Kanban: Naturally suited for Enterprise Adoption

Boston SPIN
Ajay Reddy
CIO, CodeGenesys.com, ScrumDo.com

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
Is Kanban heralding in a new era?

It’s the end of methodology!*

Reflective Improvement Frameworks** are the future!

Kanban is such a Reflective Improvement Framework

* http://alistair.cockburn.us/The+end+of+methodology
** Cockburn’s suggested name for this new class of methods
*** End of Methodology slides credit to: David Anderson LKNL

Key note

ajay@codegenesys.com @ajrdy @scrumdo @codegenesys
A methodology defines behavior

• A software engineering methodology is a description of techniques
  o what to do
  o how to do it
  o When to do it - sequences or workflows
  o Who does what - definition of roles and responsibilities

• Ideally, a methodology should tell us why and give us a context to define its appropriateness
Many styles of software engineering methodologies emerged over several decades.

- Some just personal preferences in style (e.g. PSP versus XP), but others for specific contexts or risk profiles (e.g. the many risk profiles captured in a 2-dimensional grid in Cockburn's Crystal methods).
- Some styles came in schools or movements - such as the Agile movement.
- While others came as large frameworks such as Rational Unified Process designed to be tailored to a context.
Managing change has greater leverage than picking the right methodology.

Instead the bigger challenge with the greater leverage on outcome was learning to manage change in the organization.
Traditional Change is an A to B process

- A is where you are now. B is a destination.
  - B is either defined (from a methodology definition)
  - or designed (by tailoring a framework or using a model based approach such as VSM* or TOC TP**)
- To get from A to B, a change agency*** will guide a transition initiative to install B into the organization

* Value stream mapping, ** Theory of Constraints Thinking Processes
*** either an internal process group or external consultants

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
Change initiatives fail (even) more often than projects

Change initiatives often fail (aborted) or produce lack luster results

They fail to institutionalize resulting in regression back to old behavior
Change Challenges

- Should it change
- Who changes
- What changes
- How fast does it change
- Why it changes
- What direction it changes
- How much does it change
- How sustainable is the change
- How uniform is the change
- When does it change
- How often does it change
- Where does it change
- How self perpetuating is the change

- Prescriptive vs adaptive
- Evolutionary vs Transformation

- Size of change
- Speed of change
- Source of change

ajay@codegenesys.com  @ajrdy  @scrumdo  @codegenesys
Daniel Kahneman has given us a simple model for how we process information.

**System 1**
- Sensory Perception
- Pattern Matching
- Learning by Experience
  - FAST
  - But slow to learn

**System 2**
- Logical Inference Engine
- Learning from theory
  - SLOW
  - But fast to learn

Daniel Kahneman
ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
How we process change...

I logically evaluate change using System 2

I adapt quickly

I feel change emotionally using System 1

I adapt slowly

Silicon-based life form

Carbon-based life form

Daniel Kahneman

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
Changing methodologies challenges people psychology & sociologically.

- New roles (defined in a methodology) attack their identity.
- New responsibilities using new techniques & practices threaten their self-esteem and put their social status at risk.
- Most people resist most change because individually they have more to lose than to gain.
- It is safer to be conservative and stick to current practices and avoid shaking up the current social hierarchy.
- *Only the brave, the reckless or the desperate will pursue grand changes.*
The Definition Of
The Kanban Method
The Kanban Method...

- Rejects the traditional approach to change
- Believes, it is better to avoid resistance than to push harder against it – Don’t install a new methodology
- Is designed for carbon-based life forms - Evolutionary change that is humane
The Kanban Method…

• Catalyzes improvement through use of kanban systems and visual boards*
• Takes its name from the use of kanban but it is just a name
• Anyone who thinks Kanban is just about kanban (boards & systems) is truly mistaken

*also known as "kanban" in Chinese and in Japanese when written with Chinese characters

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
The Kanban Method is a new approach to improvement

Kanban is a

method

without methodology
Kanban Method

A management & cultural approach to improvement

View creative knowledge work as a set of services

Encourages a management focus on demand, business risks and capability of each service to supply against that demand

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
Principles

Start with what you do now

Agree to pursue evolutionary change

Initially, respect existing roles, responsibilities and job titles
The Kanban Method is not...

A project management or software development lifecycle process
The Kanban Method is...

