
CE343 SOFTWARE ENGINEERING

Chapter 2 Agile Software development

Subject Teacher:

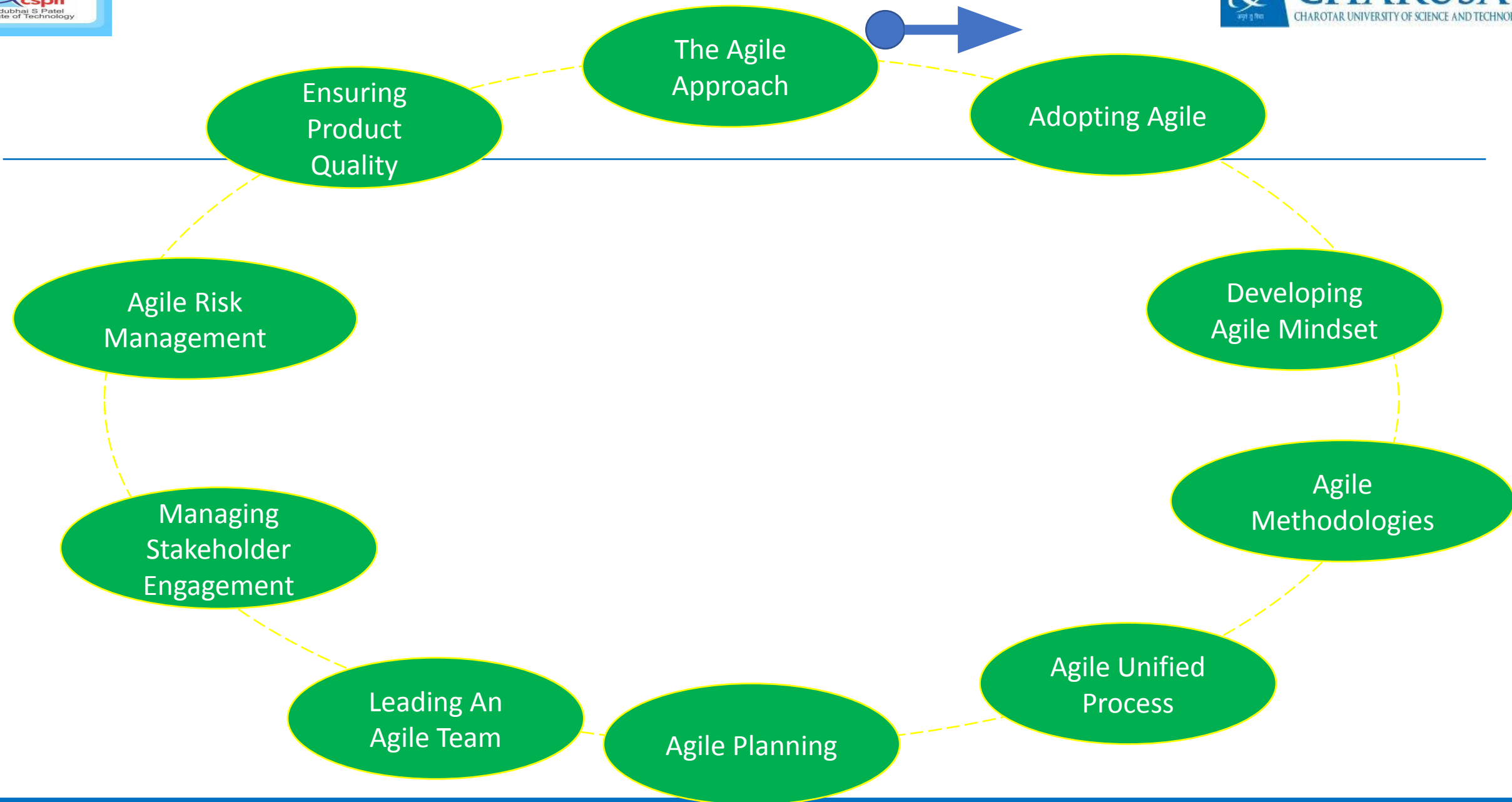
Ms. Sneha Padhiar

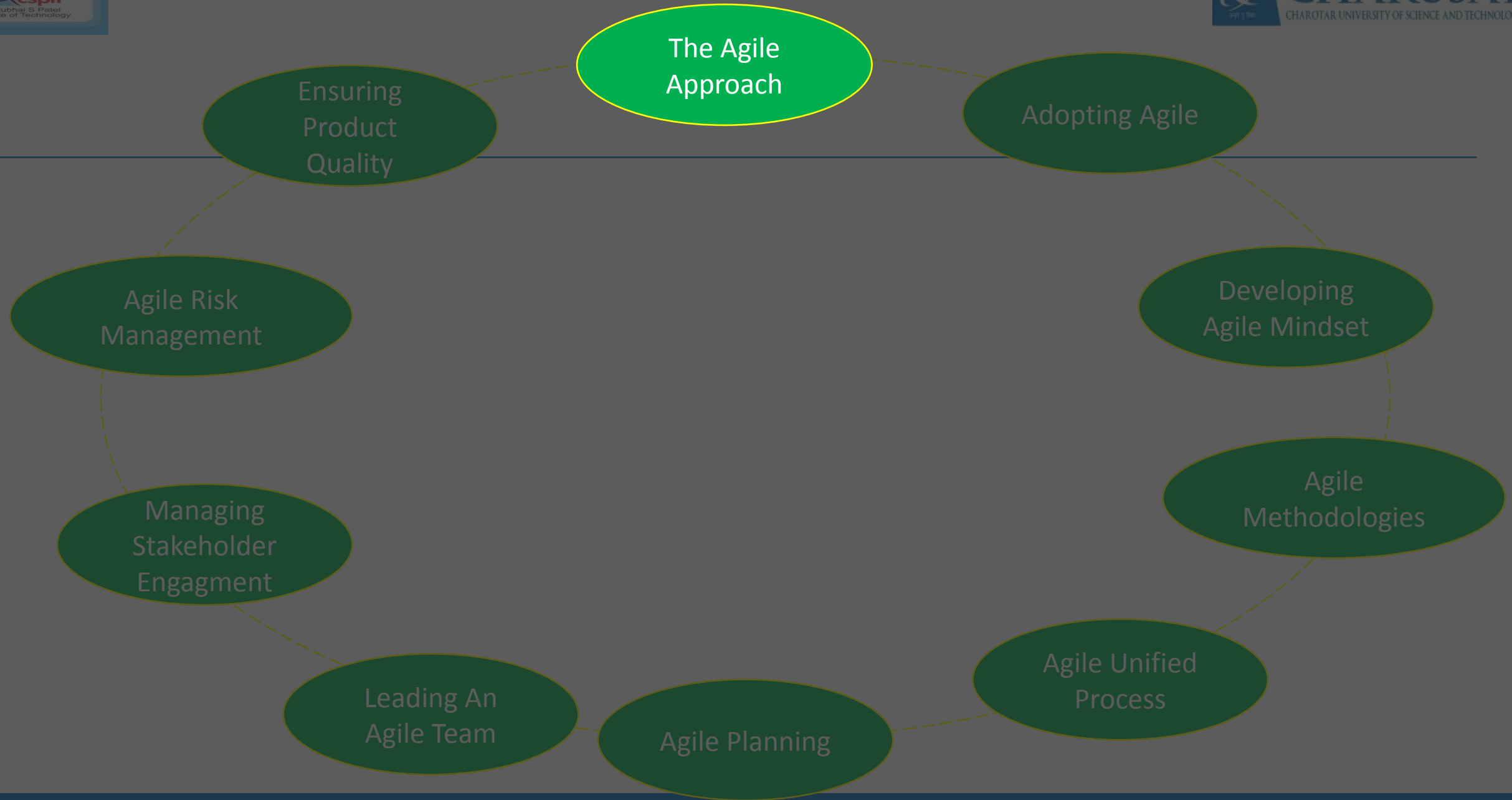
Assistant Professor,

U & P U. Patel Department of Computer Engineering

WHAT IS AGILE?

