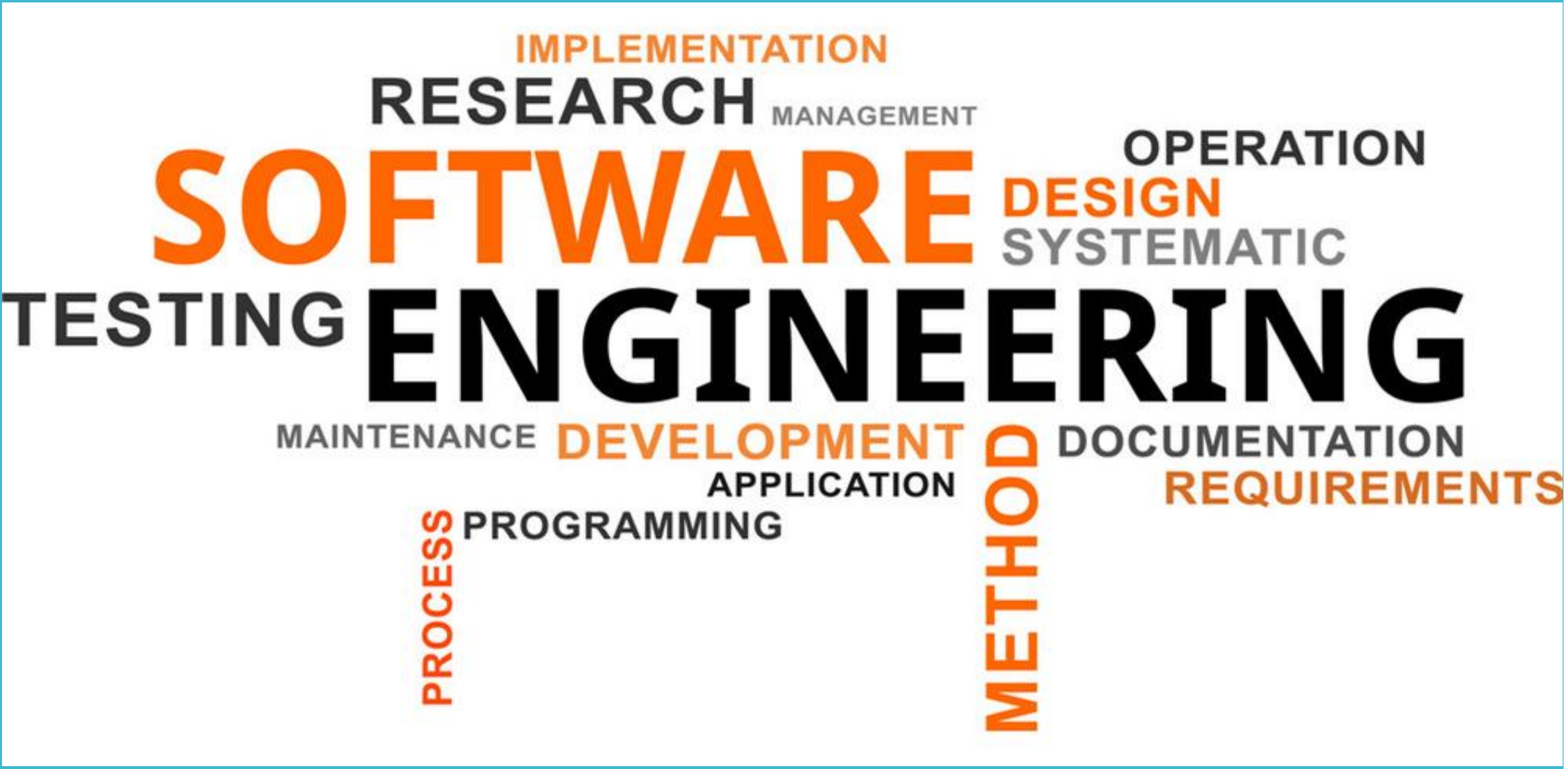


Marwadi University

Information Communication and Technology

Chapter 5
Agile Development

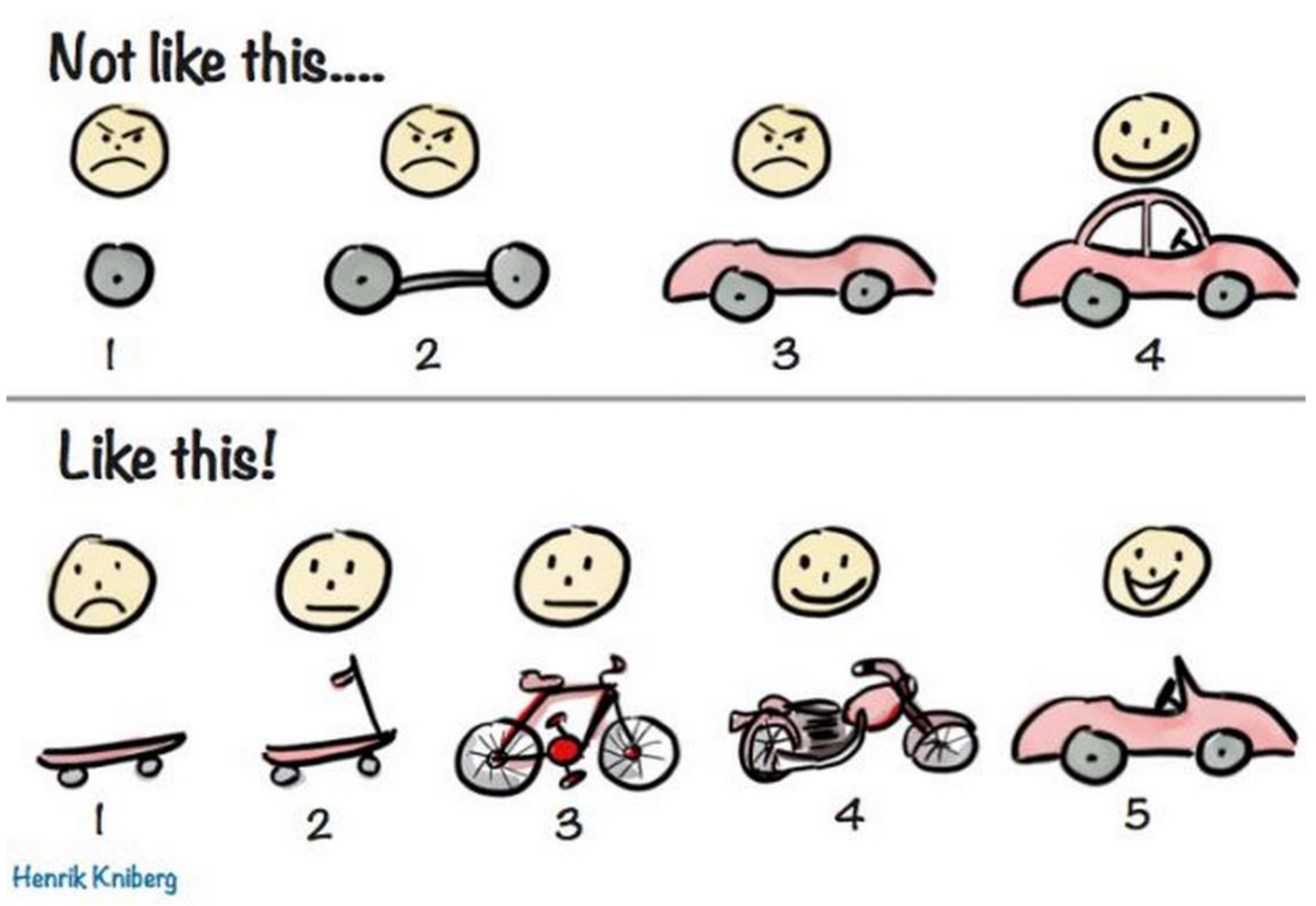
Prof. Suhag Baldaniya

Contents

What is Agile Software Development?

