

Jon Kern's Agile Manifesto Notes – Feb 2001, Snowbird, Utah

Looks like we spoke about our goals...

Jon K

1.	Benikum	Arie	Consortium	DSDM
2.	Beck	Kent	kentbeck@csi.com	revolutionize social structure of technical people
3.	Beedle	Mike	beedlem@fti-consulting.com	UNIFICATION of synergistic methods
4.	Cockburn	Alistair	alistair.cockburn@acm.org	MULTI-METHODS
5.	Cunningham	Ward	ward@c2.com	entrepreneurial methodologies
6.	Fowler	Martin	fowler@acm.org	by XP's works
7.	Grenning	James	grenning@objectmentor.com	improve
8.	Highsmith	Jim	jimh@adaptivesd.com	SCRUM, ADAPTIVE
9.	Hunt	Andy	andy@pragmaticprogrammer.com	
10.	Jeffries	Ron	ronjeffries@acm.org	w
11.	Kern	Jon	jk@togethersoft.com	
12.	Marick	Brian	marick@testing.com	Test, improve communication
13.	Martin	Robert	rmartin@objectmentor.com	
14.	Mellor	Steve	Project techniques vendor	Agile sold
15.	Schwaber	Ken	kschwaber@mindspring.com	SCRUM, process mgt.
16.	Sutherland	Jeff	jsutherland@virtmed.com	SCRUM, motivated developers
17.	Thomas	PragmaticDave	Dave@thomases.com	

james@satisfice.com, IvanBartolo@cs.com, kentbeck@csi.com, beedlem@fti-consulting.com, gbooch@catapult.com, alistair.cockburn@acm.org, ward@c2.com, fowler@acm.org, grenning@objectmentor.com, jimh@adaptivesd.com, andy@pragmaticprogrammer.com, ronjeffries@acm.org, jk@togethersoft.com, marick@testing.com, rmartin@objectmentor.com, kschwaber@mindspring.com, jsutherland@virtmed.com, Dave@thomases.com

- Educate community on power of LWP
- publish manifesto w/ key principles e.g. collaboration XP - pair
- mutual needs sprint - scrum
- why does LWP work, what are the characteristics of success?
- where can we apply the different practices
- can we provide guidelines for application of processes in context?
- goal to understand similarities and Δs of LWP's
- talk about things that are still puzzling
- can we address mismatch between process & personal values
- how can we help coordinate sales & marketing to support rapid release cycles
- shared values across departments are key.

As best as I can decipher my above notes (May have been answers to something like “What do you want to achieve/focus on?”). I obviously didn’t scribble down everyone’s comments...

Arie	DSDM
Kent	Revolutionize social structure of technical people
Mike	Unification of synergistic methods
Alistair	Multi-methods
Ward	Entrepreneurial methodologies
Martin	Surprised by XP’s worth
James	Improve
Jim	Scrum, adaptive
Brian	Test, improve communication
Steve	4GL Solution
Ken	Scrum, process management
Jeff	Scrum, motivated developers

Another list:

- Educate community on the power of LWP (Light-weight Processes)
- Publish manifesto w/ key principles (e.g., collaboration: XP-pair, Scrum-sprint)
- Mutual needs
- Why does LWP work? What are the characteristics of success?
- Where can we apply the different practices?
- Can we provide guidelines for application of processes in context?
- Good to understand similarities and differences of [each] LWP
- Talk about things that are still puzzling
- Can we address mismatch between process and personal values?
- How can we help coordinate sales and marketing to support rapid release cycles?
- Shared values across departments are key.

Probable timing of discussions, unless something major counters this suggestion:

7:30 - 12:15 Morning discussions
12:15 - 1:00 Lunch
1:30 - 4:30 Skiing or afternoon hot tub
5:30 - 7:30 Evening discussion
8:00 - Group dinner

WhatDoYouWantToAccomplishAtLwP2001

LwmMutualNeeds
LwmBranding?
LwmPromotion
LwmConsolidationOfPrinciples
LwmPresentations
LwmPaperAndArticleCollaborations
LwmBookCollaborations
LwmConference
LwmConsultingNetwork
LwmConsortium
LwmNewsletter?