- Traditional approach to managing software development projects was failing far too often, and there had to be a better way
- Agile development is a different way of managing IT development teams and projects
- Came up with the agile **manifesto** , which describes 4 important values that are as relevant today



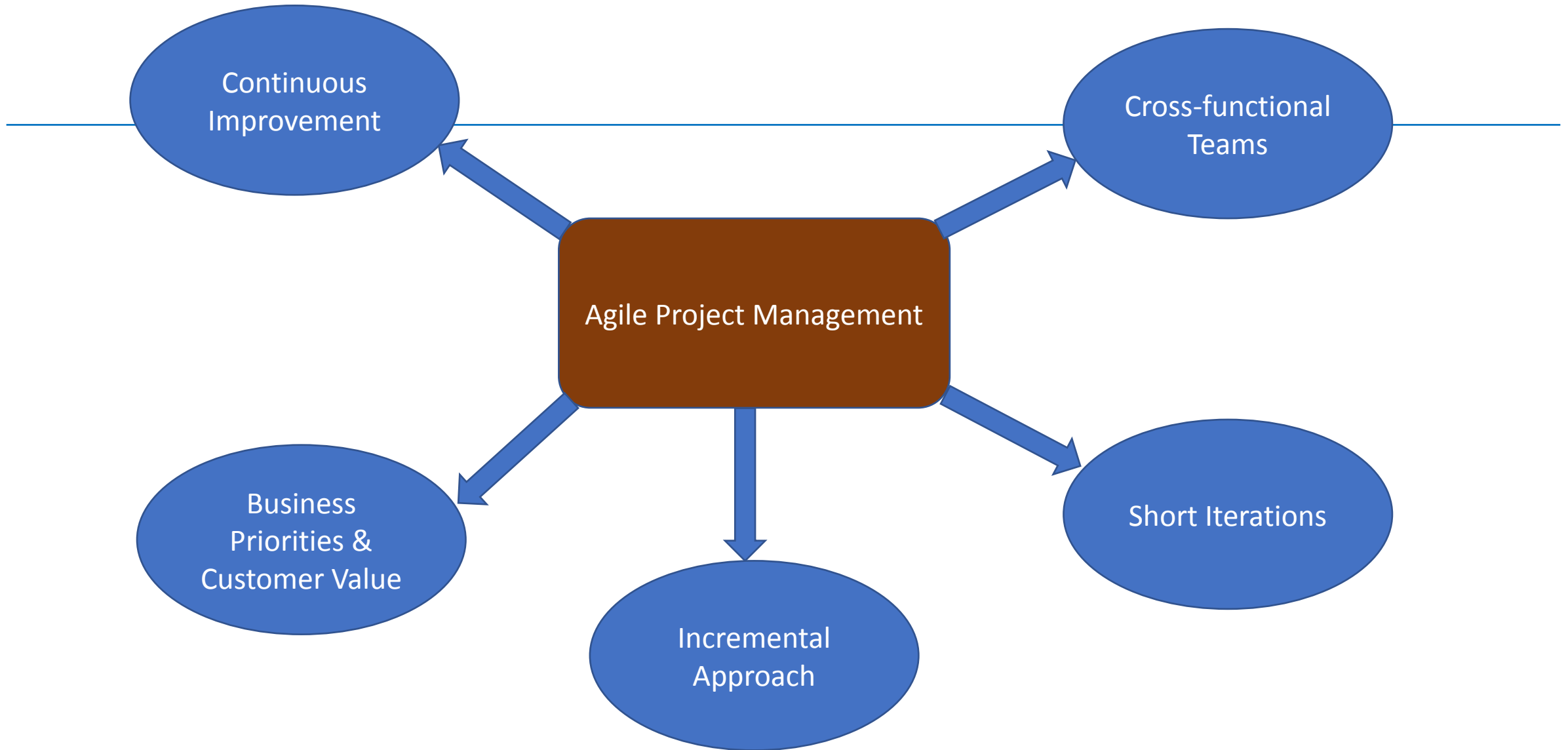


The Agile Approach

- assumes that things will change
- make suitable adjustments

Agile project management objectives

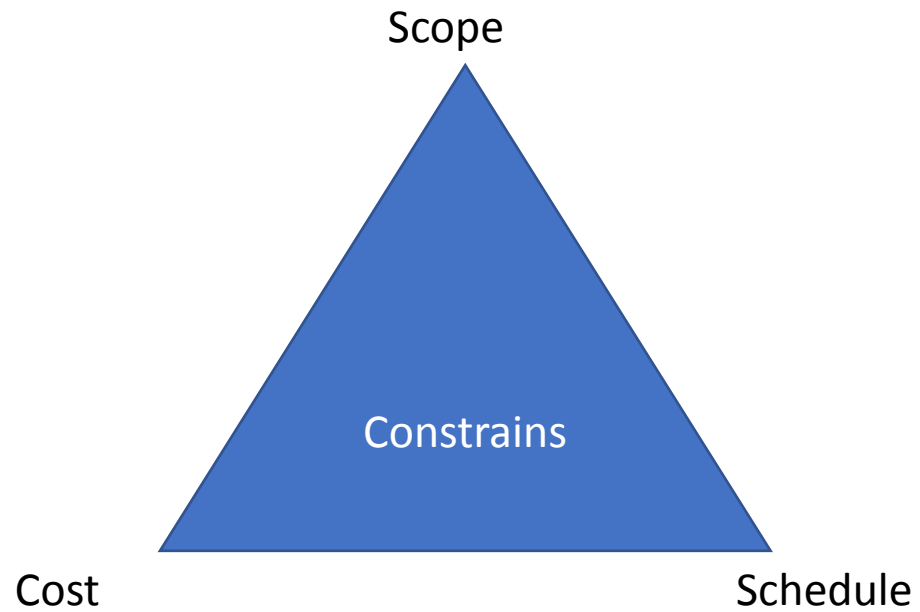
- respond to change
- uses resources effectively
- foster strong ties with customers
- create customer solutions
- increase capacity (to deliver more business value)



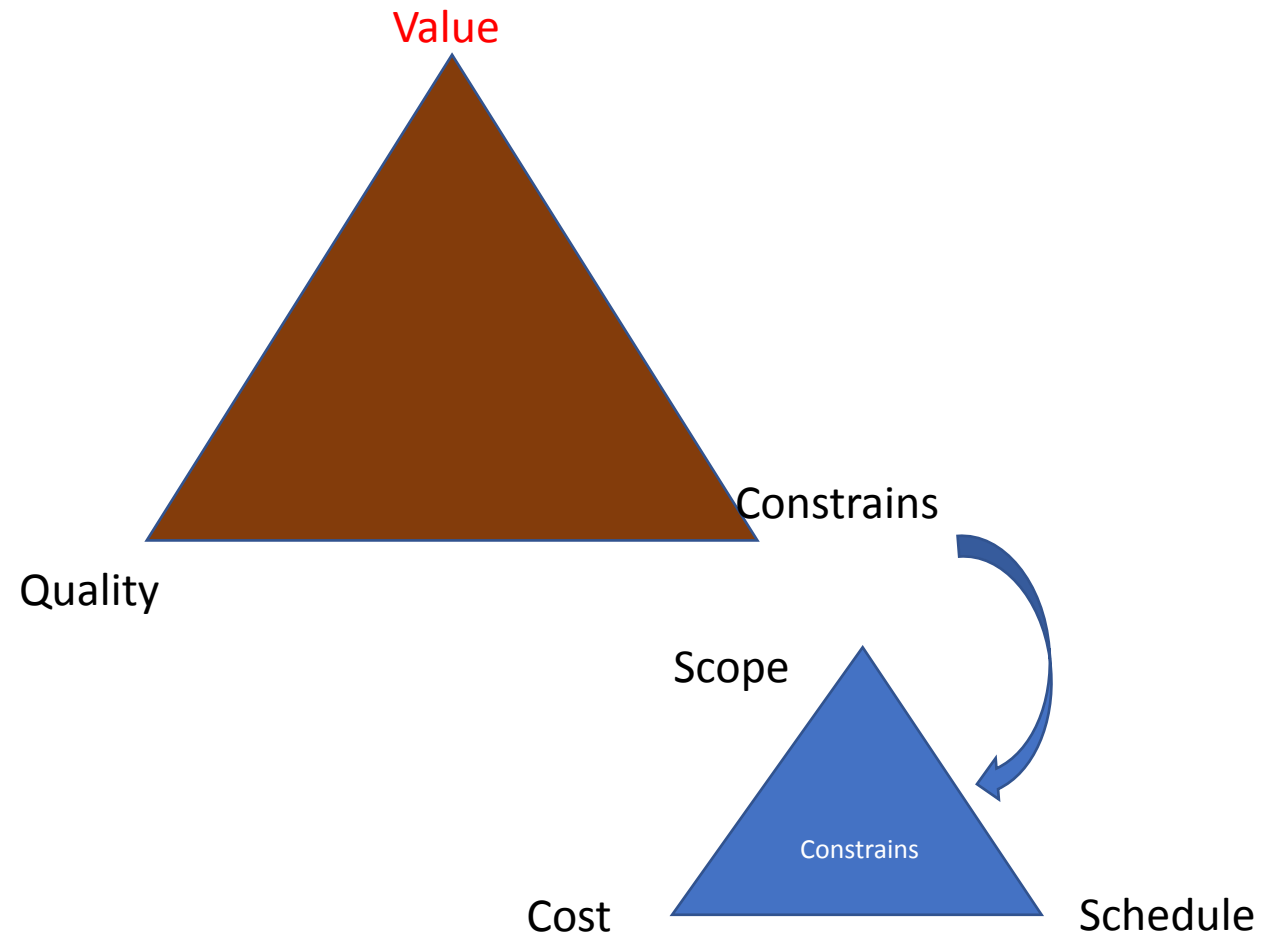
project plan

- scope
 - cost
 - schedule
 - activities
 - deliverables
 - milestones
 - resources
- AGILE PROJECT PLAN
 - DETAILED ENOUGH TO GET STARTED WITH FIRST ITERATION OF WORK
 - FOCUSES ON WHICH PRODUCT FEATURES TO DEVELOP AND WHEN
 - DOES NOT DESCRIBE ACTIVITIES IN DETAIL