- Agile is a light weight methods that are people based rather than plan based.
- It forces development team to focus on software rather than design and documentation.
- The aim is to deliver the working model of the software quickly to the customer.

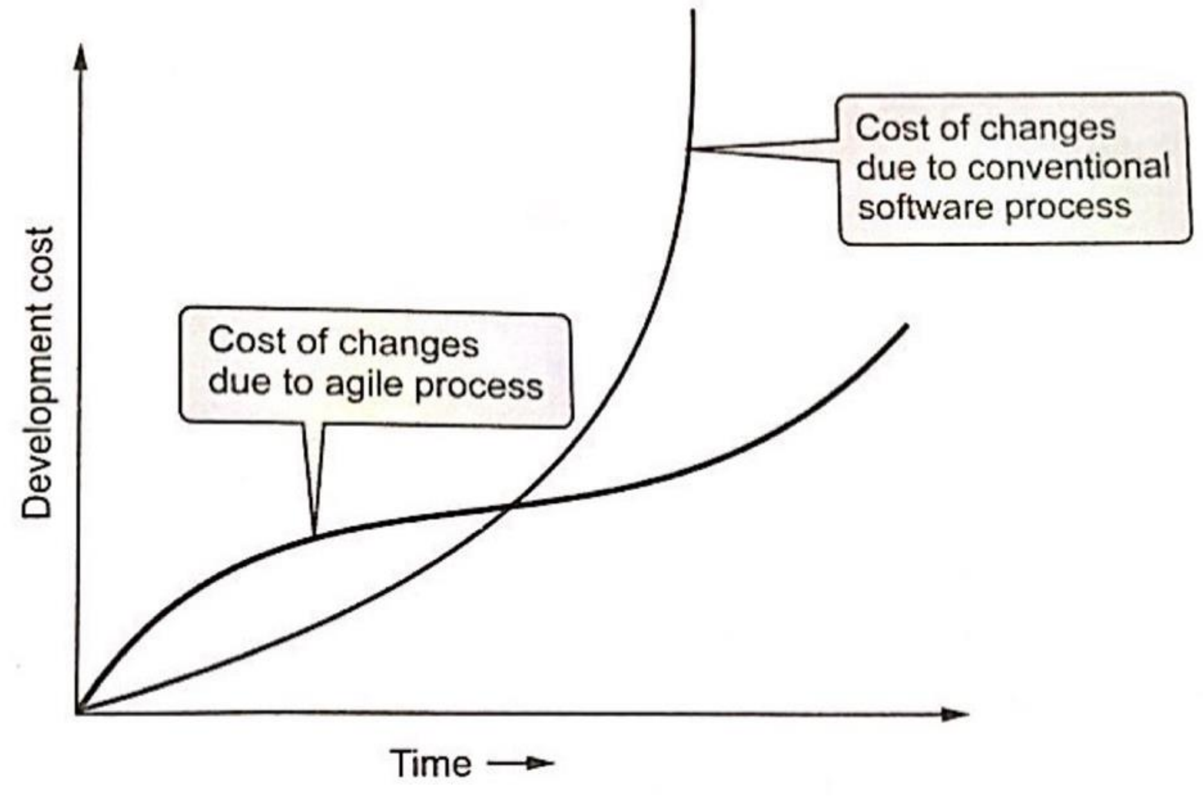
What is Agile Software Development?



Why Agile Methodology?

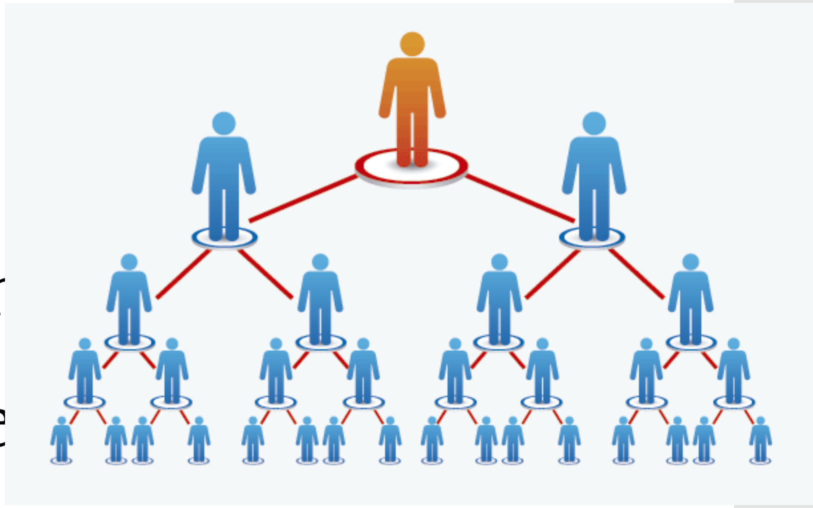
- In the conventional development method, as the project makes progress the cost of changes increases non linearly.
- The Agile methodology allow the software team to accommodate changes late in the software project without drastic cost and time impact.
- Incremental delivery + Agile practices => controlled cost changes

Agile Software Development



What is Agile Software Development?

- Agile is a term used to describe approaches to software development that employs
  - Adaptability
  - Team collaboration
  - Incremental developm
  - Learning and improve
  - Early delivery
  - Encourages flexible responses to change.



Agile Principles

- Customer satisfaction and valuable software
- Dynamic in development
- Working software is delivered frequently
- Cooperation between business people and developers
- Motivate the project developers
- F2F Communication

Agile Principles

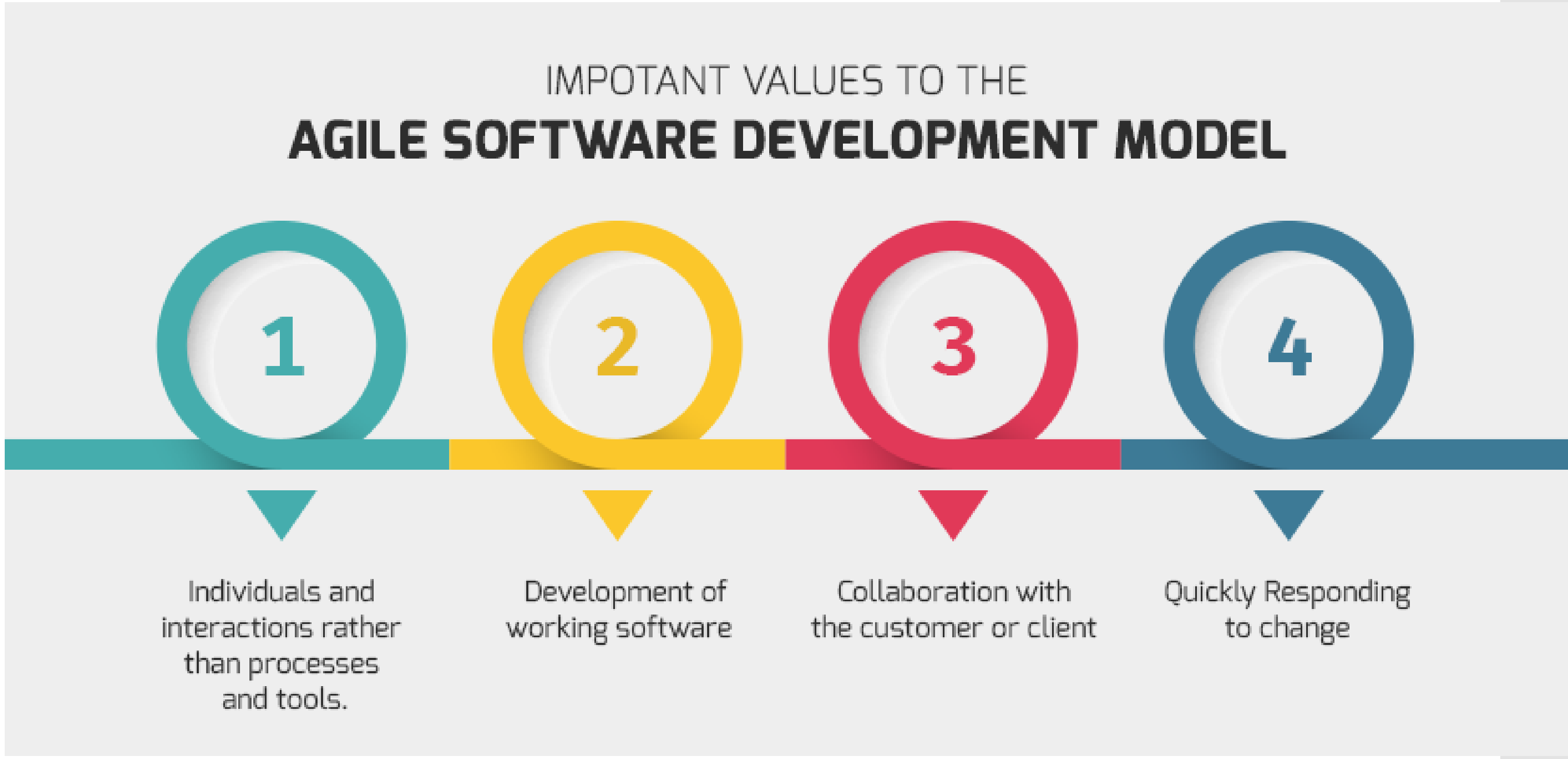
- Progressive measure
- Sustainable development
- Attention to technical design
- Simplicity
- Need of Self-organizing
- Being effective

