Testing Agile Based Software

Prof. Durga Prasad Mohapatra
Dept. of CSE, NIT Rourkela
India

Agile software Development: Motivation

- There are many SDLC models available, such as waterfall model, prototyping model, iterative model, incremental model, spiral model and many more.
- These traditional models follow the *static* working strategy in which everything is fixed, for example cost of project, requirements or scope, schedule etc. Further, the duration is in years.
- Besides, there are several other drawbacks of each model.

Agile software Development

- In order to overcome these drawbacks, several major software vendors have been adopting a form of “intelligent” development in one or more phases of their software development processes.
- Agile for example, is a well-known example of a lifecycle used to build intelligent and analytical systems.
- Agile development does not promote best practices; rather, it is about adaptive planning and evolutionary development.

What is Agile Software Development?

- Agile: Easily moved, light, nimble, and active software processes
- How agility achieved?
  - Fitting the process to the project
  - Avoiding things that waste time

Agile software Development

- ASD is based on light weight methodologies (less documentation and less planning) having dynamic nature, which is its major strength.
- This model provides an environment to accommodate frequent changes as per market standards and customer needs.
- ASD is an iterative and incremental development method and it is customer centred.

Agile software Development

- The agile process consists of multiple sprints (iterations or runs or development cycles); in each sprint a specific software feature is developed, tested, refined and documented.
- However, because agile development depends on the context of the project, testing is performed differently in every sprint.

Agile Model

- To overcome the shortcomings of the waterfall model of development.
  - Proposed in mid-1990s
- The agile model was primarily designed:
  - To help projects to adapt to change requests
- In the agile model:
  - The requirements are decomposed into many small incremental parts that can be developed over one to four weeks each.

History: The Agile Manifesto

- On February 11-13, 2001, at The Lodge at Snowbird ski resort in the Wasatch mountains of Utah, seventeen people met to talk, ski, relax, and try to find common ground—and of course, to eat.
- What emerged was the Agile 'Software Development' Manifesto.

History: The Agile Manifesto cont…

- Representatives from Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, and others sympathetic to the need for an alternative to documentation driven, heavyweight software development processes convened.

History: The Agile Manifesto cont…

- Now, a bigger gathering of organizational anarchists would be hard to find, so what emerged from this meeting was symbolic—a Manifesto for Agile Software Development—signed by all participants.

Manifesto for Agile Software Development

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.

Principles behind the Agile Manifesto

- 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.

Principles behind the Agile Manifesto cont…

- 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.

Principles behind the Agile Manifesto cont…

- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- 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.

Principles behind the Agile Manifesto cont…

- Continuous attention to technical excellence and good design enhances agility.
- 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 behaviour accordingly

Ideology: Agile Manifesto

- Individuals and interactions *over*
  - process and tools

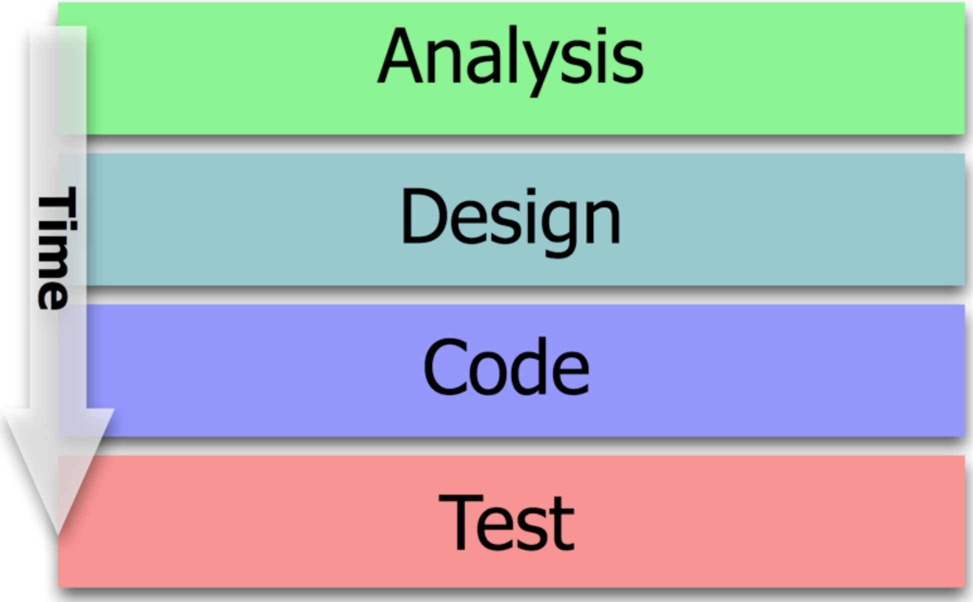
`http://www.agilemanifesto.org`

- Working Software *over*
  - comprehensive documentation
- Customer collaboration *over*
  - contract negotiation
- Responding to change *over*
  - following a plan