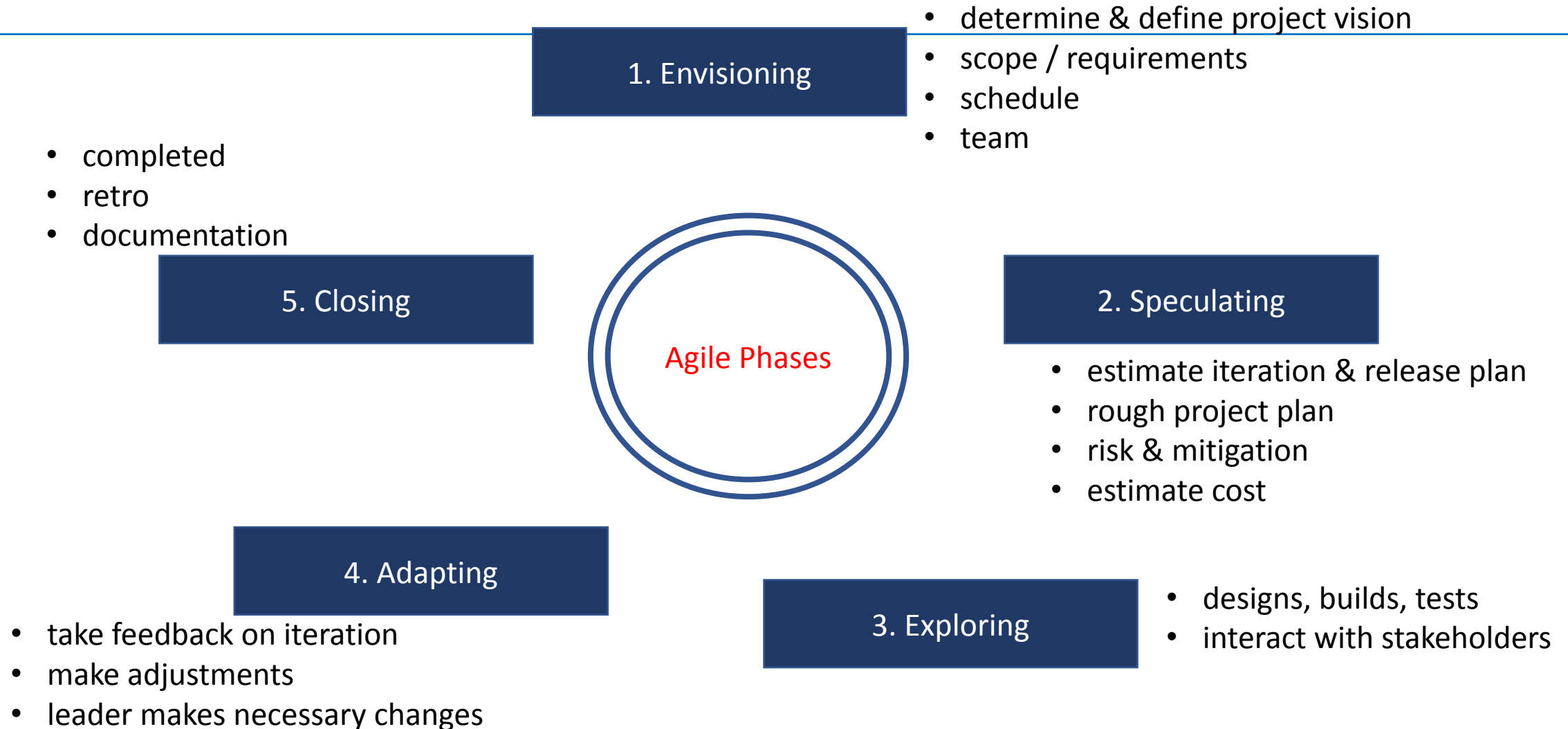
Traditional Iron Triangle



Agile Triangle



Agile project management phases



how agile project management differs

1. high level project scope
2. multiple iterations
3. self-organizing teams
4. extensive customer involvement

roles unique to agile

- PROJECT LEADER (INSTEAD OF MANAGER)
 - FOCUS ON HOW TEAM CAN ACHIEVE ITS GOALS
 - COACH
 - MORE EXPLICIT LEADERSHIP EARLY IN PROJECT'S LIFE
 - HELPS TEAM INTERACT WITH REST OF THE ORGANISATION
 - ALSO A BUSINESS ANALYST

Roles ...

- business analyst
 - if a task is taking longer than expected, BA ensures customer understands why
 - manage project requirements effectively
 - helps manage relationship between onsite customer & other team members
 - helps customer know constraints of the team

roles ...

- scrum master
 - measure progress using tools like burndown chart and scrum task boards
 - ensure project moving in right direction (acts like sheep dog -> guide & guard)
 - if delay occurs, sm's job is to determine the cause and rectify if possible
 - take leadership role if team isn't self regulating
 - conduct stand-up meetings
 - layers of scrum masters for larger teams

roles ...

- product owner / manager
 - customer representative
 - documenting vision of the product
 - sharing vision with stakeholders
 - gathering stakeholders feedback to use in generating user stories
 - prioritising release planning
 - be available to work on product iteration

agile methodologies

- scrum
- extreme programming
- lean development

SCRUM METHODOLOGY:

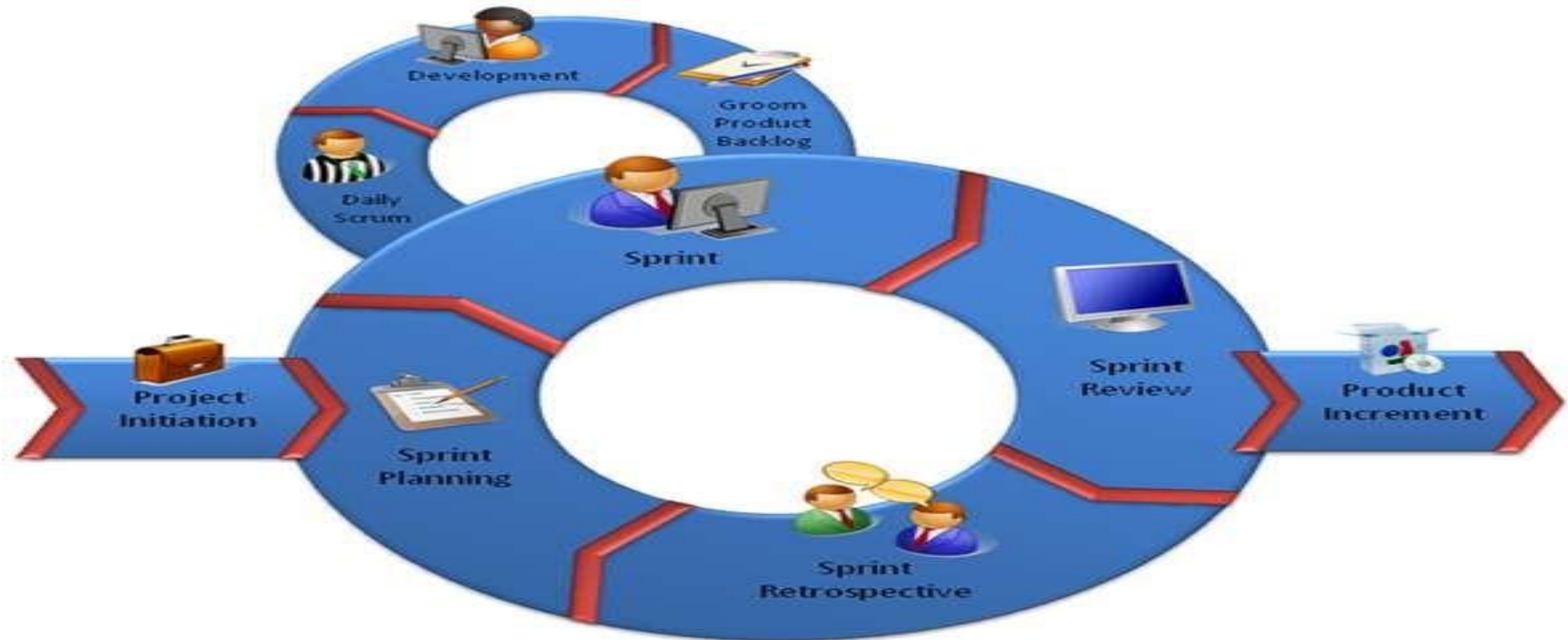
- Is also an agile development method, which concentrates particularly on how to manage tasks within a team-based development environment.
- Scrum is the most popular and widely adopted agile method
- Relatively simple to implement and addresses many of the management issues that have plagued IT development teams for decades

