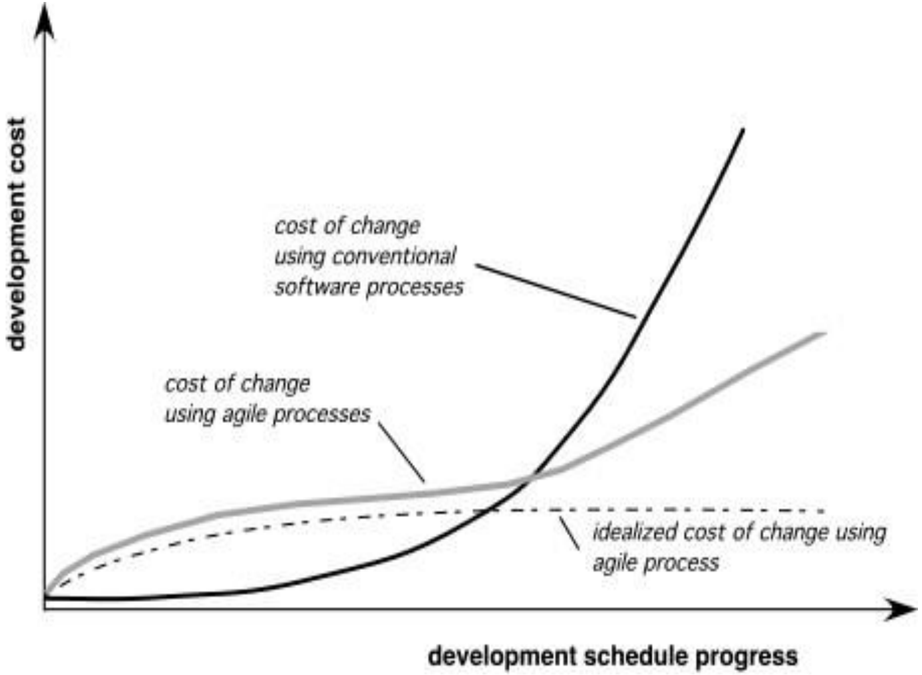
SOFTWARE ENGINEERING

Software Process

Agile Software Development

- Need for agility
  - Market conditions change rapidly, end-user needs evolve, and new competitive threats emerge without warning
  - won’t be able to define requirements fully before the project begins
  - It would be a fluid business environment.
  - Fluidity implies change, and we know change is expensive, if it is uncontrolled or poorly managed

Agility



- □ Effective (rapid and adaptive) response to change.
- □ Effective communication among all stakeholders.
- □ Drawing the customer onto the team.
- □ Organizing a team so that it is in control of the work performed.
- □ Rapid, incremental delivery of software.

Agile Software Development

- **Agile software development emphasizes on:**
  - Good communication between the client and developers
  - Rapid delivery of software
  - Change of Demand
- **It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach**
- **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

Agility Principles

- Customer satisfaction is achieved by providing value through software that is delivered to the customer as rapidly as possible.
- Developer recognize that requirements will change and welcome changes.
- Deliver software increments frequently (weeks not months) to stakeholders to ensure feedback on their deliveries is meaningful.
- Agile team populated by motivated individuals using face-to-face communication to convey information.
- Team process encourages technical excellence, good design, simplicity, and avoids unnecessary work.