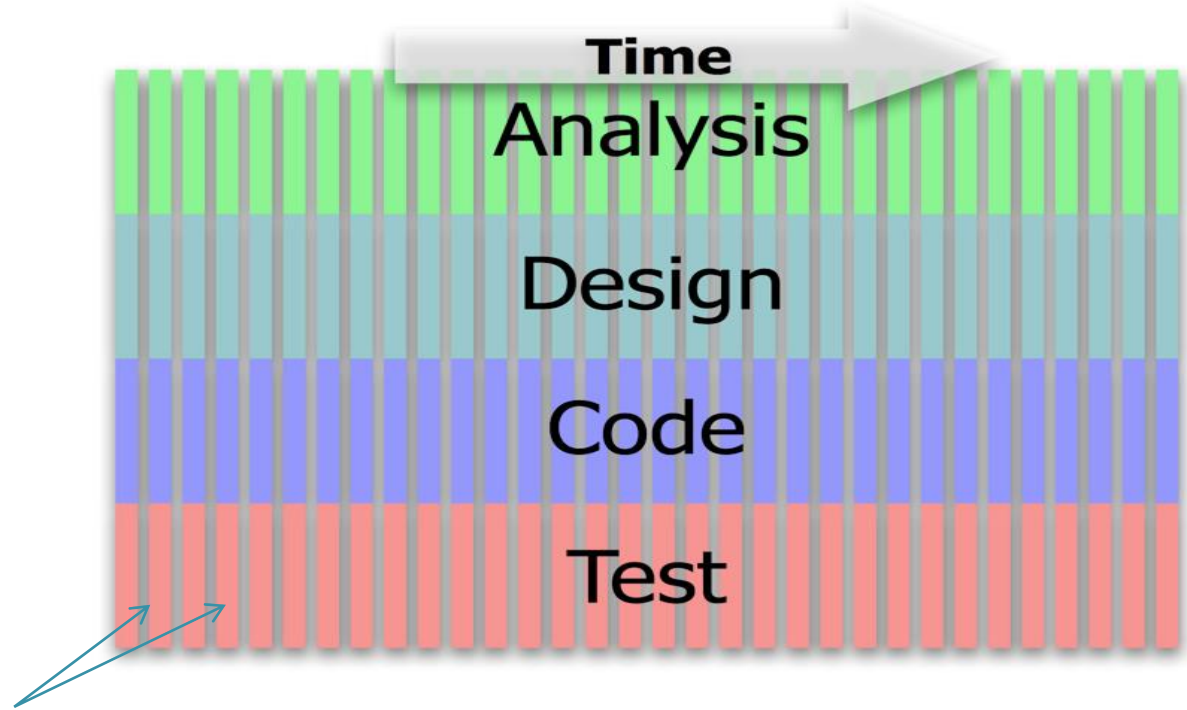
Agile Methodologies

- XP
- Scrum
- Unified process
- Crystal
- DSDM
- Lean

Traditional Software Development



Applying Lean Principles to Software Development … (a better way of doing the same)