What is Agile Software Development?

- Agile Process addresses this key assumption

1. Difficulty in predicting S/W requirement(Change) and priority of customer
2. It is difficult to predict how much of design is required before Construction.
3. Analysis, Design, Coding and Testing are not predictable as we might say it is

Manifesto for Agile Software Development


IMPOTANT VALUES TO THE
AGILE SOFTWARE DEVELOPMENT MODEL
1
Individuals and interactions rather than processes and tools.
2
Development of working software
3
Collaboration with the customer or client
4
Quickly Responding to change

Where agile methodology not work

- Project is not very urgent, too complex or novel
- Project plan & requirements are clear & unlikely to change
- Your customer requires neat documentation of each development cycle
- Unclear understanding of Agile Approach among Teams
- Big Enterprises where team collaboration is tough

Agile Model Vs Waterfall Model

Agile Model	Waterfall Model
• Agile method proposes incremental and iterative approach to software design	• Development of the software flows sequentially from start point to end point.
• The agile process is broken into individual models that designers work on	• The design process is not broken into an individual models
• The customer has early and frequent opportunities to look at the product and make decision and changes to the project	• The customer can only see the product at the end of the project
• Development process is iterative, and the project is executed in short (2-4) weeks iterations. Planning is very less.	• The development process is phased, and the phase is much bigger than iteration. Every phase ends with the detailed description of the next phase.
• It requires close communication with developers and together analyze requirements and planning	• Developer does not involve in requirement and planning process. Usually, time delays between tests and coding

Agile Process Models

- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Dynamic Systems Development Method (DSDM)
- Scrum
- Feature Driven Development (FDD)
- Crystal
- Agile Modelling (AM)

1. Extreme Programming (XP)

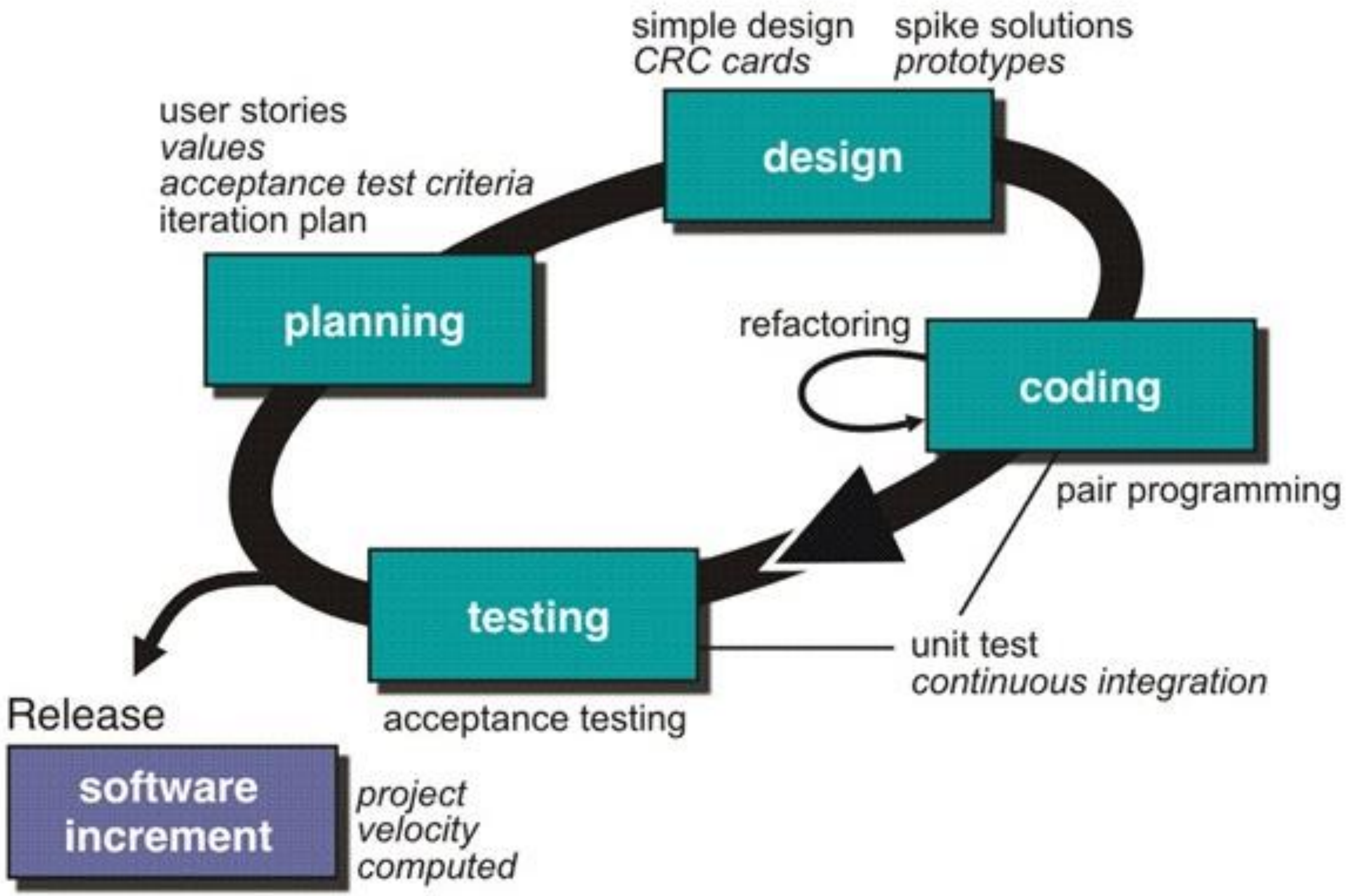
- The most widely used agile process, originally proposed by Kent Beck.
- It is lightweight, efficient, low-risk, flexible, predictable way to develop a software.
- Follows Object Oriented approach.
- Five values of XP:

1. Communication
2. Simplicity
3. Feedback
4. Courage
5. Respect

When to use?