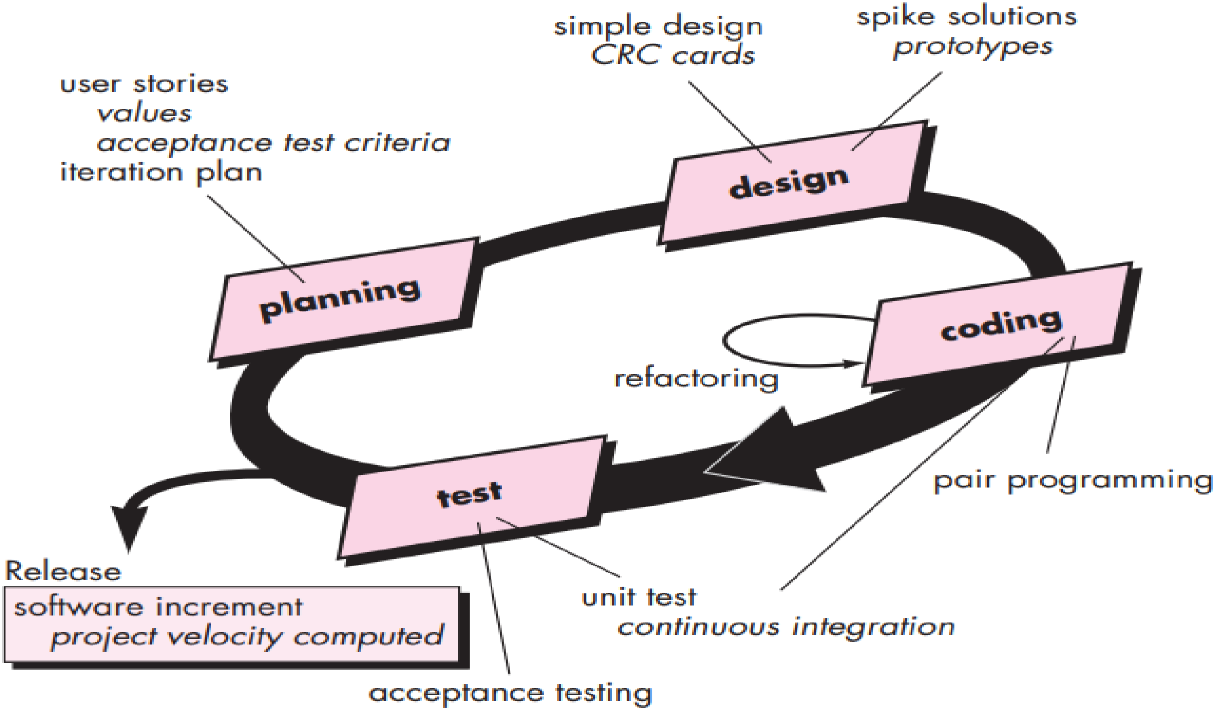
Agility Principles

- Working software that meets customer needs is the primary goal.
- Pace and direction of the team's work must be "sustainable," enabling them to work effectively for long periods of time.
- An agile team is a "self-organizing team"—one that can be trusted develop well-structured architectures that lead to solid designs and customer satisfaction.
- Part of the team culture is to consider its work introspectively with the intent of improving how to become more effective - its primary goal (customer satisfaction).

Types of Agile Methodologies

- Adaptive Software Development (ASD) - {J. Highsmith and S. Bayer, 1992}
- Crystal Methods (Crystal Clear) - {Alistair Cockburn, 1992}
- Extreme Programming (XP) - {Kent Beck, 1999}
- Feature Driven Development (FDD) - {Jeff De Luca, 1997}
- Lean Software Development - {Mary Poppendieck and Tom Poppendieck, 2003}
- Agile Unified Process (AUP) - {Scott Ambler, 2000}
- Scrum - {Ken Schwaber, Jeff Sutherland, 1991}
- Disciplined Agile Delivery - {Scott Ambler and Mark Lines, 2012}
- Dynamic Systems Development Method (DSDM) - {DSDM Consortium, 1994}

Extreme Programming



Extreme Programming

- **Extreme Programming** (XP) is based on simple philosophy: "If something is known to be beneficial, why not put it to constant use?"
- The software development team determines the various features (stories) the client would like the product to support.
  - A user story is a simplistic statement of customer about a functionality he needs.
- Based on the features (stories) the client wants:
  - The development team estimates the duration and cost of each feature
  - The client selects the features for next build using cost-benefit analysis
  - The proposed build is broken down into smaller pieces termed tasks
  - The programmer draws up test cases for a task → Test-driven development (TDD)

Extreme Programming

- Pair Programming: The programmer works together with a partner on one screen to implement the task and ensure that all the test cases work correctly.
- The TDD test cases used for the task are retained and utilized in all further integration testing.
- A spike solution is a very simple program to explore potential solutions. Build the spike to only addresses the problem under examination and ignore all other concerns.
- Framework Activities in XP Process Model:
  - Planning
  - Design
  - Coding
  - Testing

