

Kanban (development)

From Wikipedia, the free encyclopedia

Kanban is a method for managing knowledge work with an emphasis on just-in-time delivery while not overloading the team members. In this approach, the process, from definition of a task to its delivery to the customer, is displayed for participants to see and team members pull work from a queue.

Kanban can be divided into two parts:

- Kanban – A visual process management system that tells what to produce, when to produce it, and how much to produce.
- The Kanban method – An approach to incremental, evolutionary process improvement for organizations.

Contents

- 1 The Kanban method
- 2 Six core practices
- 3 Example
- 4 See also
- 5 References
- 6 External links

The Kanban method

The name 'Kanban' originates from Japanese[看板], and translates roughly as "signboard" or "billboard". The **Kanban method** as formulated by David J. Anderson^{[1][2]} is an approach to incremental, evolutionary process and systems change for organizations. It uses a work-in-progress limited pull system as the core mechanism to expose system operation (or process) problems and stimulate collaboration to continuously improve the system. One example of such a pull system is a kanban system, and it is after this popular form of a work-in-progress, limited pull system that the method is named.

The Kanban method is rooted in four basic principles:

Start with what you do now

The Kanban method does not prescribe a specific set of roles or process steps. The Kanban method starts with the roles and processes you have and stimulates continuous, incremental and evolutionary changes to your system. The Kanban method is a change management method; there is no such thing as a Kanban software development process or a Kanban project management.

Agree to pursue incremental, evolutionary change

The organization (or team) must agree that continuous, incremental and evolutionary change is the way to make system improvements and make them stick. Sweeping changes may seem more effective but have a higher failure rate due to resistance and fear in the organization. The Kanban method encourages continuous small incremental and evolutionary changes to your current system.

Respect the current process, roles, responsibilities and titles

It is likely that the organization currently 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.

Leadership at all levels

Acts of leadership at all levels in the organization from individual contributors to senior management should be encouraged.

Six core practices

Anderson identified five core properties that had been observed in each successful implementation of the Kanban method.^[2] They were later relabeled as practices and extended with the addition of a sixth.

1. Visualize

The workflow of knowledge work is inherently invisible. Visualising the flow of work and making it visible is core to understanding how work proceeds. Without understanding the workflow, making the right changes is harder. A common way to visualise the workflow is to use a card wall with cards and columns. The columns on the card wall representing the different states or steps in the workflow.

2. Limit WIP

Limiting work-in-process implies that a pull system is implemented on parts or all of the workflow. The pull system will act as one of the main stimuli for continuous, incremental and evolutionary changes to your system. The pull system can be implemented as a kanban system, a CONWIP system, a DBR system, or some other variant. The critical elements are that work-in-process at each state in the workflow is limited and that new work is “pulled” into the new information discovery activity when there is available capacity within the local WIP limit.

3. Manage flow

The flow of work through each state in the workflow should be monitored, measured and reported. By actively managing the flow the continuous, incremental and evolutionary changes to the system can be evaluated to have positive or negative effects on the system.

4. Make policies explicit

Until the mechanism of a 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.

5. Implement feedback loops

Collaboration to review flow of work and demand versus capability measures, metrics and indicators coupled with anecdotal narrative explaining notable events is vital to enabling evolutionary change. Organizations that have not implemented the second level of feedback - the operations review - have generally not seen process improvements beyond a localized team level. As a result, they have not realized the full benefits of Kanban observed elsewhere.

6. Improve collaboratively, evolve experimentally (using models and the scientific method)

The Kanban method encourages small continuous, incremental and evolutionary changes that stick. 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.

The Kanban method suggests that a scientific approach is used to implement continuous, incremental and evolutionary changes.

Common models used are:

- Theory of constraints (the study of bottlenecks)
- Deming System of Profound Knowledge (a study of variation and how it affects processes)
- Lean economic model (based on the concepts of “waste” (or muda, muri and mura)).

Example

Kanban Software Development Workflow ^[3]															
⇐ Waterfall model ⇒ (sets time and budget constrains)															
Workflow ⇒	Inbox	Specification		Ready for Development	Development (i.e. using Scrum and XP)			Code Review		Test on Local System		Test on Pre-Production System		Ready for Deployment	Deploy
WIP Limit ⇒	5	2		2	3			2		2		2			
Feature		In progress	Done		Planned	In Progress	Done	In progress	Done	In progress	Done	In progress	Done		
Login	User Story 567 User Story 214		User Story 857				User Story 654				User Story 75			User Story 754	
Register				User Story 244		User Story 751									
Password Recovery	User Story 624					User Story 245			User Story 782						
...
Billing			User Story 657	User Story 38					User Story 858						
Policies ⇒	<ul style="list-style-type: none">Product Owner writes user storiesDevelopment Team estimates user story size (planning poker)Prioritize storiesNote lead start date		Write acceptance criteria	Plan developer pairing	Note cycle start time	TDD and Refactoring	Note cycle end time	Check Policies <ul style="list-style-type: none">AntipatternsUnit testsIntegration testsCode coverageDeployment issues		Tester and Product Owner needed		Check only code functionality			<ul style="list-style-type: none">RemoveNote lead dateReview deploymentUpdate<ul style="list-style-type: none">Change fieldDeploymentLeadCodeBuildScriptReprioritize stories to new estimate

See also

- Kanban
- Kanban board
- Teamwork
- Lean software development
- List of software development philosophies

References

1. ^ Anderson, David (September 2003). *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*. Prentice Hall. ISBN 0-13-142460-2.

http://en.wikipedia.org/wiki/Kanban_(development)

3/4

2. [^] ^{*a b*} Anderson, David (April 2010). *Kanban - Successful Evolutionary Change for your Technology Business*. Blue Hole Press. ISBN 0-9845214-0-2.
3. [^] Jasper Boeg (2012-02). "Priming Kanban" (<http://www.infoq.com/minibooks/priming-kanban-jesper-boeg>) (in English). InfoQ. Retrieved 2014-02-17.

External links

- Aspects of Kanban (<http://www.methodsandtools.com/archive/archive.php?id=104>) by Karl Scotland, Retrieved on 2011-02-17
- De-mystifying Kanban (<http://www.netobjectives.com/files/resources/articles/Demystifying-Kanban.pdf>) by Al Shalloway of Net Objectives (<http://www.netobjectives.com>)
- Kanban Applied to Software Development: from Agile to Lean (<http://www.infoq.com/articles/hiranabe-lean-agile-kanban>)
- What is Best, Scrum or Kanban? (<http://agile.techwell.com/articles/weekly/what-best-scrum-or-kanban>)

Retrieved from "[http://en.wikipedia.org/w/index.php?title=Kanban_\(development\)&oldid=604257483](http://en.wikipedia.org/w/index.php?title=Kanban_(development)&oldid=604257483)"

Categories: Agile software development | Software development philosophies

-
- This page was last modified on 15 April 2014 at 05:24.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.