Agile Program Development

CSCI 479: Computer Science Design Project

Fall 2013

Xiannong Meng

What is *Agile Development?*

Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams

http://en.wikipedia.org/wiki/Agile_software_development

Agile Manifesto

- Four core values of Agile
  - Individuals and their interactions
    - Over processes and tools
  - Delivering working software
    - Over comprehensive documentation
  - Customer collaboration
    - Over contract negotiation
  - Responding to change
    - Over following a plan

http://agilemanifesto.org/

12 Principles of Agile Development 1 of 4

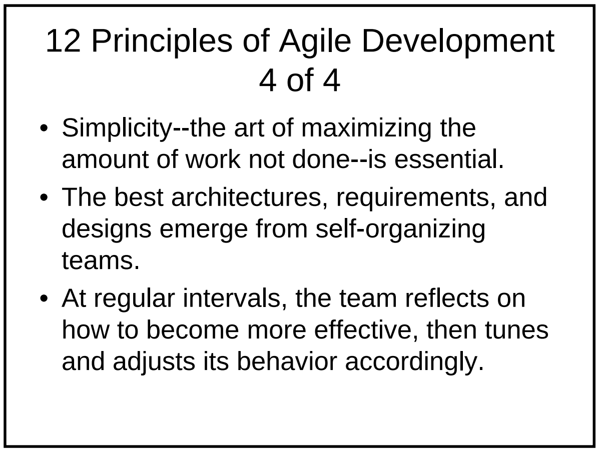
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

12 Principles of Agile Development 2 of 4

- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

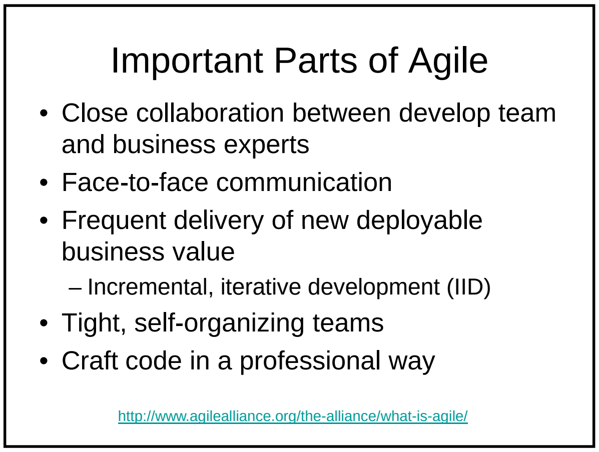
12 Principles of Agile Development 3 of 4

- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.



12 Principles of Agile Development 4 of 4

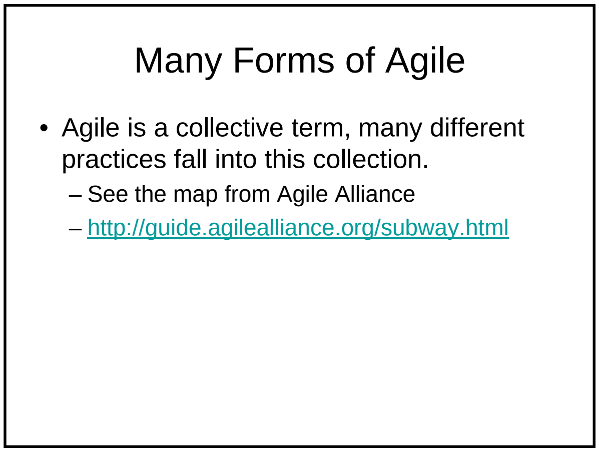
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



Important Parts of Agile

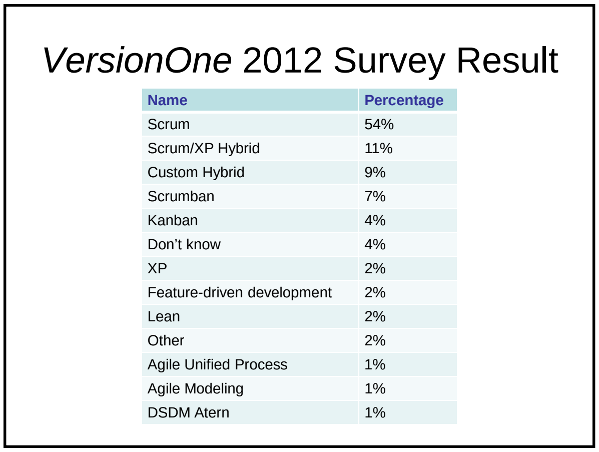
- Close collaboration between develop team and business experts
- Face-to-face communication
- Frequent delivery of new deployable business value
  - – Incremental, iterative development (IID)
- Tight, self-organizing teams
- Craft code in a professional way