Extreme Programming

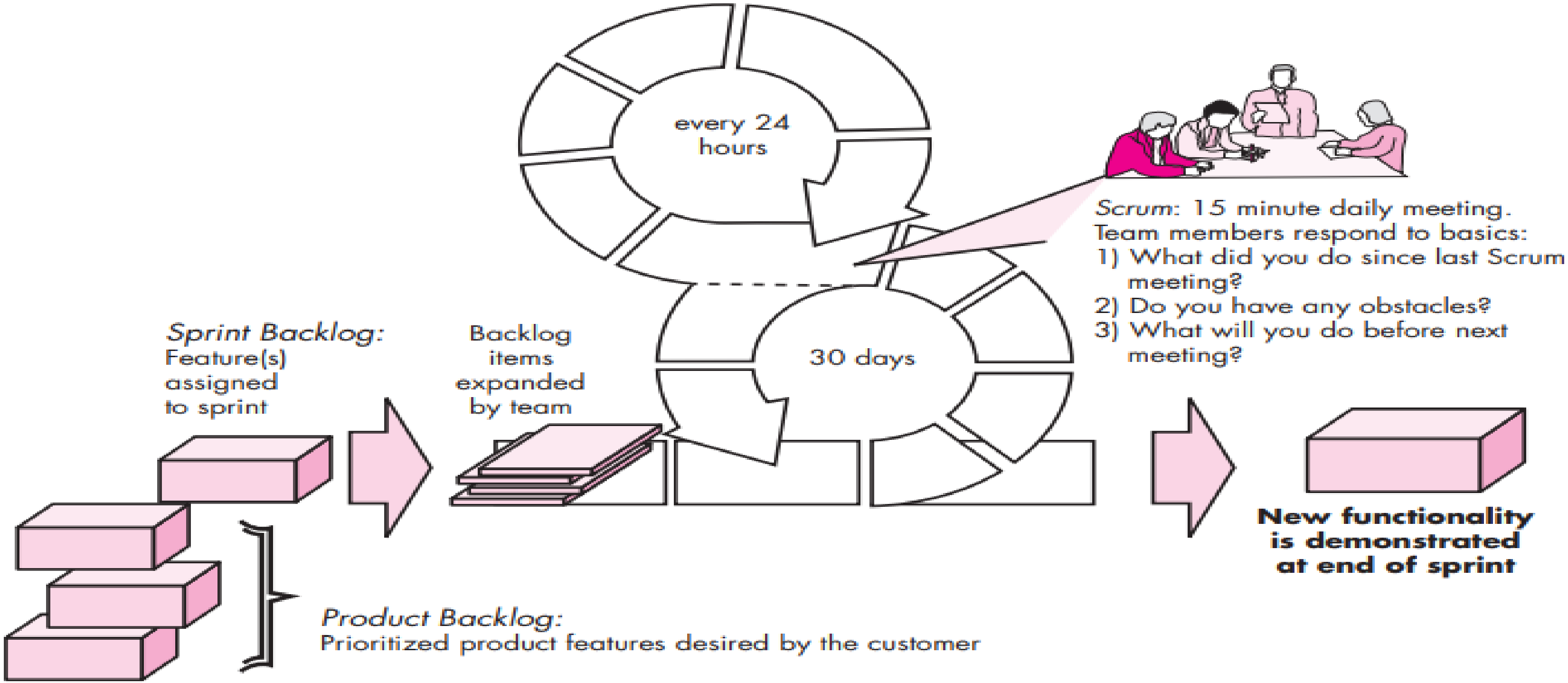
- XP is applicable for :
  - Projects involving new technology or research projects
  - Small projects

- Non suitable Project characteristics:
  - Stable requirements
  - Mission critical or safety critical systems

