impede flow, or introduce perturbations that mean flow is inconsistent or ragged, often result in a challenge to the WIP limit. The team has the option to break the limit, ignore the problem and carry on, or to face up to the issue, discuss it and suggest a change.

When teams have a shared understanding of theories about work, workflow, process, and risk, they are more likely to be able to build a shared comprehension of a problem and suggest improvement actions which can be agreed by consensus. In my book, I suggested three models which are useful: The Theory of Constraints (the study of bottlenecks); The Theory of Profound Knowledge (a study of variation and how it affects processes); and the Lean Economic Model (based on the concepts of "waste" (or muda, muri and mura)). Other models are possible such as Real Option Theory.

The use of models allows a team to make a prediction about the affect of a change (or intervention). After the change is implemented the outcome can be observed by measuring the flow and examining the data. The outcome can be compared to the prediction expected from the model and the change can be assessed as an improvement, or not. This process of evaluating an empirical observation with a model, suggesting an intervention and predicting the outcome based on the model, then observing what really happens and comparing with the prediction, is use of the scientific method in its fundamental sense. This scientific approach, is I believe, more likely to lead to learning at both the individual and organizational level. Hence the use of the scientific approach in Kanban will lead directly to the emergence of learning organizations.

**Complex Adaptive Systems**  
The Principles of the Kanban Method are designed to lay the foundation for an organization that can improve incrementally by setting out the conditions that will stimulate improvement. The improvements are emergent behavior. The outcome cannot be predicted. All that can reasonably be predicted is that things will change. As such the Kanban Method represents a Complex Adaptive Systems approach to leading change in organizations. Complex Adaptive Systems use simple rules and seed conditions to stimulate emergent behavior. The simple rules are embedded in the 5 Core Properties and the seed conditions are represented in the first principles of starting with the current process, gaining agreement to pursue incremental improvement and respecting current work practices, roles, responsibilities and job titles. What will happen next is emergent change. Beyond that we cannot predict. The system must be monitored and adapted. The simple rules being used such as WIP limits and workflow visualization must be tweaked and changed to steer the emergent behavior to produce a desirable outcome.

**The System Designer Role**  
This ability to steer the simple rules of the system and to make changes to the process implementation to produce a desired outcome, as opposed to an undesired one, is a skill. That skill is unlikely to reside in every member of a team. More likely someone takes on the role of system designer, or facilitator of the system design. This person may be a manager, a process engineer, or an external coach. The system designer must facilitate the implementation of the Kanban initiative and the design of the kanban system. The system design must change as the complex adaptive system adapts over time in response to the emergent behavior and the outcomes it is producing.