ADVANCED SOFTWARE DEVELOPMENT METHODOLOGIES

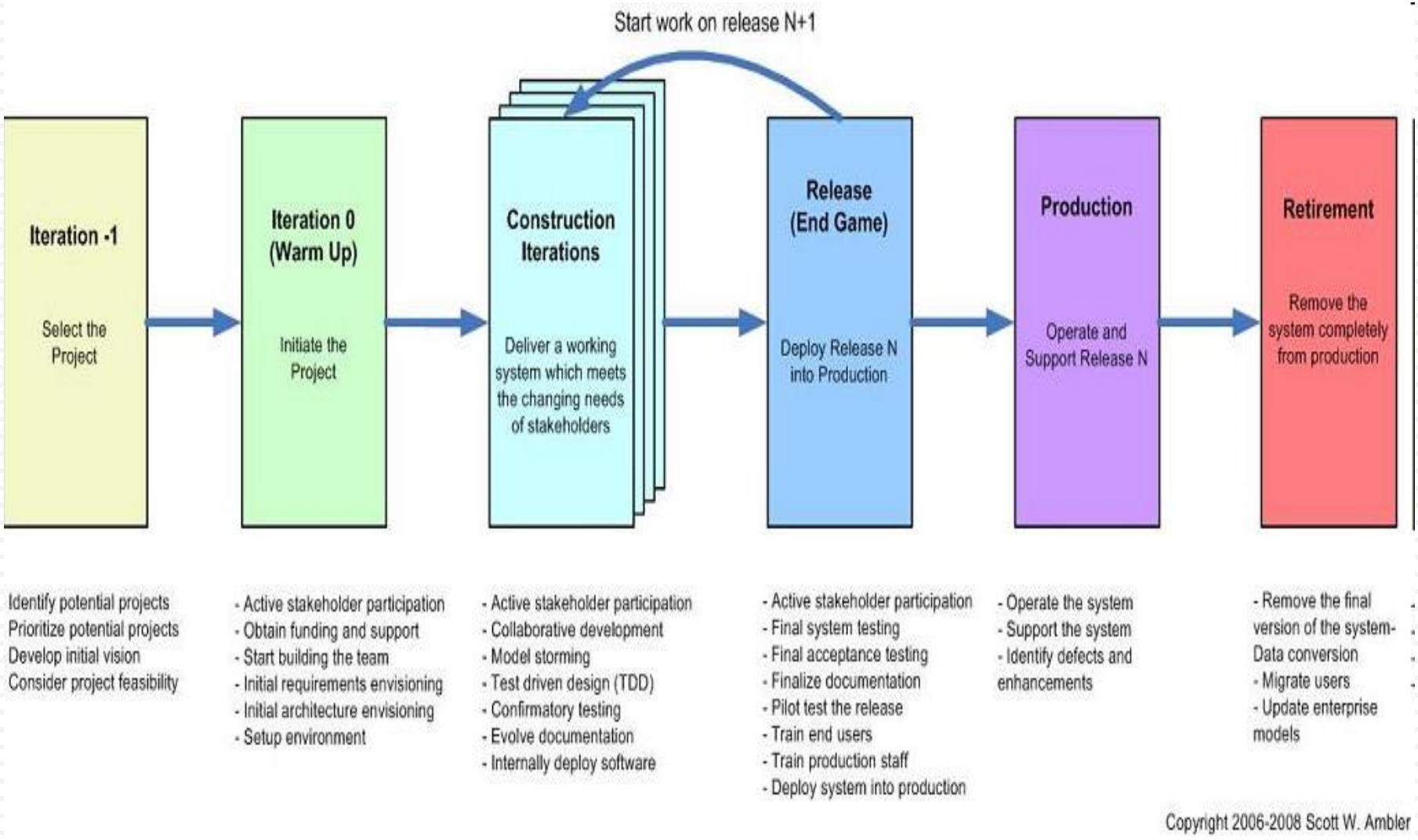
16-06-2019

Agile Software Development

Agile Development – What does it mean?

- Agile is a set of ***practices, values,*** and ***principles*** for software product development.
- In software product development, we think about "methodologies," "activities," "interactions," "results, work products or artifacts;" we think about "processes" that we use to organize the work:
  - documents
  - meetings and reviews
  - diagrams and models
  - coding and user documentation standards
- So will Agile Development define a new set of process activities? Not necessarily.

Cont..