SCRUM METHODOLOGY

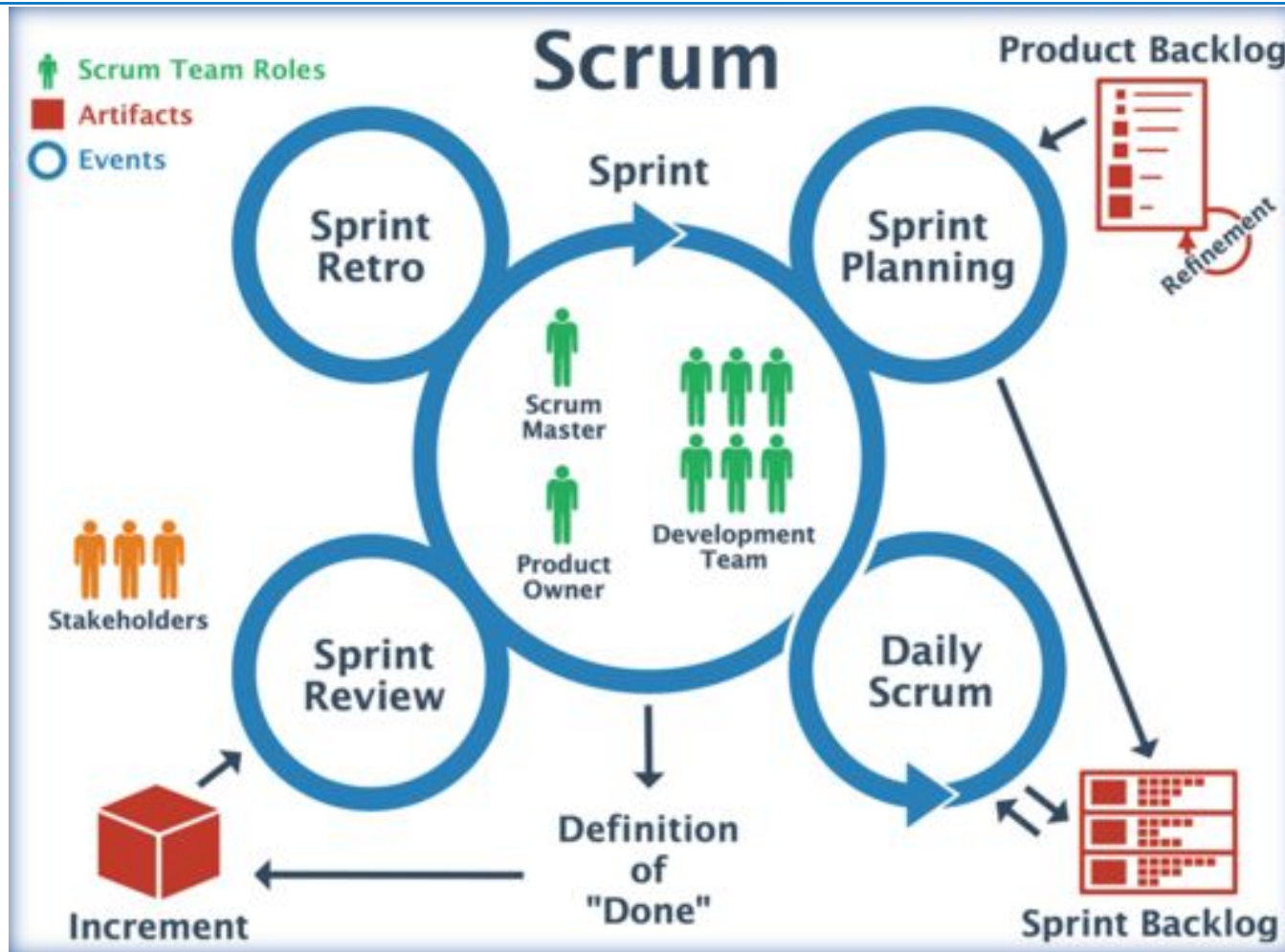
- A product owner creates a prioritized wish list called a product backlog.
- During sprint planning, the team pulls a small chunk from the top of that
wish list, a sprint backlog, and decides how to implement those pieces.
- The team has a certain amount of time — a sprint (usually two to four weeks) — to complete its work, but it meets each day to assess its progress (daily Scrum).
- Along the way, the Scrum Master keeps the team focused on its goal.
- At the end of the sprint, the work should be potentially shippable: ready to hand to a customer, put on a store shelf, or show to a stakeholder.
- The sprint ends with a sprint review.
- As the next sprint begins, the team chooses another chunk of the product backlog and begins working again-

scrum

- 10 – 15 minutes daily meetings
 - what work & why
 - what's next
 - blocked?
- fixed length iteration / sprint
 - 2-weeks / 4-weeks
 - agreed work items to complete
- planning occurs between sprints
- 2 type of backlogs
 - product - includes every possible project requirement, all features, functions, enhancements. It is single source of truth. changes can occur
 - sprint – work items to be completed for a sprint. No changes preferred
- burndown chart for work item tracking / progress



scrum



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 the value:
 1. **Individuals and interactions** over processes and tools
 2. **Working software** over comprehensive documentation
 3. **Customer collaboration** over contract negotiation
 4. **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.”

PS: Please look at these 4 values and 12 Principles of Agile on Shared video on MS Team Conversation of channel.

10 KEY PRINCIPLES OF

1. Active user involvement is imperative
2. The team must be empowered to make decisions
3. Requirements evolve but the timescale is fixed
4. Capture requirements at a high level
5. Develop small, incremental releases and iterate

6. Focus on frequent delivery of products
7. Complete each feature before moving on to the next
8. Apply the 80/20 rule (The Pareto Principle, commonly referred to as the 80/20 rule, states that **80% of the effect comes from 20% of causes**. Or, in terms of work and time management, 20% of your efforts will account for 80% of your results.)
9. Testing is integrated throughout the project lifecycle – test early and often
10. A collaborative & cooperative approach between all stakeholders is essential

eXtreme Programming

- A system of practices that a community of software developers is evolving to address the problems of quickly delivering quality software, and then evolving it to meet changing business needs.
- It aims to produce higher quality software, and higher quality of life for the development team.
- XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

eXtreme...

Taking proven practices to the extreme

- If testing is good, let everybody test all the time
- If code reviews are good, review all the time
- If design is good, refactor all the time
- If integration testing is good, integrate all the time
- If simplicity is good, do the simplest thing that could possibly work
- If short iterations are good, make them really, really short

Where XP is appropriate:

<http://www.extremeprogramming.org/when.html>

- Dynamically changing software requirements
- Risks caused by fixed time projects using new technology
- Small, co-located extended development team
- The technology you are using allows for automated unit and functional tests
- Where not appropriate??
<http://wiki.c2.com/?WhenIsXpNotAppropriate>

XP values

- Communication
- Simplicity
- Feedback
- Courage

XP practices

- The Planning Game*
- Small Releases
- Metaphor
- Simple Design*
- Testing*
- Refactoring*
- Pair Programming*
- Collective Ownership
- Continuous Integration
- 40-Hour Week
- On-Site Customer
- Coding Standards
- Open workspace
- Daily Schema migration