- Constantly changing demands or requirements
- Customer are not sure about the functionality of the system
- Small projects consisting of small teams as face to face meeting is easier to achieve
- Projects involving new technology or Research projects

1. Extreme Programming (XP)



© www.researchgate.net

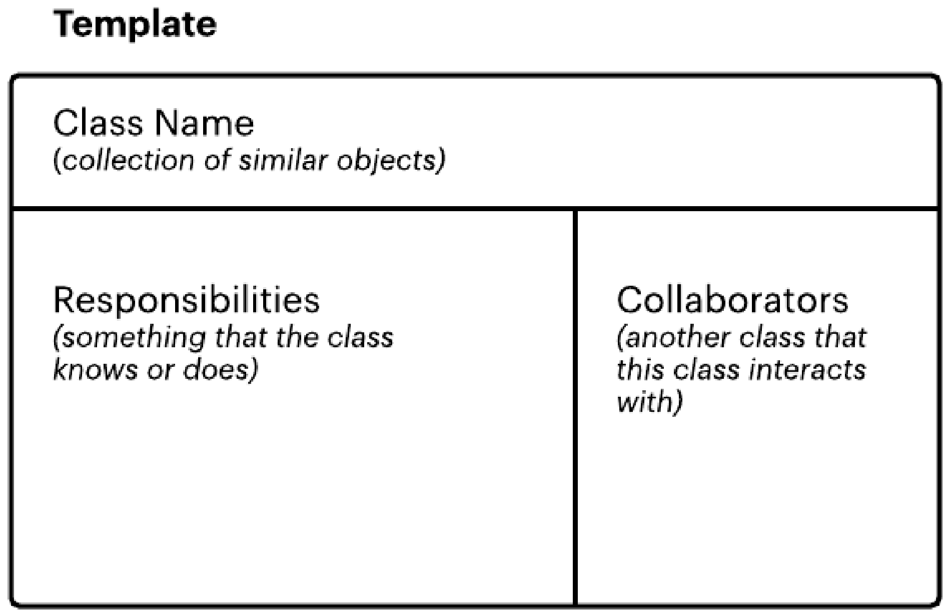
XP - Planning

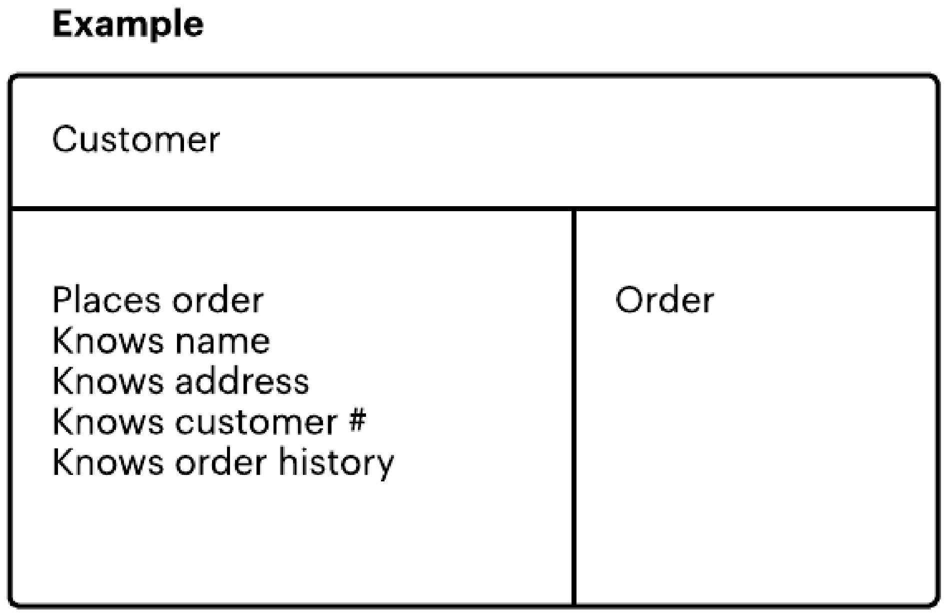
- User story-cards
- Release planning
- Small releases
- Iterative process
- Stand up meetings - F2F at same location

XP - Design

- Simple design
  - It is always good to keep the things simple to meet the current requirements
- Spike solution
  - For answering the tough technical problem
- Refactoring
  - Reduction in the redundancy, elimination of unused functionalities
- Encourage the use of CRC (class-responsibility-collaborator) cards

CRC cards (Class-Responsibility-Collaborator)


Template
Class Name
(collection of similar objects)
Responsibilities
(something that the class knows or does)
Collaborators
(another class that this class interacts with)



XP - Coding

- Pair programming - Driver/Supporter.
- Collective code ownership - All contribute, one person doesn’t become bottleneck of the project.
- Continuous integration of codes.

XP - Testing

- Unit tests should be automated.
- Encourages regression testing whenever code is modified.
- Integration and validation testing on daily basis.
- Acceptance tests are derived from user stories that have been implemented as part of software release.

Extra

Weekly Cycle

The Weekly Cycle is synonymous to an iteration

Quarterly Cycle

The Quarterly Cycle is synonymous to a release.
The customer lays out the overall plan for the team in terms of features desired within a particular quarter

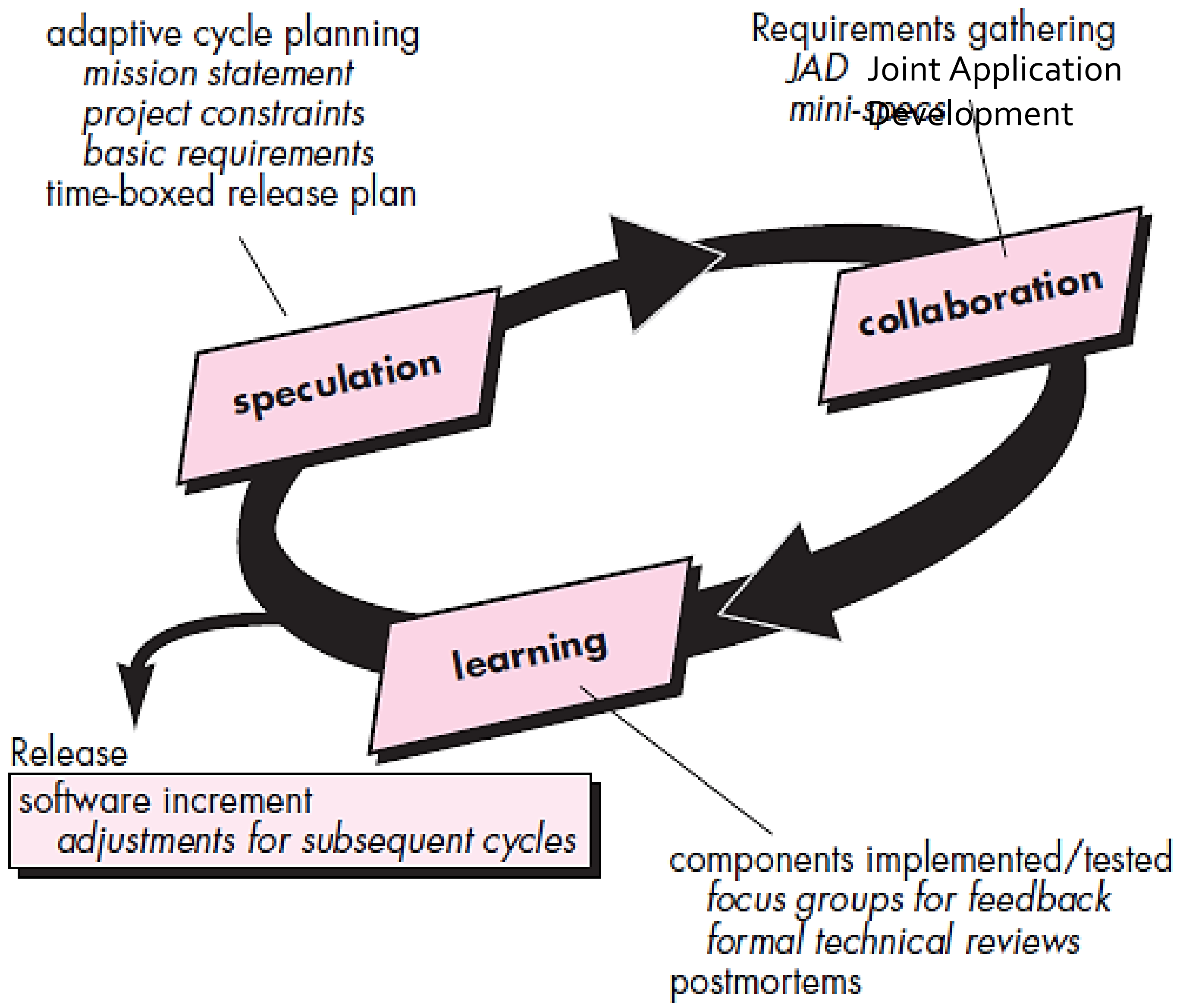
Slack

The idea behind slack in XP terms is to add some low priority tasks or stories in your weekly and quarterly cycles that can be dropped if the team gets behind on more important tasks or stories.