End-to-End small slices of work

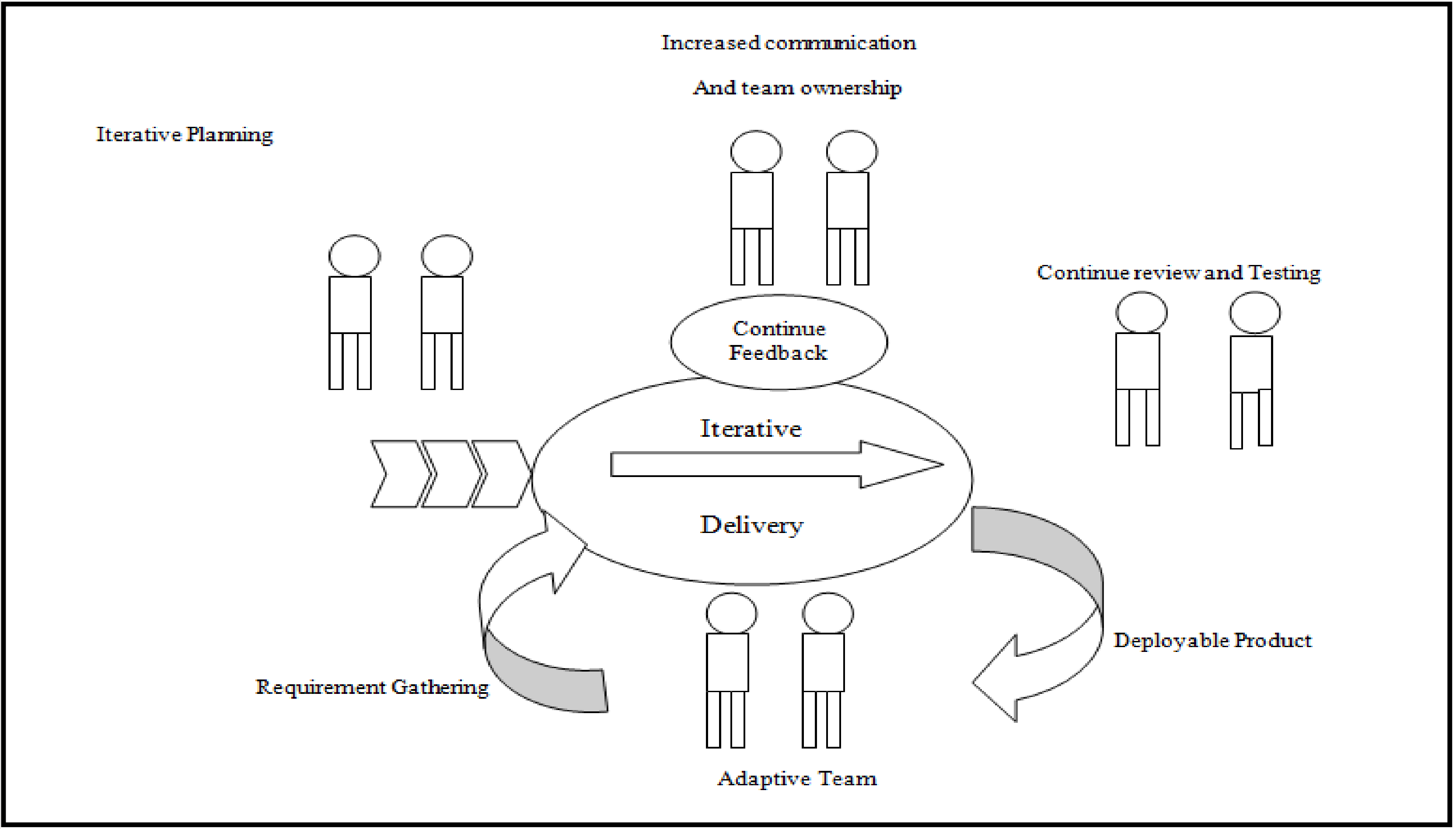
Agile Model

- The Agile model puts emphasis on the fact that whole Agile team should be a tightly integrated unit and is composed of
  - Developers
  - Quality assurance members
  - Testers
  - Project owner
  - Customer

Agile Model cont …

- The key feature of ASD is to have effective communication between all team members. For valuable communication and information exchange, daily meetings are held in ASD.
- Another important feature of agile process is iterative delivery. An iteration or delivery cycle in ASD ranges from 1 – 4 weeks.

