

Agile Process and XP

- the Waterfall model and formal process
 - goals, strengths, and weaknesses
- the Agile Process Rebellion
 - motivations and principles
 - eXtreme Programming
- Comparative Religion (agile vs. prescriptive)
 - similarities, differences, complementarities

1/30/2007

Agile Process and eXtreme Programming

2

The Wisdom of the Waterfall

- Basic Premise of the Waterfall Model
 1. understand the requirements
 2. develop a plan to satisfy them
 3. execute and manage against the plan
- Goals and Advantages
 - predictable functionality
 - predictable schedule and cost
 - minimal waste and minimal surprises

1/30/2007

Agile Process and eXtreme Programming

3

On Planning

The general who wins a battle makes many calculations in his temple ere the battle is fought. The general who loses a battle makes but few calculations beforehand.

Sun Tzu

No battle plan ever survived contact with the enemy.

Dwight Eisenhower

There is no great art to devising a good plan of operations. The entire difficulty lies in this: To remain faithful in action to the principles we have laid down for ourselves.

Carl von Clausewitz

1/30/2007

Agile Process and eXtreme Programming

4

Fallacy of the Waterfall Model

- Assumptions
 - we are executing a well crafted plan
 - plan is designed to satisfy correct requirements
- Good requirements may be unattainable
 - in which case the plan is wrong
 - it will deliver the wrong functionality
- Even an incremental model may fail
 - if 1st release doesn't deliver right functionality

1/30/2007

Agile Process and eXtreme Programming

5

Formal Process - the good news

- it institutionalizes best practices
 - techniques to avoid common mistakes
- it enables more effective management
 - better planning and measurement
 - clear responsibilities and milestones
- it is necessary for process improvement
 - following a defined process
 - capturing permanent records of what we did
 - enables process evaluation and evolution

1/30/2007

Agile Process and eXtreme Programming

6

Formal Process - the bad news

- it *can* place form over substance
 - people are goaled on process deliverables
 - real goals are customer satisfaction, R.O.I
- one size may not really fit all
 - bureaucracy greatly burdens small projects
 - not all steps & work-products are necessary
- it is a lowest-common-denominator solution
 - it can improve the work of weak teams
 - it can greatly limit strong contributors

1/30/2007

Agile Process and eXtreme Programming

7

The Agile Process Rebellion

- Don't put too much faith in paper process!
 - our only deliverables are working software
 - our key goal is customer satisfaction
- Prescribed processes are, per force, flawed!
 - continuous change is a given – deal with it!
 - the best process is collaboration
 - all stake-holders communicate regularly
 - frequent, small, updates and good feedback
- Overemphasis on task definition is myopic!
 - people, not processes, solve problems

1/30/2007

Agile Process and eXtreme Programming

8

People v.s. Process

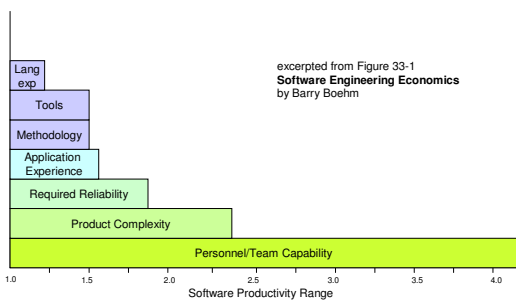
- Focus on creating good teams
 - ensure they have the right skill/perspective mix
 - ensure all the stake-holders are involved
 - ensure they are dedicated to the project
 - ensure they will work well **as a team**
- Then, let them do their job
 - don't tell them how to organize themselves
 - don't tell them how to best solve the problem
- Good teams beat good process, every time

1/30/2007

Agile Process and eXtreme Programming

9

The High Order Bits



1/30/2007

Agile Process and eXtreme Programming

10

Agile methods ...