2. Adaptive Software Development (ASD)

- Adaptive Software Development has evolved from RAD practices.
- Jim Highsmith published Adaptive Software Development in 2000.
- This is a technique for building complex software systems using iterative approach.
- ASD focus on working in collaboration and team self-organization.

2. Adaptive Software Development (ASD)



©Roger Pressman

Speculation

- The adaptive cycle planning is conducted.
- In this cycle planning mainly three types of information is used to define the set of release cycles
  - Customer’s mission statement
  - Project constraints
    - Delivery date, budgets, user description etc.
  - Basic requirements of the project

Collaboration

- In this, collaboration among the members of development team is a key factor.
- For successful collaboration and coordination it is necessary to have following qualities in every individual
  - Assist each other without offense
  - Work hard
  - Posses the required skill set
  - Communicate problems and help each other
  - Criticize without any hate

Learning

- Emphasize on learning new skills and techniques.
- There are three ways by which the team members learn
  - Focus groups - Direct Feedback from the end-user about the component being develop.
  - Formal technical review - Performed by a committee of 3-5 people ensuring the quality of the software.
  - Postmortems- self assessment by the team and make appropriate improvements.

3. Dynamic Systems Development Methods (DSDM)

- The important aspect of DSDM is that the users are required to be involved actively, and the teams are given the “power to make decisions”.
- It is based on Pareto’s Principle (80/20 Rule).
- Based on RAD process model.
- Frequent delivery of product becomes the active focus with DSDM.
- A framework to meet tight time constraints, controlled project environment
- It uses “Iterative Prototyping” Model (where all requirements can’t be gathered at the beginning).

Need for DSDM

- Delivers working system to the organization within short duration of time.
- A DSDM project delivers an operational (working) system within 6 months.
- Guides the people to work together in such a way that business goals are profitably accomplished.
- Focusses on current requirement (as works on increments).
- Acc. to DSDM, most project fails because of people's issue rather than technology.
- Therefore, it focusses more on guiding people to work together to achieve goals profitably.

3. Dynamic Systems Development Methods (DSDM)

- The DSDM project consists of 5 phases

1. Feasibility Study

- Analyse Requirement
- Constraints/Risks involved
- Whether a right candidate for applying DSDM

3. Dynamic Systems Development Methods (DSDM)

2. Business Study

- Design basic architecture of the Application
- Identify Maintainability necessities
- Examine business value of the system, user groups involved and their requirements.

3. Functional Model (Iteration)

- Identify functional prototype (key functionalities).
- Accept plan and schedule (team formation, task division and deadline setting)
- Create functional prototype
- Review the functional prototype by customer

- The incremental Prototype with iterative cycles.

3. Dynamic Systems Development Methods (DSDM)

4. Design and build (Iteration)

- Each prototype is revisited to ensure correctness and completeness
- It also addresses the non functional requirements of the system.
- Prototype provided to the testers and end users for testing.

5. Implementation

-Software increment is placed in the working environment.

- If any changes are suggested by the client/end-user, then new increment is placed for improvement.

DSDM can be combined by XP method or ASD concept to create a combination model.

4. Scrum

- This model is developed by Jeff Sutherland and Ken Schwaber in 1995.
- Scrum is an agile process model which is used for developing the complex software systems.
- It is a lightweight process framework.
- Lightweight means the minimum overhead of the process in order to maximize the productivity.
- Iterative and incremental approach.

Scrum Principles

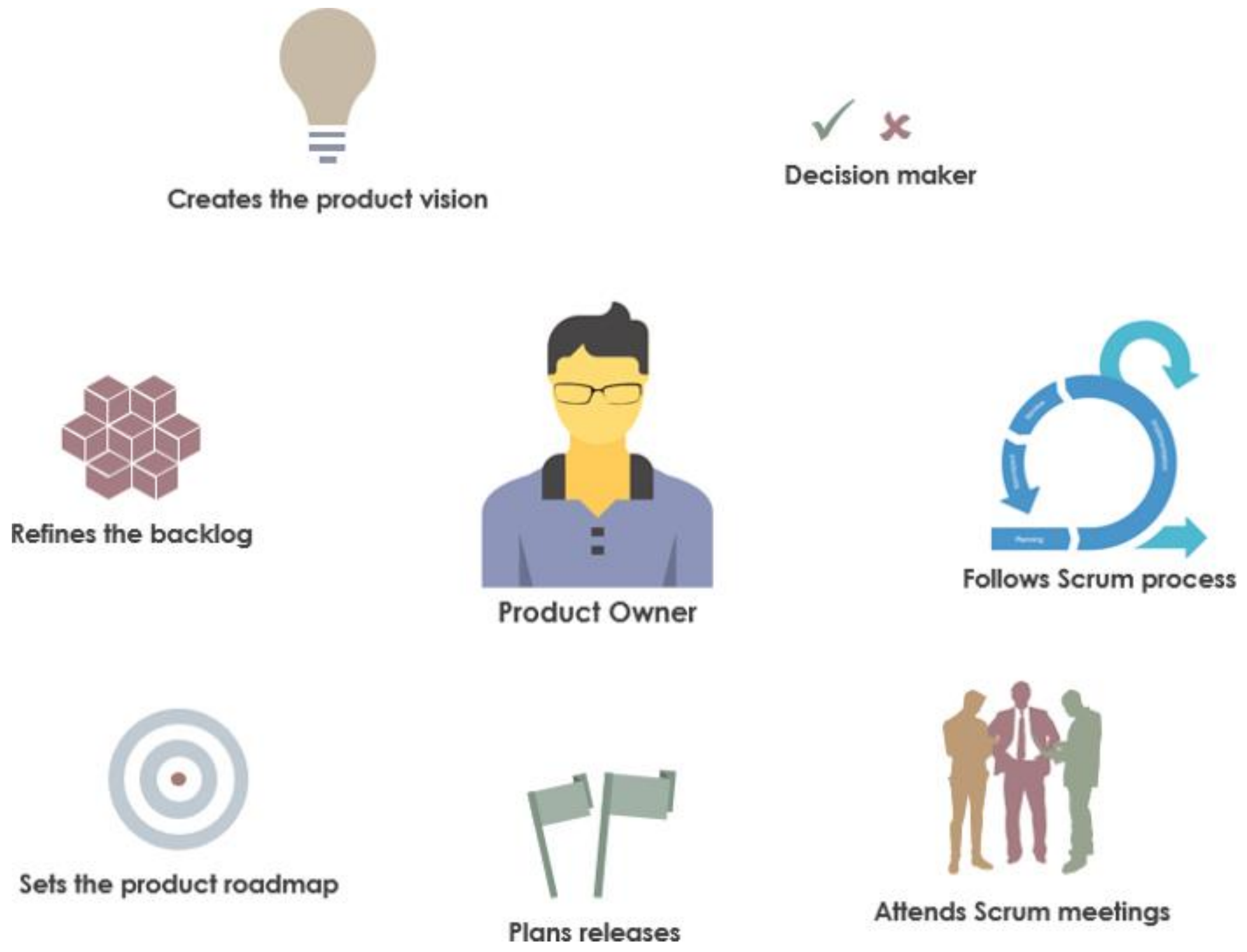
- There are small working teams on the projects due to which there is maximum communication and minimum overhead.
- The task of people must be partitioned into small and clean packets/partitions.
- The process must accommodate the technical or business changes, if they occur.
- The process should produce software increments.
- During product building, constant testing and documentation must be conducted.
- The SCRUM process must produce the working model of the product whenever required or demanded.