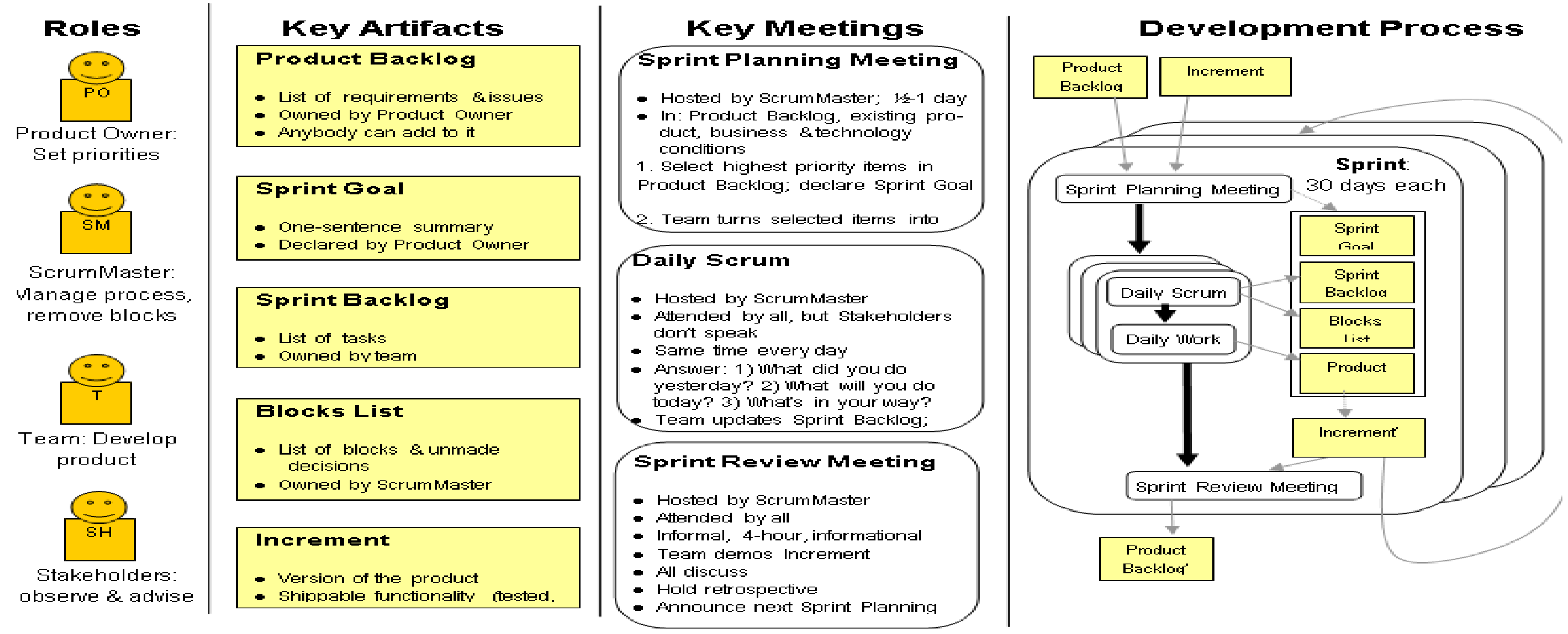
Scrum

- Developed based on the concept that software development is
  - not a defined process but an empirical process
  - with complex input/output transformations that
  - may or may not be repeated under differing circumstances.
- Scrum → distinguishing features
  - Development work is partitioned into “packets”
  - Testing and documentation are on-going as the product is constructed
  - Work occurs in “sprints” and is derived from a “backlog” of existing requirements
  - Meetings are very short and sometimes conducted without chairs
  - “demos” are delivered to the customer with the time-box allocated

Scrum


every 24 hours
Scrum: 15 minute daily meeting.
Team members respond to basics:
1) What did you do since last Scrum meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?
Sprint Backlog:
Feature(s) assigned to sprint
Backlog items expanded by team
30 days
New functionality is demonstrated at end of sprint
Product Backlog:
Prioritized product features desired by the customer

Scrum



Roles

PO
Product Owner: Set priorities

SM
ScrumMaster: Manage process, remove blocks

T
Team: Develop product

SH
Stakeholders: observe & advise

Key Artifacts

Product Backlog

- List of requirements & issues
- Owned by Product Owner
- Anybody can add to it

Sprint Goal

- One-sentence summary
- Declared by Product Owner

Sprint Backlog

- List of tasks
- Owned by team

Blocks List

- List of blocks & unmade decisions
- Owned by ScrumMaster

Increment

- Version of the product
- Shippable functionality (tested,

Key Meetings

Sprint Planning Meeting

- Hosted by ScrumMaster; ½-1 day
- In: Product Backlog, existing product, business & technology conditions

1. Select highest priority items in Product Backlog; declare Sprint Goal
2. Team turns selected items into

Daily Scrum

- Hosted by ScrumMaster
- Attended by all, but Stakeholders don't speak
- Same time every day
- Answer: 1) What did you do yesterday? 2) What will you do today? 3) What's in your way?
- Team updates Sprint Backlog;

Sprint Review Meeting

- Hosted by ScrumMaster
- Attended by all
- Informal, 4-hour, informational
- Team demos Increment
- All discuss
- Hold retrospective
- Announce next Sprint Planning

Development Process