Agile Model cont …



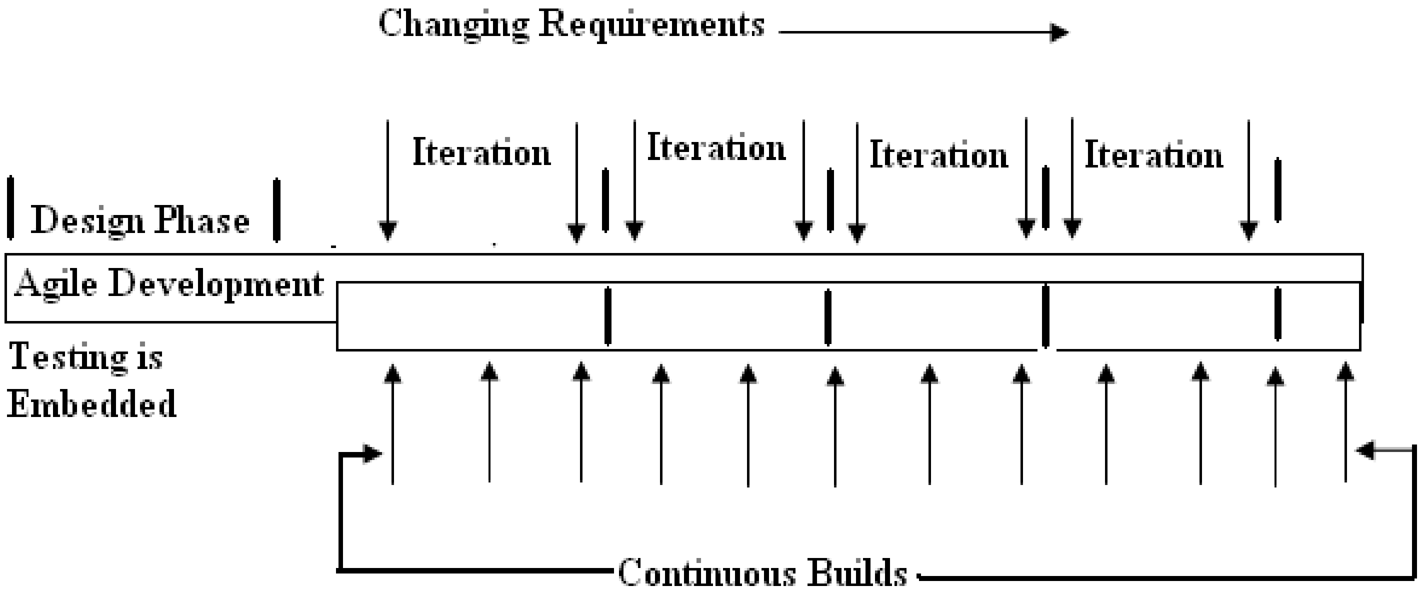
Agile Model: Principal Techniques

- User stories:
  - Simpler than use cases.
- Metaphors:
  - Based on user stories, developers propose a common vision of what is required.
- Spike:
  - Simple program to explore potential solutions.
- Refactor:
  - Restructure code without affecting behavior, improve efficiency, structure, etc.

Agile Model: Nitty Gritty

- At a time, only one increment is planned, developed, deployed at the customer site.
  - No long-term plans are made.
- An iteration may not add significant functionality,
  - But still a new release is invariably made at the end of each iteration
  - Delivered to the customer for regular use.

Agile Software Development Life Cycle



Stakeholders in Agile Life Cycle

- Agile SDLC includes specific activities performed by manager (M), developers (D), testers (T), marketing professional (MP) and customer (C).
- Table 1 lists the activities performed by different stakeholders
- At the time of production of the code or before producing the code, testing is applied by writing failed test cases, unlike the traditional approach of working.

Stakeholders in Agile Life Cycle

- Testing activity begins as soon as user stories (requirements) are finalized and prioritized, and
    - testers try to move business logic into lower levels in order to test with lower effort in the last stage.

