Agile Process

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

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

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 Eisenhauer

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

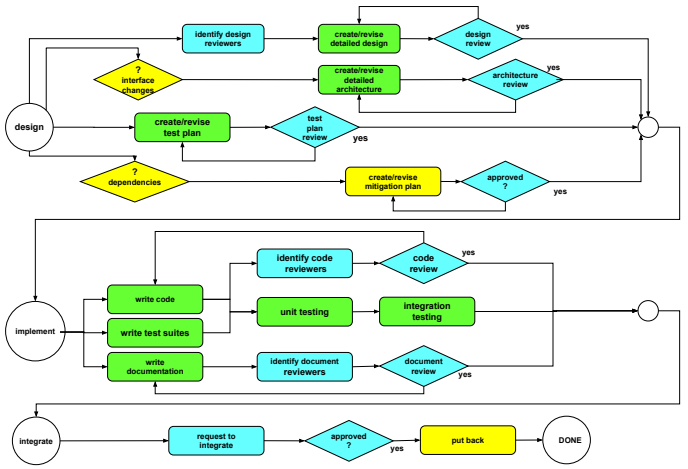
Fallacy of the Waterfall Model

- 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
  - because each incremental release is a waterfall

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

a formal process



Formal Process - the bad news

- it *can* place form over substance
  - people are goaled on process deliverables
  - real goals are customer satisfaction and ROI
- 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

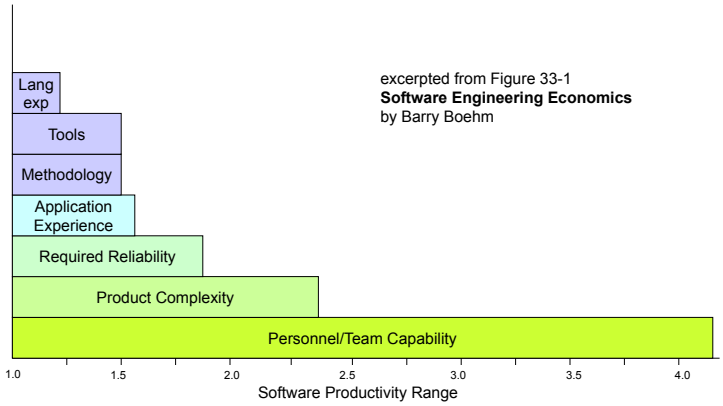
The Agile Process Rebellion

- 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 <u>task definition</u> is myopic!
  - people, not processes, solve problems

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, <u>let them do their job</u>
  - 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

The High Order Bits



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

Agile Puts Principles First

- Principles
  broad axioms of goodness and good behavior
- Methodology
  tools/techniques for solving problems
- Process
  a sequence of steps to be followed

The steps to be followed should be dictated by the applicable principles & methodology.

SCRUM

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

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

SCRUM Product Backlog

- team must take tasks from backlog <u>in order</u>
  - 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 <u>must be “ready”</u>
  - clearly understood and defined
  - broken into sprint-sized pieces
  - no blocking issues or dependencies
- deeper tasks are still “works in progress”

SCRUM Work Processes

- daily “stand-up” meeting

  <u>very brief</u> planning meeting for entire team:
  1. what I 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

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

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

Comparative Religion - differences

	Planned approaches	Agile approaches
Stable Requirements	A *sine qua non* for a successful project	Somewhere between a myth and a *canard*.
Predictable Budget and/or Schedule	A primary goal	Get real! We try to have a predictable velocity.
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

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
    - short sprints are safer and more predictable

What We Take Away

- Agile Development is a "reformation"
  - pointing out the evils of over-process-ism
  - reminding us of our real goals and strengths
- 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

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