Roles (Scrum)

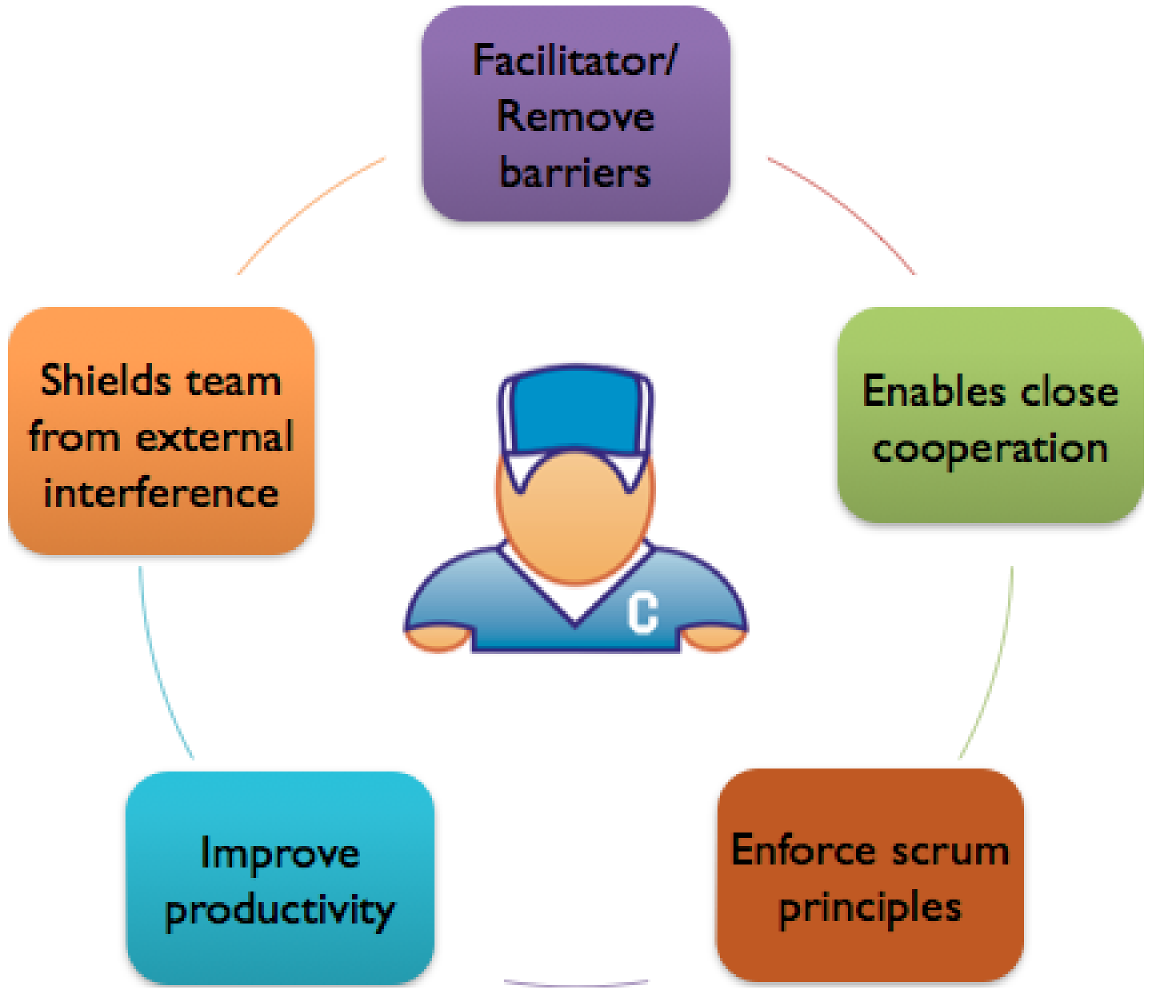
- Product Owner
  - The product owner is the project’s key stakeholder and represents users, customers and others in the process.
- Scrum Master
  - Typically a Team Leader
  - Responsible for enacting Scrum values and practices
  - Remove impediments / politics, keeps everyone productive

- Project Team
  - 5-10 members; Teams are self-organizing
  - Cross-functional: QA, Programmers, UI Designers, etc.
  - This team does not include any of the traditional SE roles such as programmer, designer, tester or architect.
  - Everyone on the project works together to complete the set of work they have collectively committed to complete within a sprint