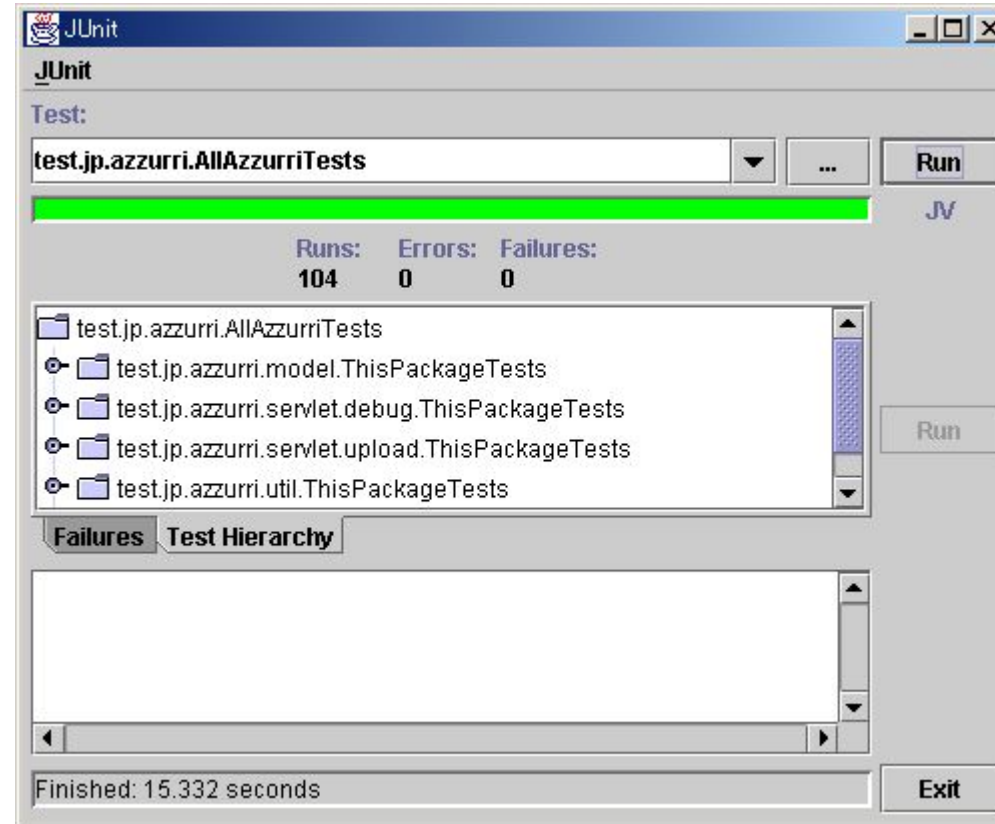
The Planning Game

- Business writes a story describing desired functionality
- Stories are written on index cards
- Development estimates stories
- Velocity determines number of stories per iteration
- Business splits and prioritizes stories and determines the composition of releases
- Velocity is measured and adjusted every iteration
- Customer steers development

Testing

- Unit Tests and Functional Tests
- Test a little, code a little...
 - “Test-first programming”
- Tests become the specification
- Tests give confidence in the system
- Tests give courage to change the system

Unit tests



Pair Programming

- Two people looking at one machine, with one keyboard and one mouse
- Two roles: implementation and strategy
- All production code is written in pairs



Pair Programming Benefits

- 15% less time and better than 2 solo programmers
- Continuous code review: better design, fewer defects
- Confidence to add to or change the system
- Discipline to always test and refactor
- Teach each other how the system works (reduced staffing risks)
- Learn from partner's knowledge and experience (enhances technical skills)

Simple design

Do the simplest thing that could possibly work

- Passes all the tests
- No duplicate code
- States every intention
- Fewest possible classes and methods

Refactoring

- Design becomes everybody's daily business
- Continuously improve quality of the code
- Unit Tests and Pair Programming give courage
- Code **Refactoring** is the process of clarifying and simplifying the design of existing code, without changing its behavior.
- **Agile** teams are maintaining and extending their code a lot from iteration to iteration, and without continuous **refactoring**, this is hard to do.

Result:

- Fast development speed
- Code becomes easy to change

Why XP works

- Light-weight: discipline without bureaucracy
- Under stress, people do what is easiest
 - All XP practices have short-term benefits as well as long-term benefits
- Development as a Conversation
- The code is the documentation
- XP is fun

Who benefits from XP?

Programmers:

- get clear requirements & priorities
- can do a good job
- can make technical decisions
- don't work overtime

Customers:

- get most business value first
- get accurate feedback
- can make informed business decisions
- can change their mind

XP

- programmer focussed, disciplined
- rapid releases
- small self directed team of programmers
- estimate programming task based on customer provided user stories
- ...-> simplicity of design -> Testing -> Refactoring -> Feedback ->...
- Principles of XP
 - pair programming
 - sustainable pace (not encouraged to work more than 40 hours a week. work 40 hrs consistently)
 - ongoing automated testing
 - TDD (test driven development)

Scrum Vs XP

Scrum	Extreme Programming (XP)
In Scrum framework, team work in iterations called Sprint which are 1-2 month long.	In Extreme Programming (XP), teamwork for 1-2 weeks only.
Scrum model do not allow changes in their timeline or their guidelines.	Extreme Programming allow changes in their set timelines.
Scrum emphasizes self-organization.	Extreme Programming emphasizes strong engineering practices
In Scrum framework, team determines the sequence in which the product will be developed.	In Extreme Programming, team have to follow a strict priority order or pre-determined priority order.
Scrum framework is not fully described. If you want to adopt it then you need to fill the framework with your own frameworks method like XP, DSDM or Kanban.	Extreme Programming (XP) can be directly applied to a team. Extreme Programming is also known for its Ready-to-apply features.

other methodologies of Agile

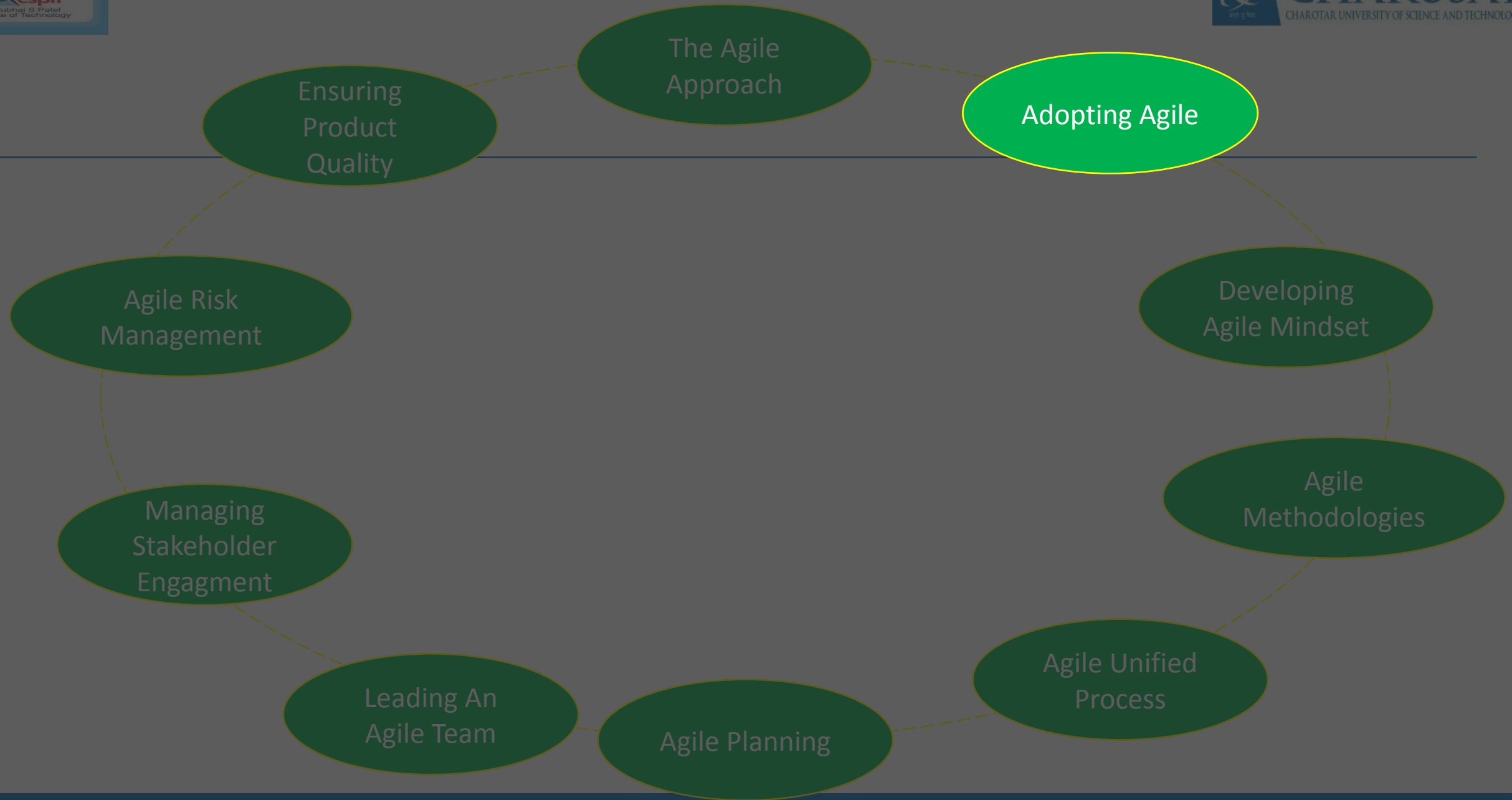
- crystal – based on colours of crystal hardness (clear, yellow, orange, red)
- feature dd
- Tdd(Test driven development)
- dsdm (dynamic systems development method)
- asd (adaptive systems development)
- unified process

Quiz time!