Start work on release N+1
Iteration -1
Select the Project
Iteration 0 (Warm Up)
Initiate the Project
Construction Iterations
Deliver a working system which meets the changing needs of stakeholders
Release (End Game)
Deploy Release N into Production
Production
Operate and Support Release N
Retirement
Remove the system completely from production
Identify potential projects
Prioritize potential projects
Develop initial vision
Consider project feasibility
- Active stakeholder participation
- Obtain funding and support
- Start building the team
- Initial requirements envisioning
- Initial architecture envisioning
- Setup environment
- Active stakeholder participation
- Collaborative development
- Model storming
- Test driven design (TDD)
- Confirmatory testing
- Evolve documentation
- Internally deploy software
- Active stakeholder participation
- Final system testing
- Final acceptance testing
- Finalize documentation
- Pilot test the release
- Train end users
- Train production staff
- Deploy system into production
- Operate the system
- Support the system
- Identify defects and enhancements
- Remove the final version of the system-Data conversion
- Migrate users
- Update enterprise models
Copyright 2006-2008 Scott W. Ambler

What is Agile? (Agile vs. Sequential)

- Many of us are familiar with the Waterfall Model – it is a "framework" for the software development process
  - Waterfall Model talks about "development activities through time"
  - Waterfall Model talks about "teams of people"

Development activities	Teams
Divide the work into stages	A separate team of specialists for each stage
At each stage, the work is passed from one team to another	Some coordination is required for the handoff from team to team – using "documents"
At the end of all of the stages, you have a software product ready to ship	As each team finishes, they are assigned to a new product

Cont..

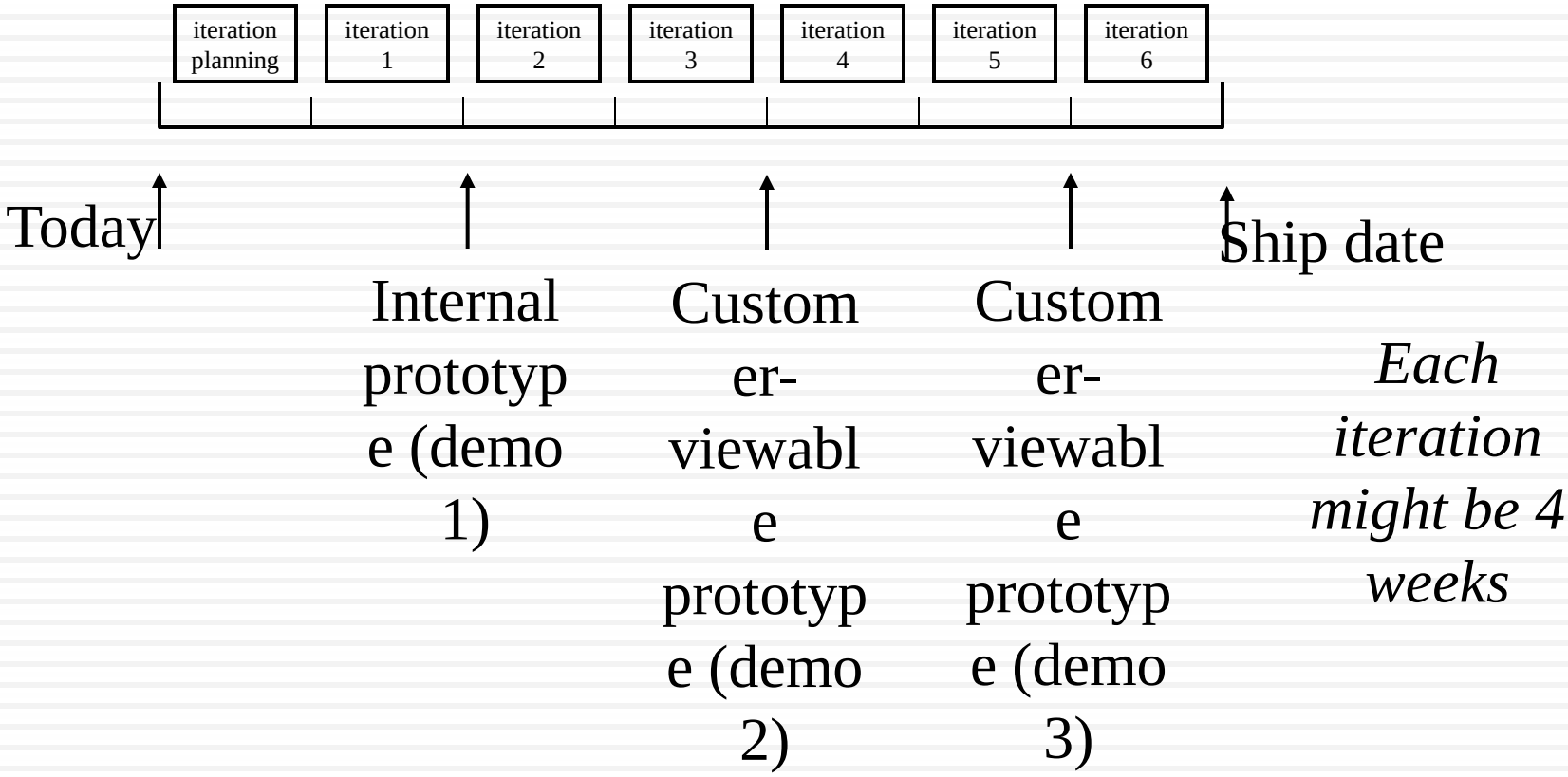
- The core ideas in Agile Development:
  - Adaptive
  - Iterative/incremental
  - People-oriented