LwmCmm -- find ways for a CMM level chaser to use LWMs.

LwmRup -- Find ways for a Committed RUP user to use LWMs

LwmWebsite -- a single place for people to go to find out about LWMs.

LwmManifesto -- A statement from the group to the industry about LWMs.

LwmName -- Finding a new name for LWM.

LwmManifesto Edit Text of this page

A Document describing what light weight methods are, who should use them, and what their benefits/costs are. Authored by all of us.

RobertMartin -- I want to get two things done. 1. Set up a website that people can go to to learn about LwPs?. 2. Create a manifesto that the group signs, describing the fundamental tenets of LwPs?, and recommending them to the industry at large.

LwmMutualNeeds

What are our mutual needs? What is it we are each attempting to protect or move forward? Understanding that, we can probably see how to work together, and without understanding that, we'll probably have trouble closing on the other agenda items.

I am interested in fending off the fat methodology armory, the vast quantity of RUP, Anderson, SEI, etc. salespeople putting ideas into CIOs' minds that they should have lots of paperwork to be "safe". countering the "I want a turnkey methodology, one that works straight out of the box" mentality.

Those are the two battles I could use help in. --

AlistairCockburn

The situation is probably even a bit worse than that -- not only are the CIOs convinced that need lots of paperwork to be "safe", but they need lots of bodies as well. Bodies which

I
Anderson, EDS, et al are more than happy to provide. With all those bodies, you need lots of managers. With so many managers, you need lots of paperwork. By now you're tracking development hours in six-minute increments and progress is slow, you add more people and the cycle just feeds itself.

I think there needs to be a major effort to put forth the ideas that "less is more" across the board in software development. Part of the minimalist approach would be that you need to "roll your own" processes, as any off-the-shelf process is so loaded with generality and bloated by committee that it ain't minimal, and hence not as effective as a lean-and-mean (or well-focussed and targeted) approach. FWIW, I think this is *the* key battle to be fought to advance the industry (and not just in the area of processes/methods, either.) --AndyHunt

My current thinking is to separate Understandings (Aha's if you will) from Techniques, from Project Policies. Thus, The Value of Iterative Development is an "understanding". Various techniques can be applied to manage it. It may or may not be a Policy to use iterative development. As an example.

This could help...

A. drop Techniques from the discussion entirely.

B. Name the various Understandings we draw each upon for our methodology work. Here we should find some commonality, and can merge, disagree, form subsets, etc.

C. Each methodology representative names the Policies that come along with that methodology. Here there should be a much smaller overlap.

D. Then we can discuss the variations of the Policies according to circumstances and understandings (pulling up some new Understandings we discover we draw upon).

e.g. Crystal has incremental development as a required understanding, and also as a policy. It also has a project debriefing and methodology-tuning workshop every roughly 6 weeks as a policy. It does not have pair programming (PP) as a policy.

XP has PP as a policy. I don't know if a post-iteration is an XP policy, but it's pretty close. Methodology tuning workshops are not XP policies, although some advanced practitioners may do them. XP draws upon Emergent Design as an understanding (this is one I don't have).

Alistair: an Understanding is "I understand that this works". It doesn't have to be a principle. E.g., I understand that incremental development works and its value (No, Intellectually, Viscerally). I understand that iterative development works and its value. I understand this emergent design things and that it works and its value (which, by the way, I don't). I understand that multiple methodologies are needed. I understand that methodologies can be tuned to the organization on the fly and that it is worth the effort to do so. I understand that having a user on the project is worth the cost.

I don't think many of those are principles. They simply allow the conversation to proceed. If someone doesn't understand one of them, they really can't participate in shaping a methodology based on it.... e.g., I understand Emergent Design intellectually

but not viscerally, and I don't really agree with it. So I can participate in some of the discussions about using emergent design, but not in others, and I certainly wouldn't set it as a project policy.