Roles Product Owner



Roles
Scrum Master

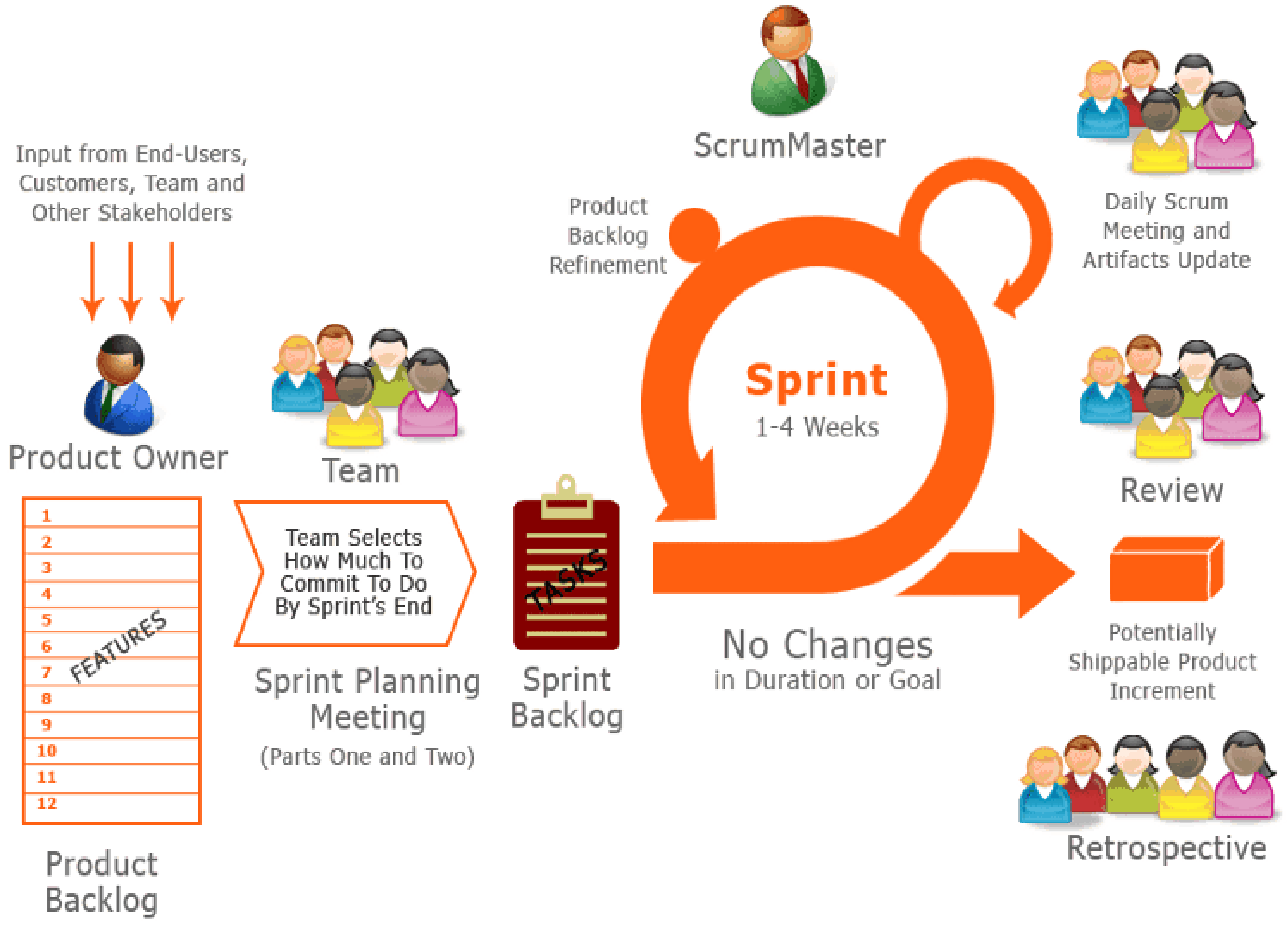

Facilitator/ Remove barriers
Enables close cooperation
Enforce scrum principles
Improve productivity
Shields team from external interference
C

Some more Terminology

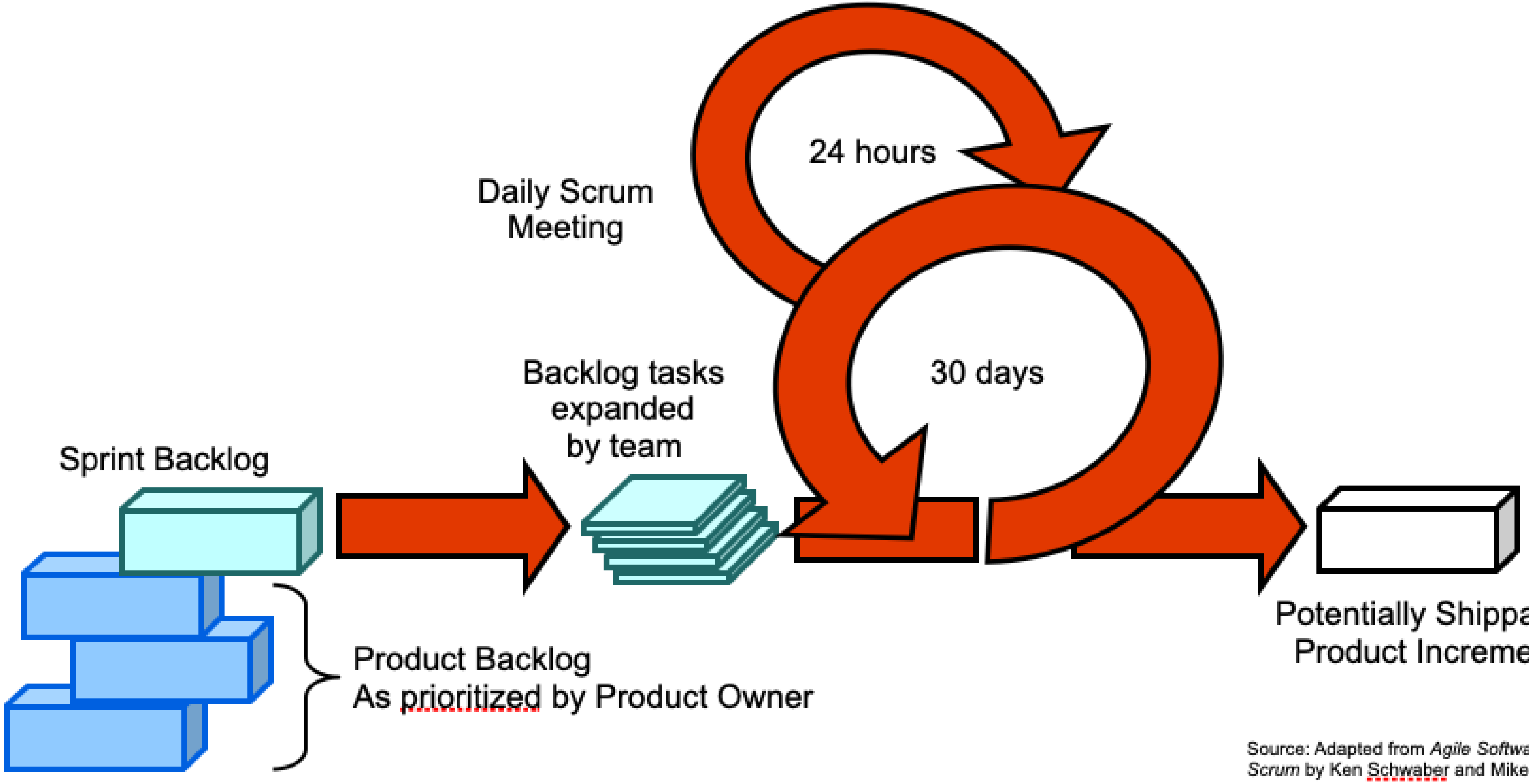
Product backlog: The product backlog is a prioritized features list containing every desired feature or change to the product.

Sprint backlog:The sprint backlog is a list of tasks to be completed in a sprint.

Scrum


Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
Team
ScrumMaster
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Sprint
1-4 Weeks
Review
1
2
3
4
5
6
7
8
9
10
11
12
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
Retrospective
Product Backlog

Scrum at a Glance


24 hours
Daily Scrum Meeting
30 days
Backlog tasks expanded by team
Sprint Backlog
Product Backlog
As prioritized by Product Owner
Potentially Shippable Product Increment

Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

Sprint Planning

- At the start of each sprint, a sprint planning meeting is held, during which the product owner presents the top items on the product backlog to the team.
- The Scrum team selects the work they can complete during the coming sprint.
- That work is then moved from the product backlog to a sprint backlog, which is the list of tasks needed to complete the product backlog items the team has committed to complete in the sprint.