- **Adaptive** means that the teams and the process should be flexible in the presence of “rapid-fire change”.

- **Iterative and incremental** means that Agile Development produces working products in stages – a growing set of “completed and working software”.

- **People-oriented** means the team organization and processes will support good people, who are the most important ingredient to project success.

Iterative development

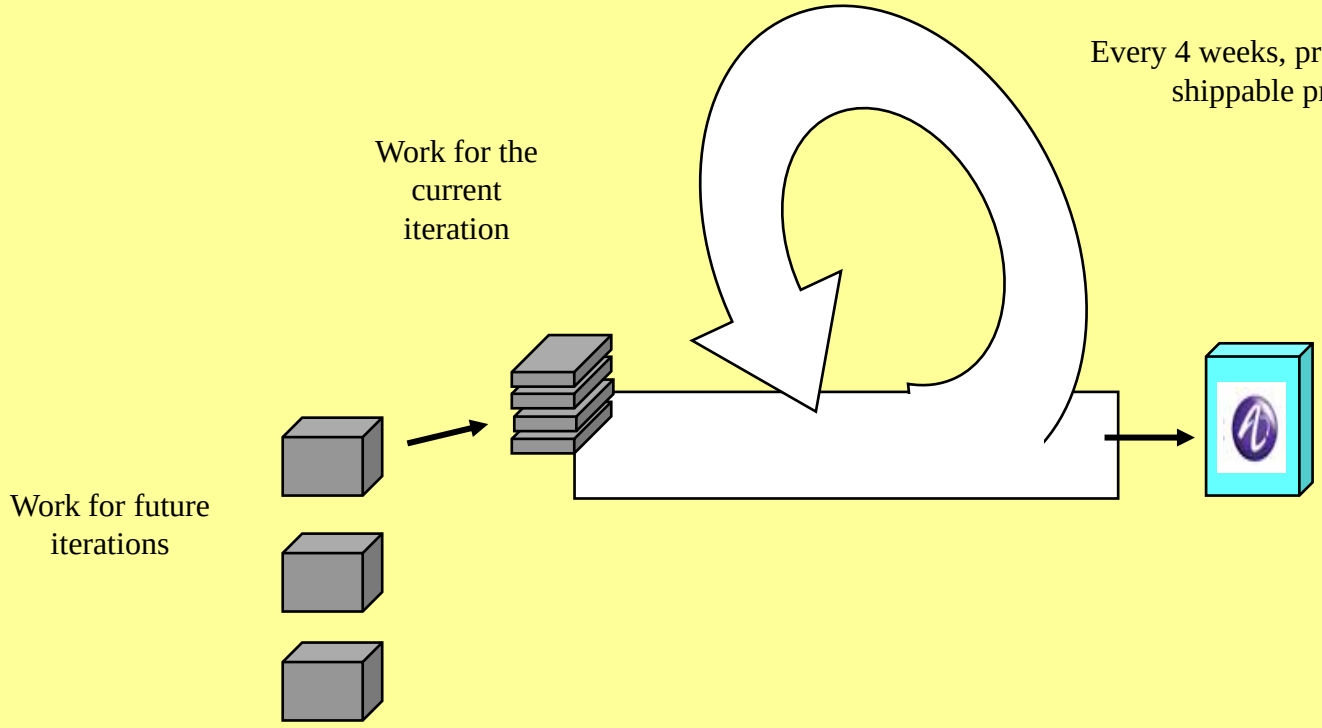


- One way to organize agile development is using short iterations:

