

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 may greatly burden small projects ●
 - it makes assumptions that may not be true
- it is a lowest-common-denominator solution
 - it can improve the work of weak teams ●
 - it can greatly limit strong contributors ●

11/23/2012

Agile Process

8

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

11/23/2012

Agile Process

9

People v.s. Process

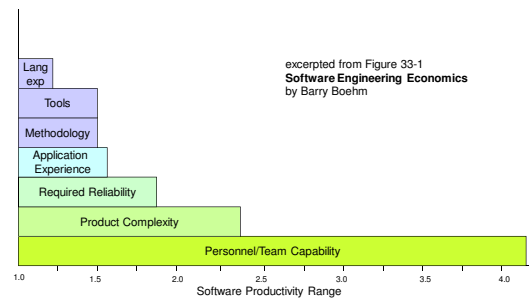
- Focus on creating good teams
 - ensure they have the right skill/perspective mix
 - ensure all the stake-holders are involved
 - ensure they are dedicated to the project
 - ensure they will work well **as a team**
- Then, let them do their job
 - don't tell them how to organize themselves
 - don't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

11/23/2012

Agile Process

10

The High Order Bits



11/23/2012

Agile Process

11

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
 - good for simple or poorly understood projects

11/23/2012

Agile Process

12

SCRUM

- good example of agile s/w development
 - mature, widely used
 - well documented roles and processes
- etymology
 - Rugby: all the forwards, moving together
 - same root at the American “scrimmage”

11/23/2012

Agile Process

13

Key Elements of SCRUM

- role of the product owner
 - manages and prioritizes the task backlog
 - provides continuous feedback to developers
 - decides whether or not s/w is acceptable
- short, fixed cadence sprints
 - a small number of well understood tasks
 - team commits to complete all of this work
 - ends w/working s/w delivered to product owner

11/23/2012

Agile Process

14

SCRUM Product Backlog

- team must take tasks from backlog in order
 - discuss the meaning and design of each item
 - estimate amount of work involved (in points)
 - team decides how much work they can handle
- tasks near the top must be “ready”
 - clearly understood and defined
 - broken into sprint-sized pieces
 - no blocking issues or dependencies
- deeper tasks are still “works in progress”

11/23/2012

Agile Process

15

SCRUM Work Processes

- daily “stand-up” meeting
 - very brief planning meeting for entire team:
 1. what accomplished yesterday
 2. what I plan on accomplishing today
 3. any obstacles I am currently facing
- continuous communication and assistance
 - technical problems: solved by the team
 - requirements issues: taken to product owner
 - other issues: addressed by SCRUM Master

11/23/2012

Agile Process

16

SCRUM Points and Velocity

- task difficulty is estimated in points
 - one point might be “four best-case hours”
 - this is a difficulty, not a completion date
 - points are only awarded for accepted s/w
- sprint velocity
 - measured in (accepted) points per sprint
 - improves as team becomes more effective
 - used to estimate how much work to accept
 - used to estimate time to completion of backlog

11/23/2012

Agile Process

17

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 •

11/23/2012

Agile Process

18

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	The primary goal
Progress Measurement	Process related - project milestones	Customer related - delivered stories

11/23/2012

Agile Process

19

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
 - getting it right from the start isn't always bad
 - agile processes can improve waterfall projects
 - traditional models ignored crucial team factors
 - traditional models assume perfect requirements

11/23/2012

Agile Process

20

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/large, user/infrastructure, adapt/plan
- We don't have to choose between extremes
 - Hegel's Dialectic: thesis, antithesis, synthesis

11/23/2012

Agile Process

21

For Next Lecture

McConnell 21.1-2 - collaborative development
McConnell 28.1, 28.5 – good practices
McConnell 33 – personal character
Extreme Programming:

- project activities, rules
- values, collective ownership
- Williams: Pair Programming
- Rosenberg: Problems w/Pair Programming

CACM: Global Software Development

11/23/2012

Agile Process

22

Supplementary Slides

11/23/2012

Agile Process

23

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

11/23/2012

Agile Process

24

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

11/23/2012

Agile Process

25

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

11/23/2012

Agile Process

26

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

11/23/2012

Agile Process

27