2

At XP Immersion 5, we listed about 15 Understandings needed to adopt XP... that was a tough list, imho. The value of that list is that when someone says, "I don't get that one (or I don't agree with it)", it tells you where the next conversation should go... It is a matter of sharing sufficient concept space to have the next stage of dialog. --AlistairCockburn

UnderstandingsForCrystal

incremental development (policy)
value of customer links (policy)
value of informal communications
value of same-room/near-room communications
just-in-time methodology construction (policy)
project and iteration retrospectives (policy)

note: for CrystalClear?,
make those two communications items into policy.

UnderstandingsForXp

incremental development (policy)
iterative development (policy)
short 1-4 week iterations (policy)
pair programming (policy)
rotating pairs (policy)
collective code ownership (policy)
nano increments / "episodes" (policy)
automated regression unit tests (policy)
test-first development (not policy)
emergent design
customer on staff (policy)
just-in-time requirements (policy)
customer controls requirements (policy)
strict code standards (policy)
value of same-room/adjacent-room communications (policy)

UnderstandingsForAsd

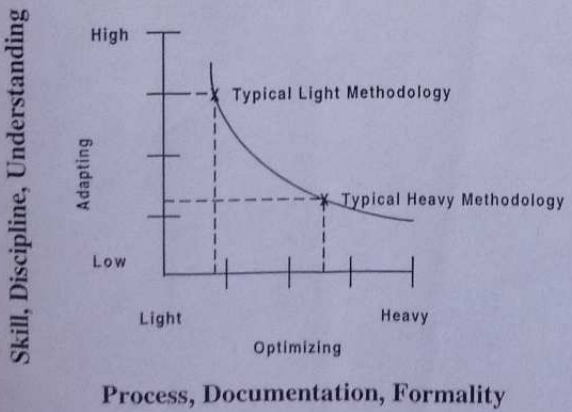
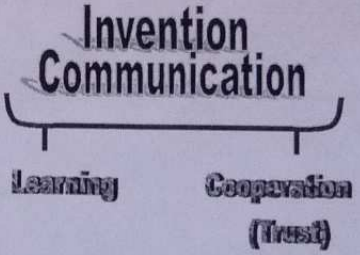
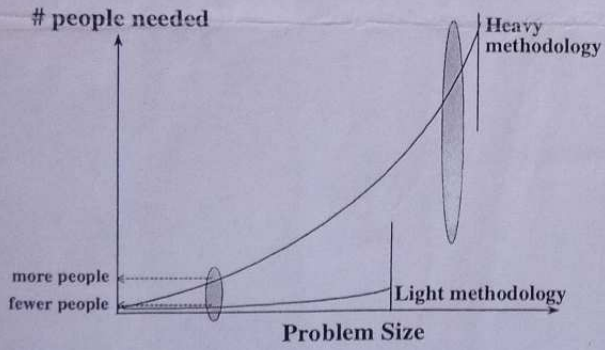
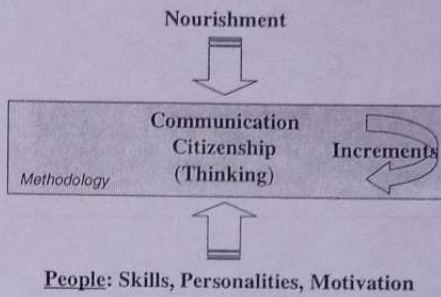
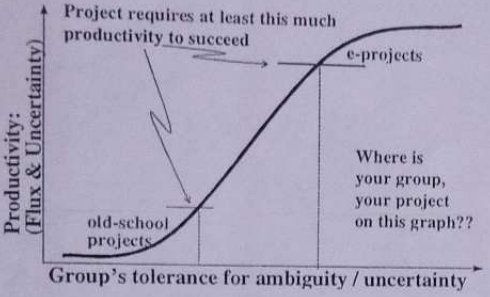
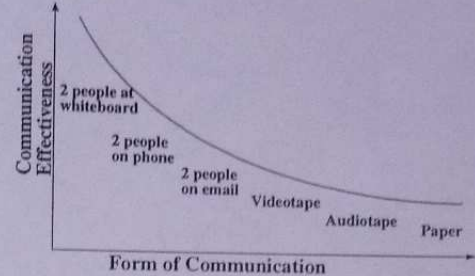
incremental development (policy)
light (varying by situation) team rules (policy)
emergent team behavior (policy)
tracking state of deliverables (policy)
timeboxing (?policy)