http://www.agilealliance.org/the-alliance/what-is-agile/



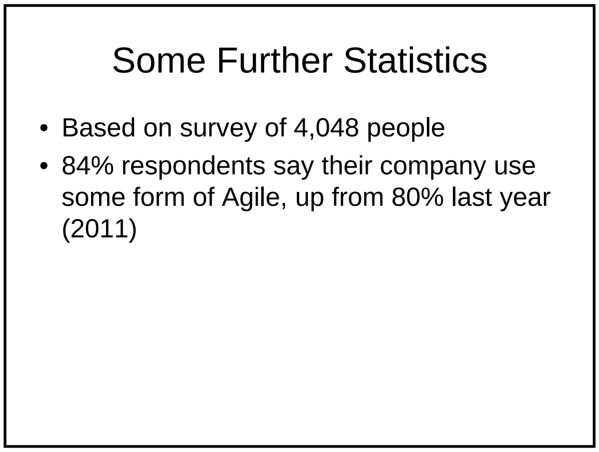
Many Forms of Agile

- Agile is a collective term, many different practices fall into this collection.
  - – See the map from Agile Alliance
  - – http://guide.agilealliance.org/subway.html



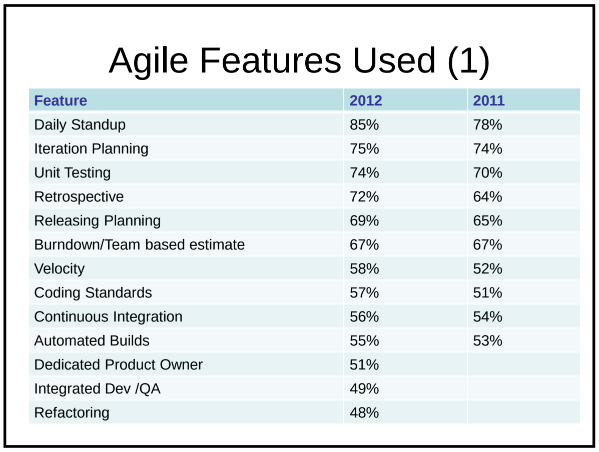
VersionOne 2012 Survey Result

Name	Percentage
Scrum	54%
Scrum/XP Hybrid	11%
Custom Hybrid	9%
Scrumban	7%
Kanban	4%
Don't know	4%
XP	2%
Feature-driven development	2%
Lean	2%
Other	2%
Agile Unified Process	1%
Agile Modeling	1%
DSDM Atern	1%



Some Further Statistics

- Based on survey of 4,048 people
- 84% respondents say their company use some form of Agile, up from 80% last year (2011)



Agile Features Used (1)

Feature	2012	2011
Daily Standup	85%	78%
Iteration Planning	75%	74%
Unit Testing	74%	70%
Retrospective	72%	64%
Releasing Planning	69%	65%
Burndown/Team based estimate	67%	67%
Velocity	58%	52%
Coding Standards	57%	51%
Continuous Integration	56%	54%
Automated Builds	55%	53%
Dedicated Product Owner	51%	
Integrated Dev /QA	49%	
Refactoring	48%	

Agile Features Used (2)

Feature	2012	2011
Open Work Area	43%	38%
Test Driven Development (TDD)	40%	38%
Digital Taskboard	39%	32%
Story Mapping	38%	35%
Kanban	32%	24%
Collective Code Ownership	32%	28%
Pair Programming	30%	
Automated Acceptance Testing	27%	25%
Analog Taskboard	24%	22%
Continuous Deployment	23%	
Agile Games	17%	15%
Cycle Time	13%	12%
Behavior Driven Development (BDD)	10%	9%

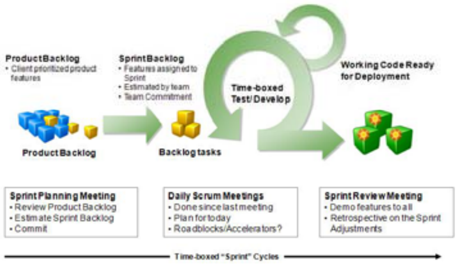
What Do These Terms Mean?