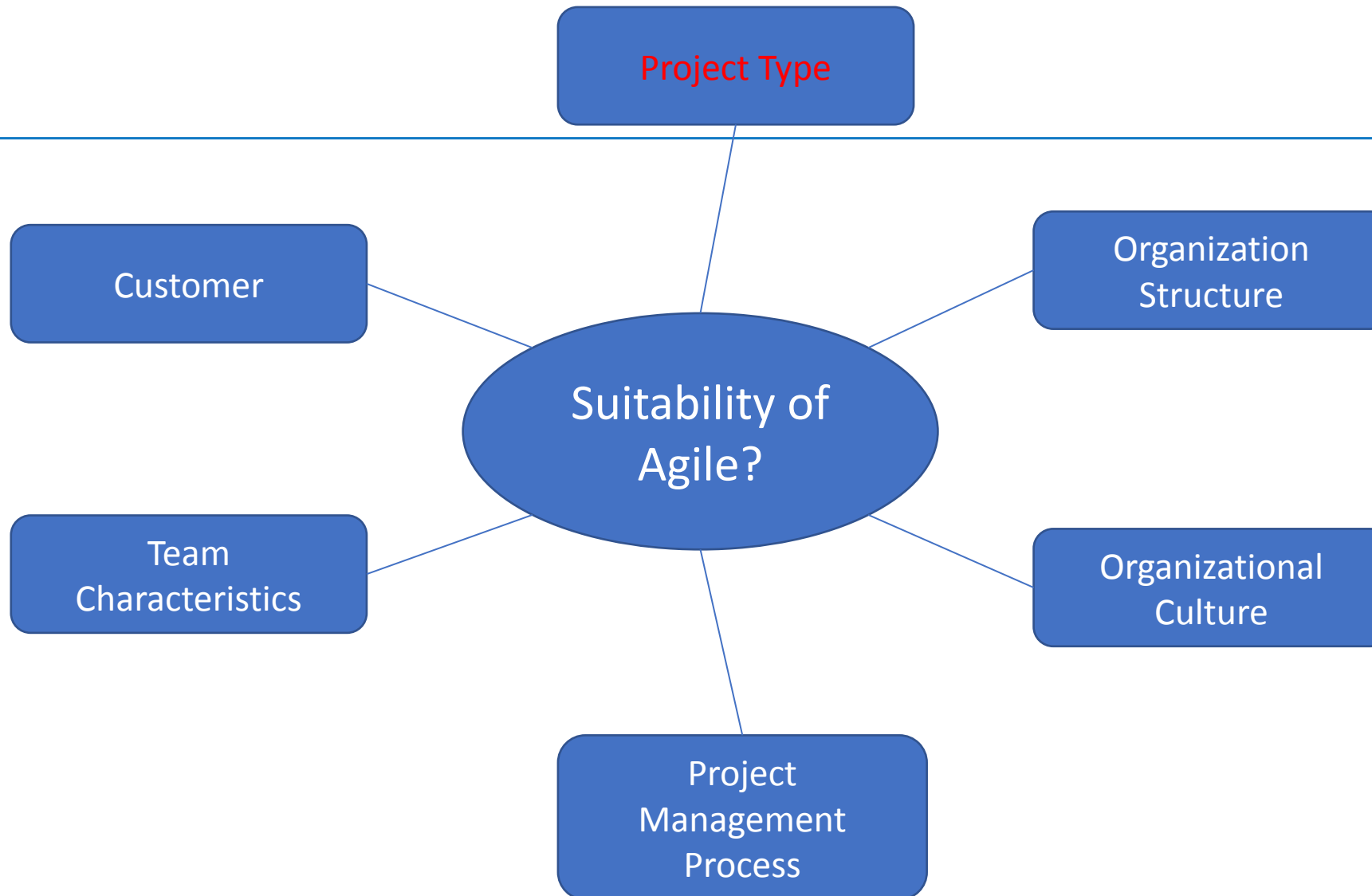
- What is the single most important thing that the agile process measures its success by?
- Why does agile approach advocates acceptance of change and doing adjustments on the fly?
- What needs to be done to get started on a project using agile?
- Present any experience that you may have in implementation of a project that you did in your professional careers?

exercise

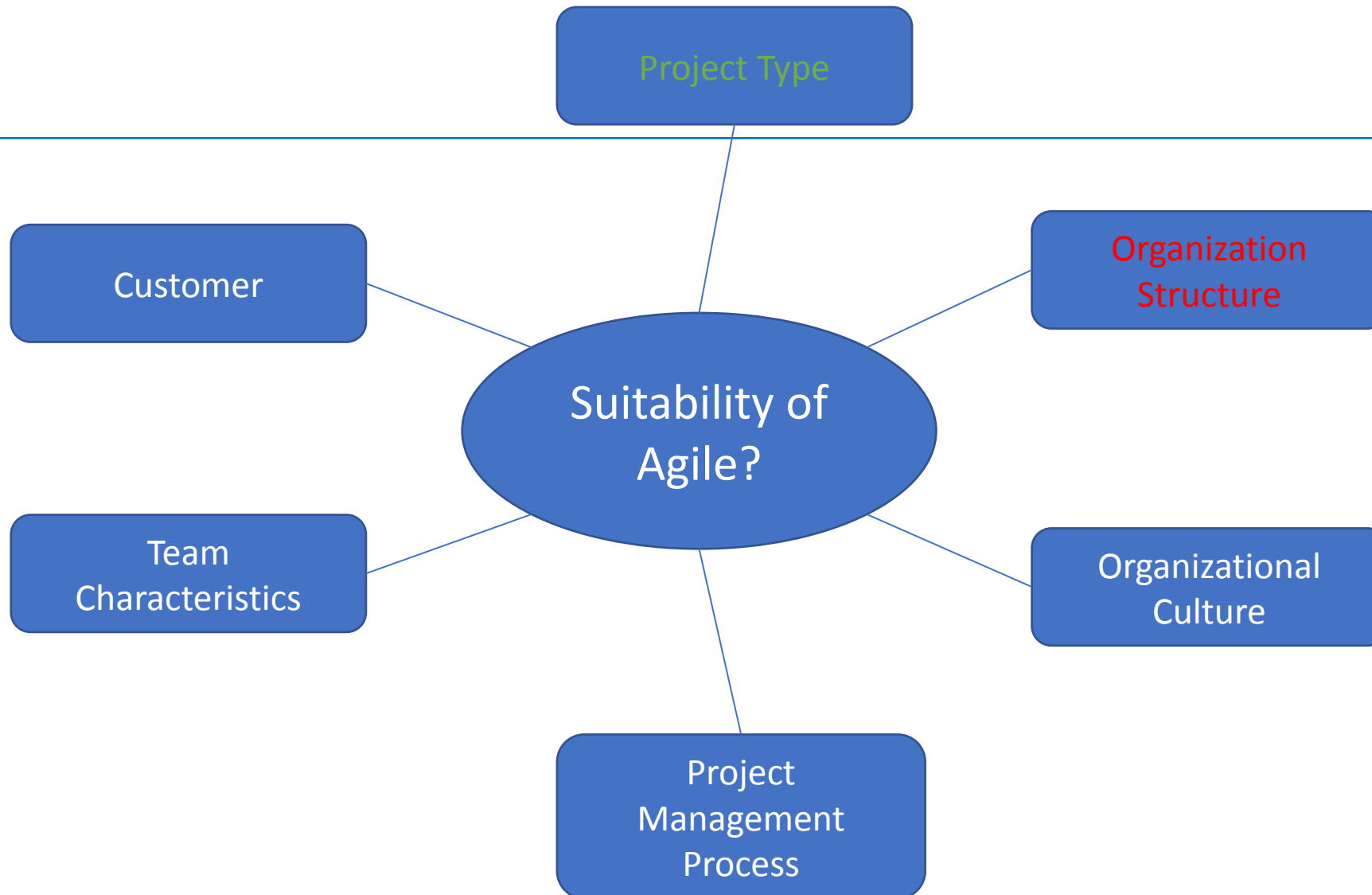
- Let's try pair programming as advocated in XP
 - make team of two each
 - we are going to build a console application for movie ticket booking
 - you can use any programming language that you want