UnderstandingsForScrum

incremental & iterative development (policy)
short 1-3 week iterations (policy)
shifting project priorities (policy)
timeboxing (policy)

Project Characteristics

Criticality (defects cause loss of...)	Optimized for Traceability (Optimized for Productivity)						
	L3	L10	L30	L80	L150	L300	L600
Life (L)	L3	L10	L30	L80	L150	L300	L600
Essential money (E)	E3	E10	E30	E80	E150	E300	E600
Discretionary money (D)	D3	D10	D30	D80	D150	D300	D600
Comfort (C)	C3	C10	C30	C80	C150	C300	C600
	1-4	6-20	20-40	50-100	100-200	200-500	500+ total staffing sizes ±20%



Looks like I scribbled down process characteristics as each person spoke.

XP - write features, estimate, implement ~~week~~ by week
 Split in Business or Tech decision makers
 Tech must work in sustainable pace
 (6) Stories
 Iterations
 Releases
 OnSite Expert

(7) Test-first
 Continuous Integration
 Pairing
 Open workspace

comfortable size 4-12

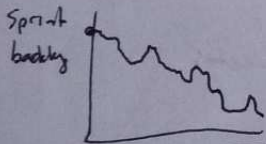
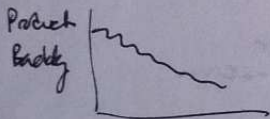
SCRUM - web site: controlled chaos

Rules: Backlog is a list of features that evolves over time
 Each item gets a rough estimate
 List is prioritized by a single person

Iterations: 30 days - balance between capability to produce & how long marketing stays quiet

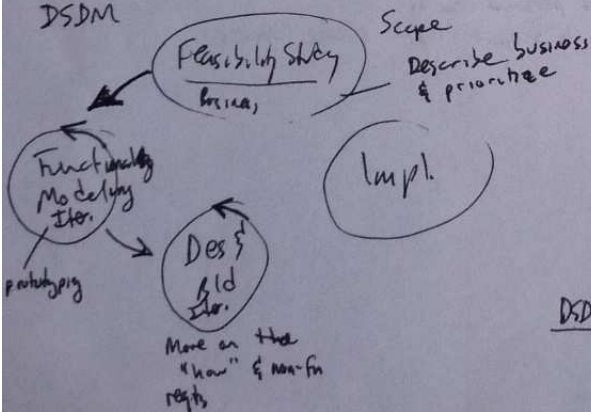
Mgt: chooses from the backlog and creates sprints Backlog that has a goal and are items that are 4-8 hours long.

Team Rules: meet once a day, project team only speaks
 what have I done yesterday? } Mgt is there to help
 what do I need to do now? } team succeed
 what's in my way?



every day, team can change estimate (of what's left)
 It is not time reporting

DSDM



Techniques

- Prototyping
- Integrated Testing
- Workshops
- Iterations - Investigate, refine, consolidate
- Vis. Team: Advisors, Visionary
- Dev. Team: Amb. ambassador, Visor, Dev, Scribe

MOSROW
 U H O A
 F S V N
 L U R
 b b

DSDM Consortium

www.dsdm.org

- Methods & Techniques
- Training / Certification
- Member Services
- Marketing & Promotion

Crystal: no single process works on all projects.
Cooperative game of invention & communication
Eg: docs & running code compete for resources
every 3-6 weeks have a process tuning, a how goes it.

Adaptive: Instead of anticipating the future (forecasting), must adapt over time better.
Another concept is that good teams will naturally organize
Emergent order from interaction, creativity & innovation is a result
speculate → collaborate → learn

Individual & collective skills are important. Learning from customer, peers are key.
How does this scale up? And try to keep it light. Not a process driven approach.
How can we share work artifacts across teams?
How do we switch from inflexible Command & Control to a better condition.