- Check out the guidelines from
  - http://guide.agilealliance.org

A Brief Overview of Scrum

- *Scrum is an iterative and incremental agile software development framework for managing software projects and product or application development.*
  - http://en.wikipedia.org/wiki/Scrum_(software_development)

Scrum Process



http://sandervanpeer.wordpress.com/2012/01/05/why-should-we-choose-scrum-for-our-projects/

Sprint

- A *sprint* is the basic unit of development in Scrum.
  - The sprint is a “timeboxed” effort, i.e., a specific duration
  - The duration usually is a week to a month
  - Each sprint is proceeded by a planning meeting (e.g., daily standing), followed by a review meeting

Sprint Meetings

- *Planning meeting*: where the tasks for the sprint are identified and an estimated commitment for the sprint goal is made
- *Review meeting* : where the progress is reviewed and goals for the next sprint are identified.
- *Retrospective meeting*: where team members reflect their experience and draw lessons from it

Daily Scrum Format

- Each day during the sprint, a project team communication meeting occurs. This is called a *daily scrum*, or *the daily standup*. This meeting has specific guidelines:
  - All members of the development team come prepared with the updates for the meeting.
  - The meeting starts precisely on time even if some development team members are missing.
  - The meeting should happen at the same location and same time every day.
  - The meeting length is set (timeboxed) to 15 minutes.
  - All are welcome, but normally only the core roles speak.

Daily Scrum Content

- During the meeting, each team member answers three questions:
  - What have you done since yesterday?
  - What are you planning to do today?
  - Any impediments/stumbling blocks? Any impediment/stumbling block identified in this meeting is documented by the Scrum Master and worked towards resolution outside of this meeting. No detailed discussions shall happen in this meeting.

End of Cycle

- At the end of a sprint, two meetings are held
  - Sprint review meeting, four things to do
    - Review the work that was completed and the planned work that was not completed
    - Present the completed work to the stakeholders (a.k.a. "the demo")
    - Incomplete work cannot be demonstrated
    - Four-hour time limit

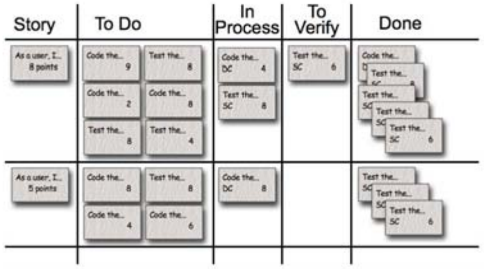
End of Cycle

- At the Sprint Retrospective:
  - All team members reflect on the past sprint
  - Make continuous process improvements
  - Two main questions are asked in the sprint retrospective: What went well during the sprint? What could be improved in the next sprint?
  - Three-hour time limit
  - This meeting is facilitated by the Scrum Master

Scrum Artifacts

- Product Backlog
  - an ordered list of "requirements" that is maintained for a product.
- Sprint Backlog
  - the list of work the Development Team must address during the next sprint.
- Increments
  - the sum of all the Product Backlog Items completed during a sprint and all previous sprints
- Burn down chart
  - publicly displayed chart showing remaining work in the sprint backlog.

Scrum Burn Down Chart



http://www.mountaingoatsoftware.com/system/asset/file/29/MockedTaskBoard.jpg

What Are We Going To Do?

- Have team meetings with the customers every week including the instructor
- At each meeting discuss and record what’s accomplished and what’s to do for next week
- Have milestones (sprints) twice during the semester and one at the end of the semester

References

- VersionOne 7th Annual State of Agile Development Survey, 2012
  - http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf
- Agile Manifesto
  - http://agilemanifesto.org/
- Agile Principles
  - http://agilemanifesto.org/principles.html
- Scrum
  - http://en.wikipedia.org/wiki/Scrum_(software_development)