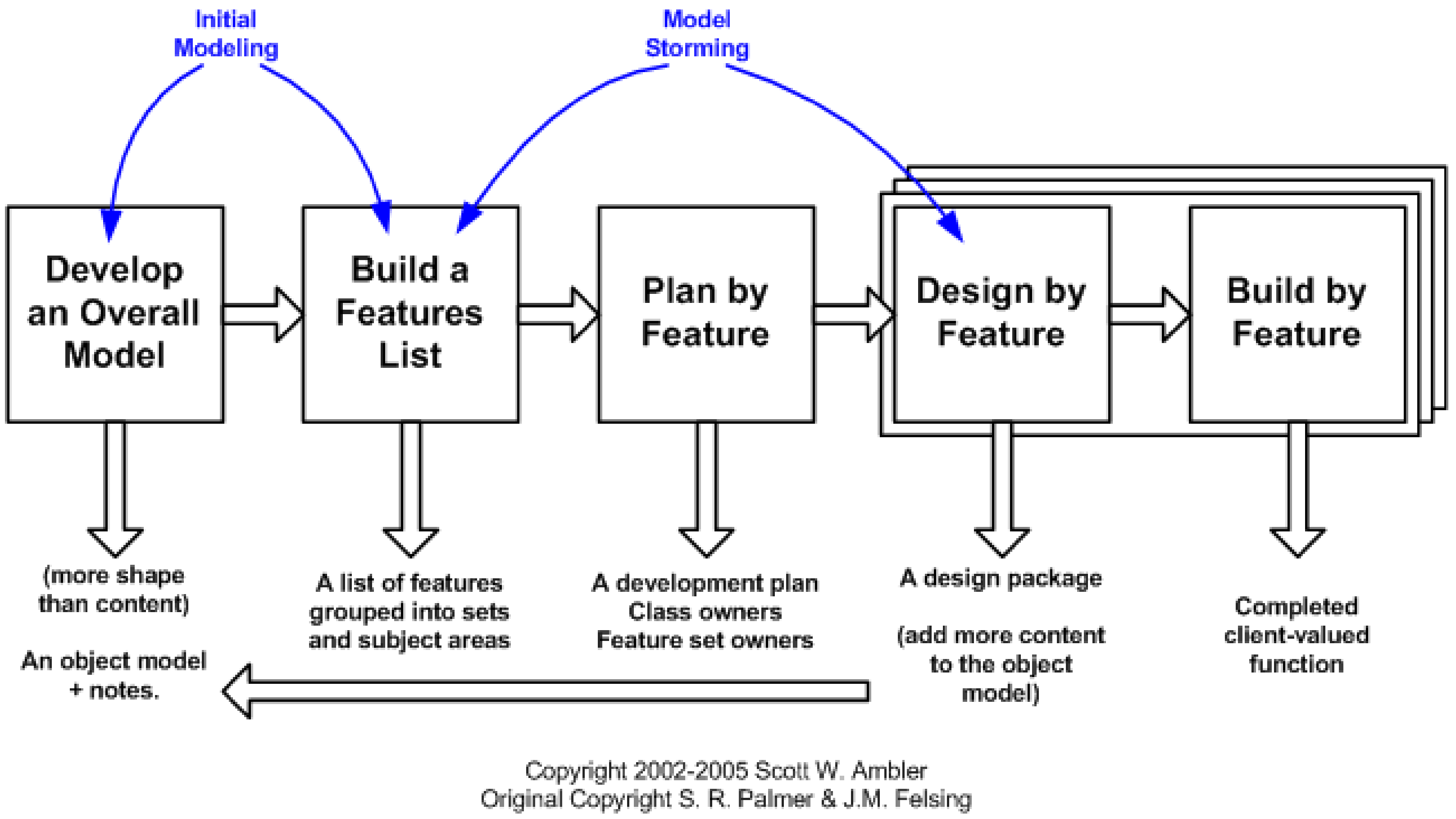
Scrum Development Activities

- Backlog
  - It is a list of project requirements or features that must be provided to the customer.
  - The items can be included in the backlog at any time.
  - The product manager analyses this list and updates the priorities as per the requirements.
- Sprint
  - These are the work units that are needed to achieve the requirements mentioned in the backlogs.
  - Typically the sprints have fixed duration or time box (1 to 4 weeks).

Scrum Development Activities

- Meetings
  - There are 15 minutes daily meetings to report the completed activities, obstacles and plan for next activities.(Development Team)
  - Following are three questions that are mainly discussed during the meetings.
    1. What are the tasks done since last meeting ?
    2. What are the issues that team is facing ?
    3. What are the next activities that are planned ?
- Demo
  - During this phase implemented functionalities are demonstrated to the customer
- Review:
  - Activity to Introspect and Adapt

5. Feature Driven Development (FDD)



Initial Modeling
Model Storming
Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than content)
An object model + notes.
A list of features grouped into sets and subject areas
A development plan
Class owners
Feature set owners
A design package
(add more content to the object model)
Completed client-valued function
Copyright 2002-2005 Scott W. Ambler
Original Copyright S. R. Palmer & J.M. Felsing

5. Feature Driven Development (FDD)

- FDD combines many of the best practices of other agile models
- FDD was initially created for and is more geared towards large project teams
- FDD puts less focus on initial design and quickly gets to the point where the team can deliver new functionality to the project feature by feature
- In FDD, the feature means client valued function.

5. Feature Driven Development (FDD)

- Various phases in the FDD life cycle

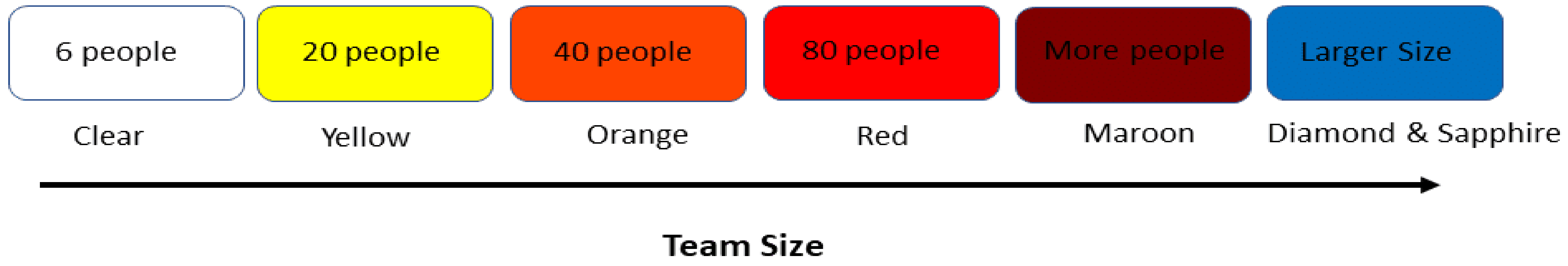
1. Develop overall model
2. Build feature list
   - <action> the <result> <by| for |to> a(n) <object>
   - Issue Book by student
   - Make Payment by Customer
3. Plan by feature
4. Design by feature
5. Build by feature

6. Crystal

- Crystal method is an agile software development approach that focuses primarily on people and their interactions when working on a project rather than on processes and tools
- Properties of Crystal method:
  1. Frequent delivery
  2. Reflective improvement
  3. Communication
  4. Personal Safety
  5. Focus
  6. Easy access to expert/users
  7. Technical Tools

The Crystal Family:

- Crystal method depends on three dimensions:
  - First, Team size
  - Second, Criticality
  - Third, the priority of the project



6. Crystal

- CLEAR - 6 or less people involved, small projects, requires some documentation.
- YELLOW - 7 to 20 people, Small to medium project, feedback is collected from customer, automated testing is conducted.
- ORANGE - 20-40 people, medium project, teams are divided as per functional skills, 1-2 years, incremental development with delivery every 3-4 months.
- RED - 40-80 people, Large projects, teams are formed as per work required,
- MAROON - 80-200 people, similar to crystal red.
- Diamond and Sapphire - very large teams, used in extremely critical project, high risk is involved

7. Agile Modeling (AM)

- Agile Modeling can be used for large and complex projects.
- The problem can be partitioned effectively among the group of people.
- The quality of working model can be assessed at every step of development.
- Scott Ambler described the Agile Modeling as
  - Agile modeling is a collection of values, principles and practices for modeling the software that can be applied on a software development project in an effective and light-weight manner.
  - AM is a practice-based methodology for effective modeling and documentation.

7. Agile Modeling (AM)

- Various features suggested for AM
  1. Specify the purpose for the model
     - Goal and objective of the model must be known to developers
  2. Make use of multiple models
     - Set of models can be useful for building desired software product
  3. Follow a definite path
     - Use only those models that give long term value

7. Agile Modeling (AM)

4. Give importance to contents and not the presentation
   - Correct information is more important than the representation
5. Understand the models and supporting tools
   - Understand the strengths and weakness of the model and tools
6. Adapt locally
   - Needs of modelling must be satisfied by adopting appropriate model

7. Agile Modeling:- Core Principles

- AM core Principles:-
  1. Model with a purpose
  2. Assume Simplicity
  3. Embrace Change
  4. Enabling the Next Effort is Your Secondary Goal
  5. Incremental Change
  6. Maximize Stakeholder Investment/Needs
  7. Multiple Models
  8. Quality Work
  9. Rapid Feedback
  10. Software is Your Primary Goal
  11. Travel Light

THANK YOU