Stakeholders in agile life cycle

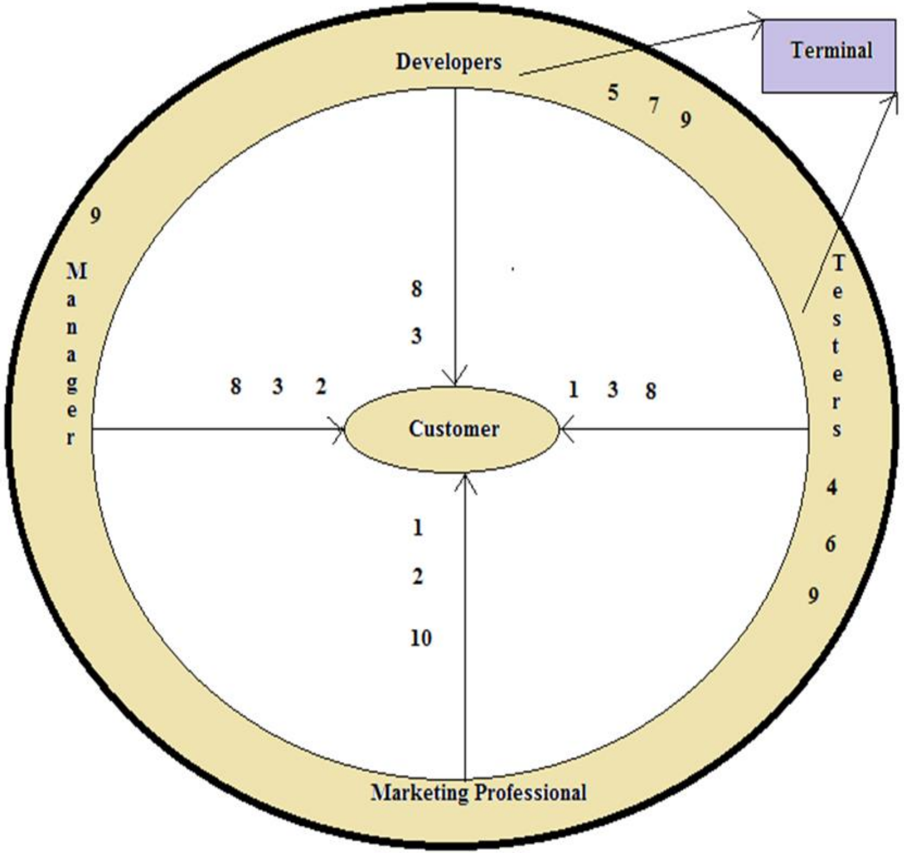


Fig 3: Stakeholders in agile life cycle

Table 1 : Actor Activity Chart - Agile Life Cycle

S. No.	Actor	Activity
1	C, MP, T	Requirements Gathering
2	M, MP, C	Effort estimation [28], cost, risk, capacity & resource Estimation
3	M, D, T, C	User stories Prioritization
4	T	Designing of Test Cases
5	D	Coding for the user story in the iteration
6	T	Feedback
7	D	Refactoring for the user story
8	C, M, D, T	Review meeting with Demonstration
9	D, T, M, MP	Lesson Learning phase or Retrospective session after the iteration
10	C, MP	Release

Agile software development life cycle

- In agile, a quality(Q) product is delivered by operational teams, and acceptance factor (A) is related to the rate at which customer accepts the delivered product.
- Effectiveness (E) of the team is related to two factors as shown in below equation