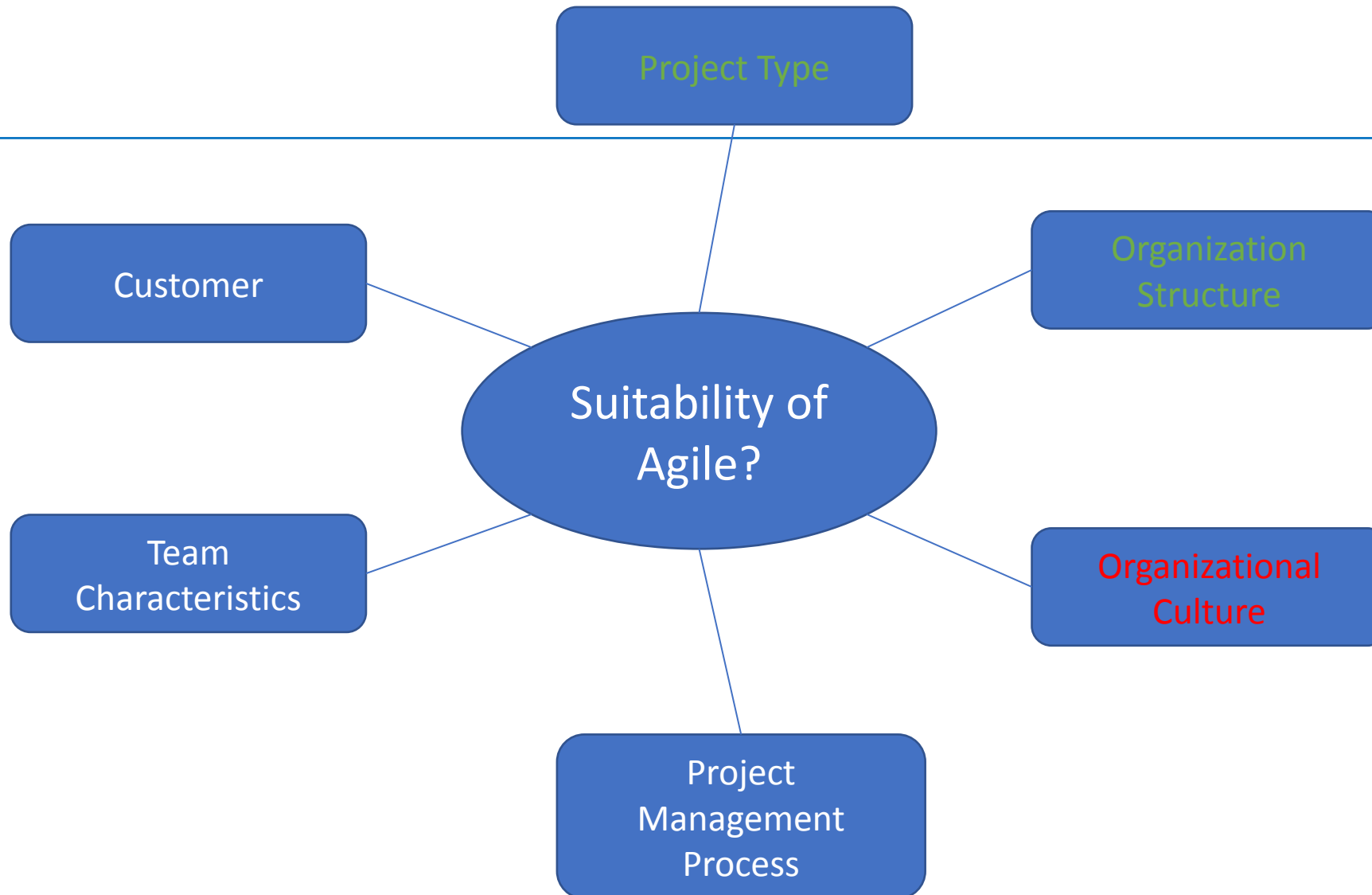
Adopting an Agile approach to project management



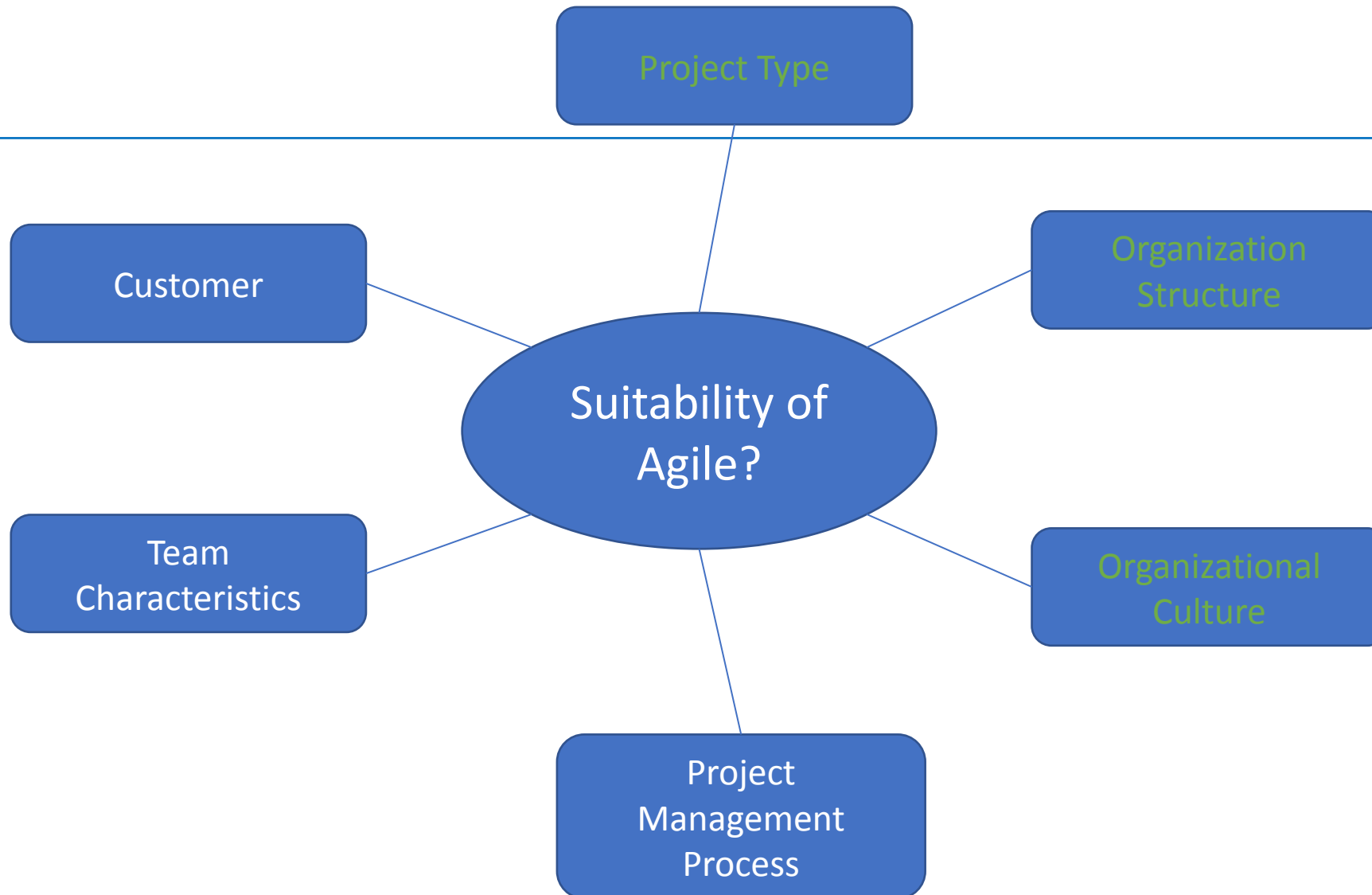
-
- agile not suitable for certain kind of projects
 - hybrid approach might be the way
 - how to determine if agile is suitable?
 - Project Type
 - high level of internal un-certainty
 - highly creative work
 - well defined outcome (agile not suitable)
 - Before moving to agile, consider
 - criticality of the project – if impact of failure is very high, consider hybrid approach
 - safety and security are primary requirements – not suitable