Main characteristics of Agile Development

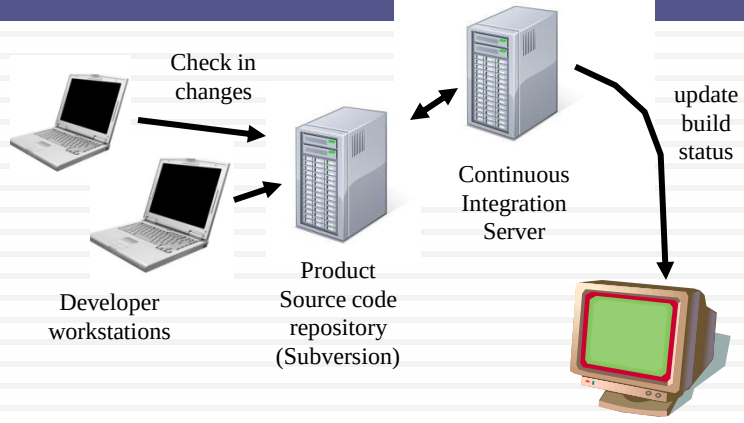
- Agile Development as a “software development framework” says:
  - keep things small
  - deliver partially-completed software frequently
  - talk to the customer often
  - write more code than documentation
  - everyone on the team learns together

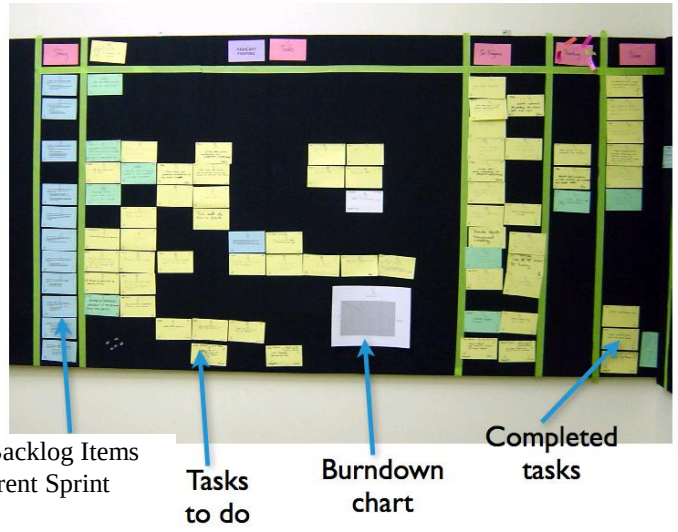
Cont..


Every 4 weeks, produce a new shippable product
Work for the current iteration
Work for future iterations

Agile Practices

- There are many Agile practices:
  - short time boxed iterations
  - continuous integration
  - daily unit testing
  - regular retrospectives
  - direct communication between developers and the customer or a customer surrogate
  - a single list of features and tasks
  - short-term estimation of development tasks





Cont..

- information radiators
- refactoring

- Will you use every Agile practice? Maybe not…. they are not all required.
- What *is* required? Agile values…

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions over processes and tools**
- **Working software over comprehensive documentation**
- **Customer collaboration over contract negotiation**
- **Responding to change over following a plan**

That is, while there is value in the items on the right, we value the items on the left more.

Agile Alliance: http://www.agilealliance.org

Agile principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	9. Continuous attention to technical excellence and good design enhances agility.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	10. Simplicity--the art of maximizing the amount of work not done--is essential.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	7. Working software is the primary measure of progress.	11. The best architectures, requirements, and designs emerge from self-organizing teams.
4. Business people and developers must work together daily throughout the project.	8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile – questions and challenges?

- **Documentation –** it is still important in an Agile project.
  - If it is the only kind of communication in your project, it isn't good
  - Real working code is more valuable than documents – less ambiguous
  - Documents – easy to leave something out, easy to misinterpret
- **Development plans –** also important in an Agile project
  - the format of an Agile development schedule is a bit different from a conventional project plan.

Agile – questions and challenges?

- Development plan includes "iterations"
- Each iteration gives the team has a chance to incorporate what they learn, rather than just following a non-adaptive plan

- **Contracts** – we expect to have contracts, but we need to talk with the customers as well.
  - Customer collaboration is one way to reduce development costs
  - Do you want to deliver "everything" the customer asked for in the original contract? No – if the customer no longer needs it, the extra code will increase maintenance costs
  - Always ask: Who needs this feature and how does it contribute to the value of the product?

Why is Agile Development important?

- The world is a lot different today. A large feature set might only increase costs for the customer.
  - There is a constant introduction of new technology
  - New players enter the market,
  - New requirements are added
  - "Small is Beautiful"
  - If we are listening to the customer, we will reduce our chances of being "blindsided" by a smaller, more flexible competitor
  - Anything that helps reduce maintenance costs will contribute to the bottom line

How hard is it to be Agile?

- "Don't do Agile, be Agile"
  - Just doing "development in iterations" isn't enough
- Agile Development is about:
  - Keeping the process lightweight
  - Making real progress in each iteration
  - Communicating – face-to-face when possible
  - Actively gathering customer input – early and often
  - Being willing to make minor changes to your process