$$E = Q \times A$$

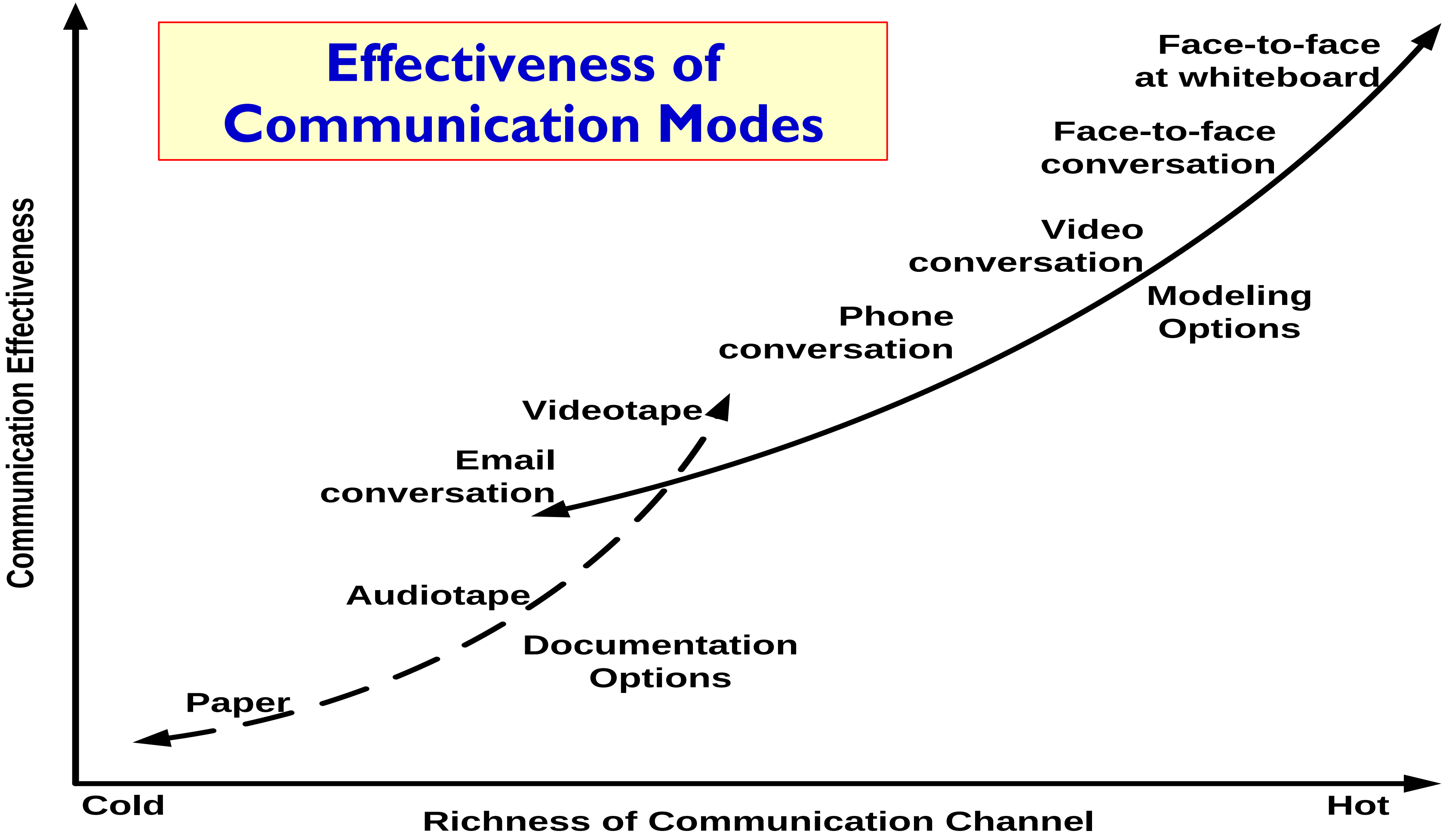
Agile software development life cycle cont…

- Out of these 2 factors, quality is more significant as acceptance is totally dependent on Q.
- Q is more when there is lesser number of backlogs and it is less when backlog items are more.
- Backlogs can be decreased when automation is the preferred approach over a manual way of testing.

Methodology

- Face-to-face communication favoured over written documents.
- To facilitate face-to-face communication,
  - Development team should share a single office space.
  - Team size is deliberately kept small (5-9 people).
  - This makes the agile model most suited to the development of small projects.