A method for directly improving service delivery

A mechanism for catalyzing continuous improvement
Kanban Method

Uses visualization of invisible work and kanban systems

Develops an adaptive capability in your business processes, enabling you to respond successfully to changes in your external business environment
Core practices of the Kanban method

• Visualize (with a kanban board 看板)
• Limit Work-in-progress (with kanban かんぱん)
• Manage Flow
• Make Policies Explicit
• Implement Feedback Loops
• Improve Collaboratively, Evolve Experimentally (using models & the scientific method)

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
Case Studies
System

- Leadership structure
- Executive Expectation
- 70+ member team
- Business artificially satisfied
- High degree of specialization
- History with process
- Politics
- Enterprise Data Movement

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
System

- 8 different demand sources
- Demand negotiated down with artificial show of capacity
- Waste explained as necessary evil
- Varied methods in the same team
- Shared resources unprotected
- Very few feedback loops
System

- Heavy Documentation
- Invisible initiatives
- Dependencies
- Relatively slow moving in adaptation
- Limited options for external influence
- High degrees of specialization in skills
- Hierarchical
- Separation of concerns
- Local optimization
- Relatively bigger buying power
- Rewards individual without real consideration of team
- Less tolerance for proxy measures
- Less tolerance for risk

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
CEO says “Pursue this”

- Time to market
- Improve Predictability
- Simplicity
- Sustainability
- Enterprise Agility
History

- ‘Agile 1’
- ‘Agile 2’
- ‘Agile 3’
Quotes from David

• I came to discover this technique as I perceived that people, companies, and organizations were resisting the adoption of Agile software development methods. The idea that you try and persuade people to switch to some defined method that’s written in a textbook, and make that switch all at once, was prevalent particularly in the earlier part of the last decade. This approach is what people were resisting.

• Agile methods were really sold as an all or nothing bet. They were marketed as fragile ecosystems where you had to, for example, do all 12 practices in Extreme Programming or it wasn’t guaranteed to work. So adopting all 12 of those practices all at once was very challenging, and there was really no guidance on how to incrementally adopt the practices in Extreme Programming in those early days. To be honest, even today, there are very few Agile coaches and consultants who are capable of correctly assessing a context and advising a client on incremental adoption.

• The critical difference is that Kanban is not a method, like an Agile software development method, or a project management method. It’s a way of provoking and catalyzing changes in an organization. What those changes might be is contextual, situational and will be unique for each different organization that tries it.

• The idea that Kanban is a meta-method for incremental evolutionary change, maybe that’s accurate!
Dependencies between teams create demand

Kanban Adoptions Starts Here

Product Strategy

App Dev 1

App Dev 2

App Dev 3

Demand

Service

Platform Development

Viral Spread

@ajrdy @scrumdo @codegenesys
Demarkate waiting area(s) for external dependencies

5

4

3

4

2

2 = 20 total

Input Queue

Analysis

Dev

Development

Build

Release

Queue

In Prog

Ready

In Prog

Done

In Prog

Done

Ready

Test

... Ready

Waiting on External Group

Dots denote clock ticking on SLA

Late against SLA

ajay@codegenesys.com @ajrdy @scrumdo @codegenesys
Emergence of Service-Oriented Organizational Structure

- **Tag Work Items with Shared Resource Dependency**
  - Clearly mark as impeded
  - Provides visibility onto the problem

- **Provide shared resource with separate Kanban board/system**
  - Offer SLA and track while waiting
  - Escalate late items using issue management system

- **Treat Integration Items as Fixed Date class of service**
  - Base fixed date on planned integration point
Each team that receives demand from another treats them as a customer and designs a service to supply them (from first principles)
Symptoms of organic scaling

- Flatter structure - Exit of PM
- Executive Expectation - Focus on the system
- 4 teams inter operating with SLEs
- Business – Focus on the system
- Shared Resources
- Contagious
- Exit of 1 PM, repurposing of other team members - Some protections for the team
- Management

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
Why it was a good fit

- Does not demand
- Team owned
- Respects roles, functions and realizes real incremental, realistic, sustainable growth
- Visualization of inventory, waste, value, progress
- Low costing enforcement of policy
- Keeping business and teams honest - Effectively balances demand and capacity
- Focus away from proxies and getting close to actuals
- Real cross functions
- Jinka Gudoki Kaalu Ethi Chuupichindanta

ajay@codegenesys.com  @ajrdy @scrumdo @codegenesys
Following Data from Kanban at Scale Instance in Seimen's Healthcare-Dan Vacanti - ajay@codegenesys.com @ajrdy @scrumdo @codegenesys
CFD at story level
Benefits of Kanban Systems...

Visibility

Eliminates overburdening

Reduces or eliminates multi-tasking

Controls or eliminates interruptions, disruptive task-switching and variability

Shorter lead times

Better quality

Deferred commitment
...and the Results it Produces

Improved predictability

Improved business agility
- more frequent selection & commitment, more frequent delivery, short lead times

Improved governance & risk management

Adaptive capability – the ability to evolve processes & workflows in response to a changing external environment
Thank you

Training events:

codegenesys.com/events/

Careers:

codegenesys.com/jobs/

ajay@codegenesys.com  @ajrdy  @scrumdo  @codegenesys