- address people and teamwork issues
 - which are far too important to be ignored
- focus more directly on real goals
 - which is always a good thing
- put principles & methodology over process
 - which is almost surely right
- still enumerate required process activities
 - but avoid over-specifying tasks/deliverables
 - which makes sense for relatively simple projects

1/30/2007

Agile Process and eXtreme Programming

11

eXtreme Programming

- an agile development philosophy
 - embracing change, short feedback-driven cycles, high stake-holder involvement, communication intensive
- a set of techno-moralistic principles
 - personal courage, mutual respect, responsible behavior, sustainable pace, minimalism, incrementality, continuous integration
- a collection of development practices
 - requirements gathering, design, coding, testing

1/30/2007

Agile Process and eXtreme Programming

12

eXtreme Programming Practices

- Define Requirements through user stories
 - functional scenarios that fit on a 3x5 card
 - these are prioritized and mapped into tasks
- Pair Programming
 - two people working on a single task
 - one coding, one thinking, reviewing, planning
- Test Driven Development
 - implement features as they become testable
 - tests and features developed in parallel
- Continuous Integration

1/30/2007

Agile Process and eXtreme Programming

13

Comparative Religion - similarities

- They all follow the same basic process
 - understand the problem
 - start w/concept, gather & prioritize requirements
 - plan the solution
 - move from high level to more detailed design
 - prototype to reduce risk
 - execute the plan
 - implementing and testing proceed in tandem
 - monitor progress, look for problems, re-plan
- Agile approaches are intrinsically iterative

1/30/2007

Agile Process and eXtreme Programming

14

Comparative Religion - differences

	Planned approaches	Agile approaches
Stable Requirements	A <i>sine qua non</i> for a successful project	Somewhere between a myth and a <i>canard</i> .
Predictable Budget and/or Schedule	A primary goal	Get real!
User Satisfaction	Should be achieved if the requirements are right and we correctly implement them	<u>The</u> primary goal
Progress Measurement	Project milestones	Delivered functionality scenarios

1/30/2007

Agile Process and eXtreme Programming

15

Complementary Religions

- There are different types of projects
 - with different goals and constraints
 - some require definition and control
 - others require investigation and iteration
- They are not mutually exclusive
 - agile processes can benefit from best practices
 - XP practices can be used in waterfall projects
 - traditional models ignored crucial team factors

1/30/2007

Agile Process and eXtreme Programming

16

What We Take Away

- Agile Development is a “reformation”
 - pointing out the evils of over-process-ism
 - reminding us of our real goals
- Agile Development takes a broader view
 - addressing key principles and people issues
- Agile Development may be the other pole
 - small vs. large, U/I vs. infrastructure
- We don't have to choose between extremes
 - thesis, antithesis, synthesis

1/30/2007

Agile Process and eXtreme Programming

17

For the next lecture

- McConnel
 - introduction to collaborative construction
- Project Work Partitioning
 - simple basic guidelines
- two papers on Pair Programming
 - an examination of how and why it works
 - how to go wrong with Pair Programming
- Global Software Development
 - position papers on agility meets global s/w dev

1/30/2007

Agile Process and eXtreme Programming

18

Supplementary Slides

1/30/2007

Agile Process and eXtreme Programming

19

Q: How much process do I need?

A: Just enough to ensure your success

- How large and complex is the problem?
 - one person-hour, or 40,000 staff years
- How well understood is the problem?
 - do it every day, clear, imponderable
- How critical is quality?
 - throw-away, product, mission/life critical
- How critical are cost and schedule?
 - not a problem, +/-50%, deal-breakers

1/30/2007

Agile Process and eXtreme Programming

20

Q: How formal a process?

A: Enough to prevent misunderstandings?

- How many stake holders are there?
 - two, twenty, two-hundred, twenty thousand
- How good is their communication?
 - inseparable, can't be in the same room together
- Have they successfully done this before?
 - regularly, a few times, never
- How complex are the responsibilities?
 - obvious, I don't even know what they are

1/30/2007

Agile Process and eXtreme Programming

21

Kampe's Advice on Process

- there will never be a perfect process
 - life is change, every problem is different
 - one size will never fit all
- but specified process is still important
 - check lists and clear expectations are vital
 - too much is bad, too little can be even worse
- specify "what", not "how"
 - enumerate the things that need to be done
 - don't try to tell people how to do their jobs

1/30/2007

Agile Process and eXtreme Programming

22

Kampe's Advice on Process

- only mandate things that are vital
 - like approval check-lists and authority
 - standard forms are good if they make sense
 - most things can be recommended practices
- always do cost-benefit analysis
 - process is expensive, try to keep it lean
 - watch out for "social engineering"
- prototype your process changes
 - prove them before instituting them

1/30/2007

Agile Process and eXtreme Programming

23