Effectiveness of Communication Modes



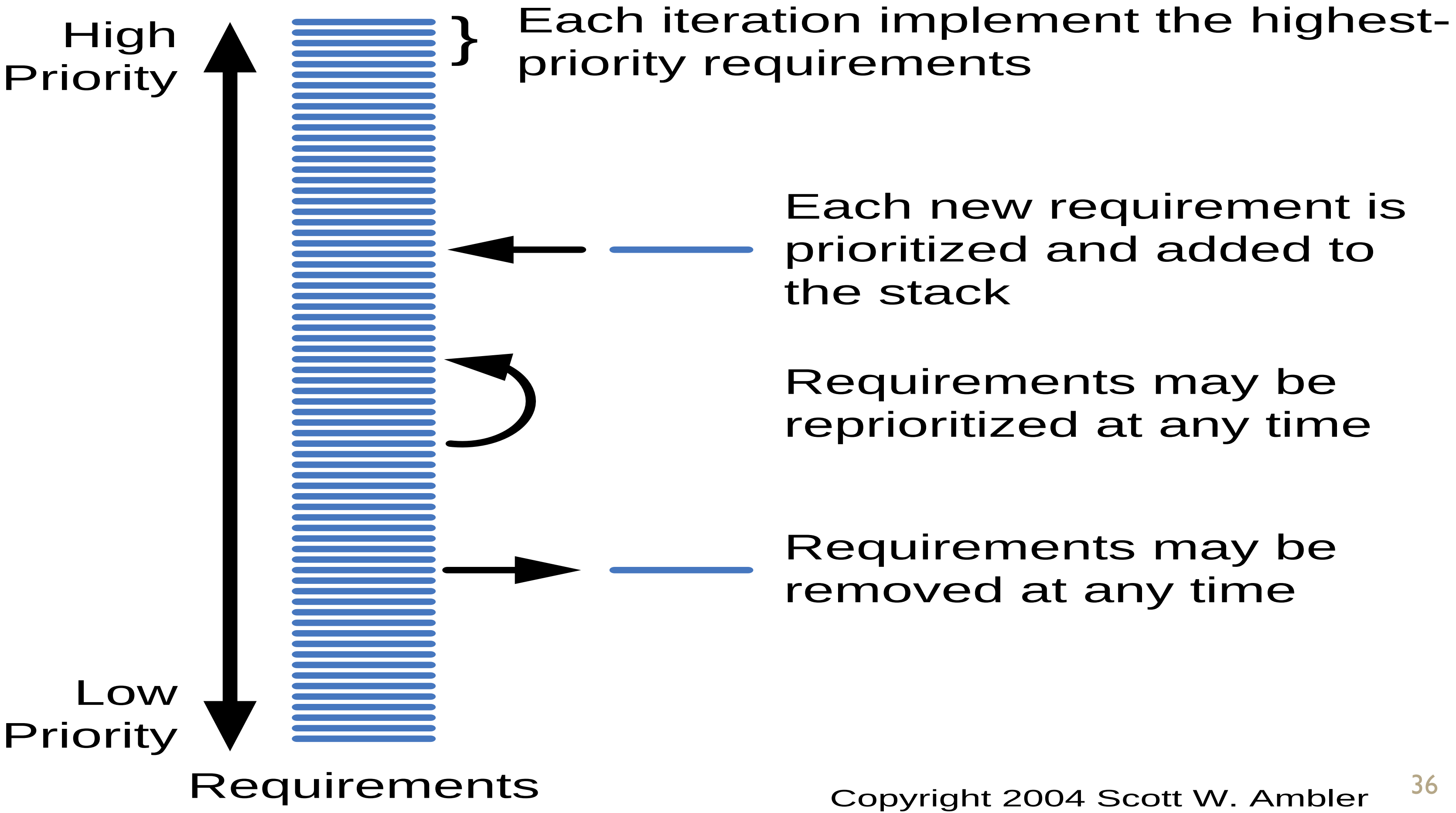
Agile Model: Principles

- The primary measure of progress:
  - Incremental release of working software
- Important principles behind agile model:
  - Frequent delivery of versions --- once every few weeks.
  - Requirements change requests are easily accommodated.
  - Close cooperation between customers and developers.
  - Face-to-face communication among team members.

Agile Documentation

- Travel light:
  - You need far less documentation than you think.
- Agile documents:
  - Are concise
  - Describe information that is less likely to change
  - Describe "good things to know"
  - Are sufficiently accurate, consistent, and detailed
- Valid reasons to document:
  - Project stakeholders require it
  - To define a contract model
  - To support communication with an external group
  - To think something through

Agile Software Requirements Management



High Priority

Low Priority

Requirements

Each iteration implement the highest-priority requirements

Each new requirement is prioritized and added to the stack

Requirements may be reprioritized at any time

Requirements may be removed at any time

Adoption Detractors

- Sketchy definitions, make it possible to have
  - Inconsistent and diverse definitions
- High quality people skills required
- Short iterations inhibit long-term perspective
- Higher risks due to feature creep:
  - Harder to manage feature creep and customer expectations
  - Difficult to quantify cost, time, quality.

Agile Model Shortcomings

- Derives agility through developing tacit knowledge within the team, rather than any formal document:
    - Can be misinterpreted…
    - External review difficult to get…
    - When project is complete, and team disperses, maintenance becomes difficult…

Agile Model vs Waterfall Model

- Steps of Waterfall model are a planned sequence:
  - Requirements-capture, analysis, design, coding, and testing .
- Progress is measured in terms of delivered artefacts:
  - Requirement specifications, design documents, test plans, code reviews, etc.
- In contrast agile model sequences:
  - Delivery of working versions of a product in several increments.

Agile Model vs Waterfall Model cont …

- As regards to similarity:
  - We can say that Agile teams use the waterfall model on a small scale.

Thank You