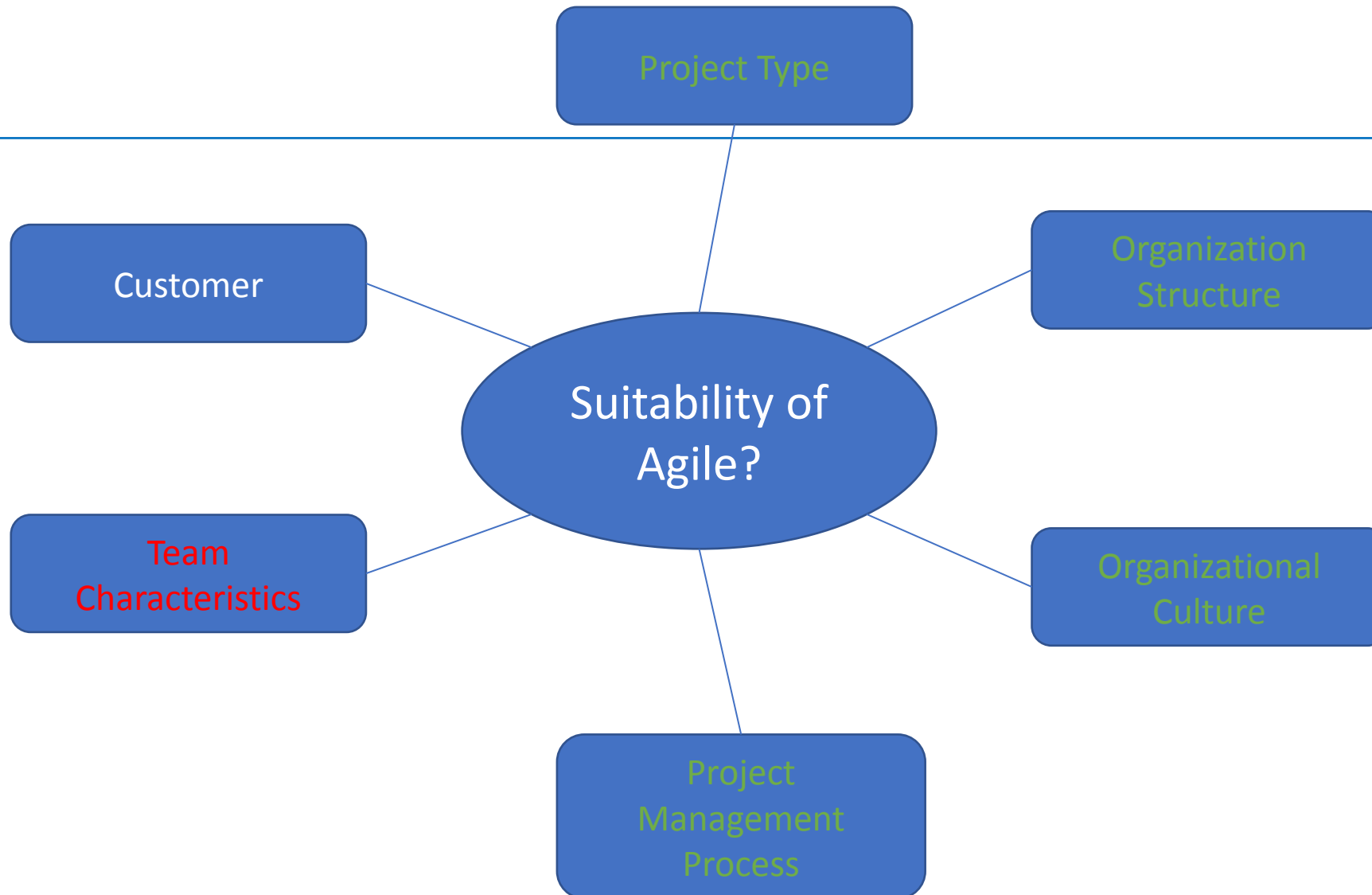
-
- Organizational structure
 - how much an org structure support collaboration
 - **org structures**
 - hierarchical
 - restricts open communication
 - employees follow process and are managed
 - co-operative
 - work in teams
 - co-op / collaboration encouraged
 - hybrid
 - begin with co-op and grow and then be hierarchical – establish balance b/w coop and hierarchical
 - Project based
 - goals of projects aligned to business model
 - **best suited for agile**



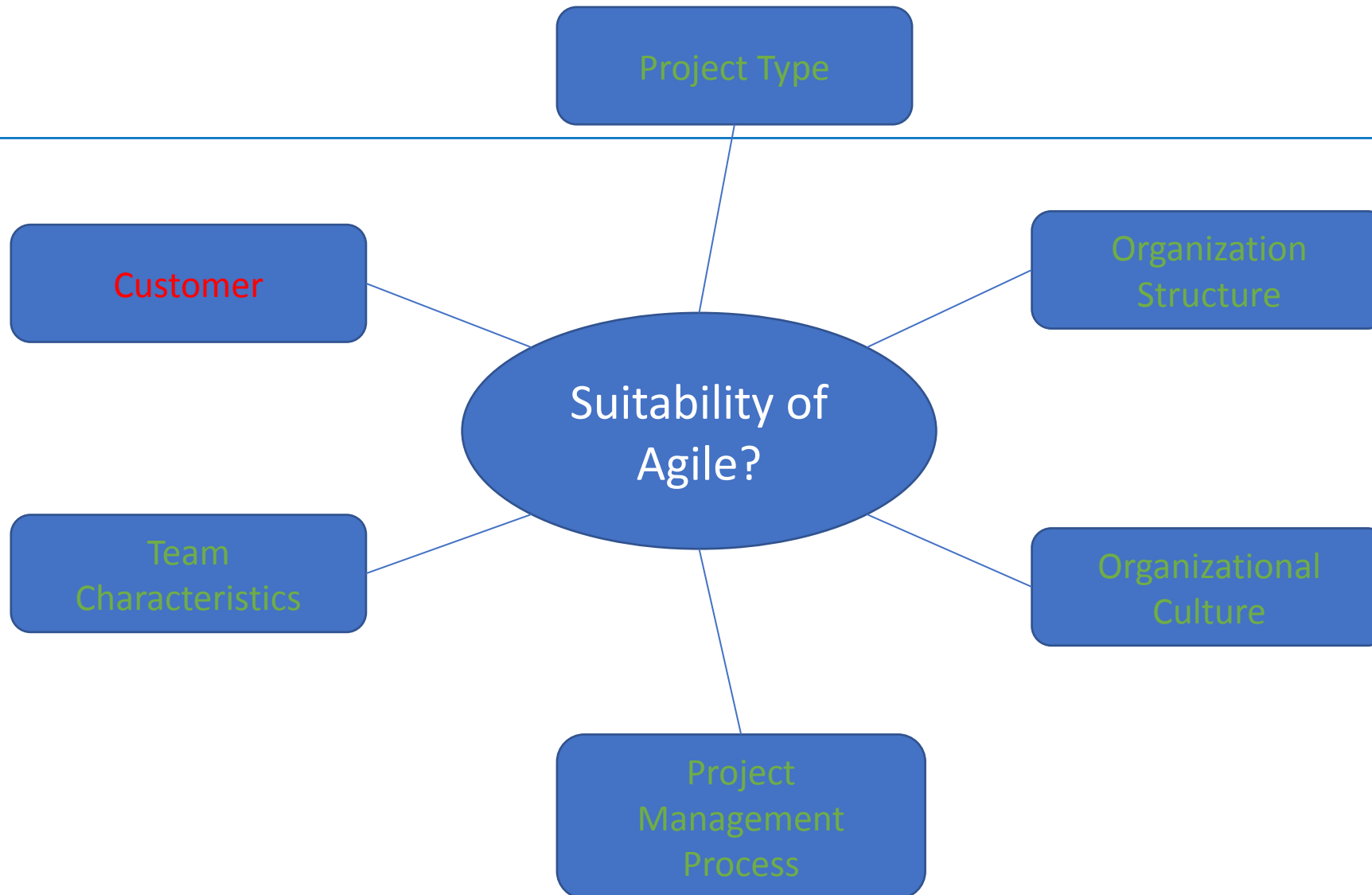
-
- Organizational culture
 - trust
 - openness
 - responsibility
 - adaptability



- Project Management process
 - more flexible and informal – easier is the transformation to agile
 - less emphasis on predefined criteria
 - Dsdm(dynamic system development method) provides higher level of formality, may facilitate transition
 - requirement management is flexible and informal (telephone call can be used to gather enough information and start project)
 - configuration management
 - release and build configurations are well defined
 - configs well defined, but process flexible



-
- Team characteristics
 - skill and experience
 - strong skills & in different functional areas
 - interpersonal skills for collaboration
 - team size
 - most suitable for small teams of upto 15 members
 - divide large teams and determine how to cooperate
 - team members location
 - close on-going collaboration
 - co-located teams considered ideal



- Customers

- must be willing to participate in development
- must be available to participate
- direct interaction between team and customer
- customer should be on-site
- activities of customer
 - vision of product
 - prioritizing
 - reviewing
 - help make decisions

Exercise

Discuss and Decide the suitability of agile process for the following projects

- mobile application for a bank
- ISRO invites bids for building fuel flow control software for next generation pslv
- ecommerce business application – like amazon, snapdeal
- university administration software
- robotic arms embedded software for automobile manufacturing unit

Teams:

- make team of two each
- discuss and present