Game Theory

- Trust in agile
- Trust is an action of believing

Software is an intangible product – implies the developers can't actually present the exact progress of the work. Sometimes this may lead into misunderstandings among team members which also leads to ill relationships in the team.

Game Theory (increase in trust -> enhances the cooperation among team members)

1. This theory depends on the way an individual takes a decision where such decisions are interdependent.
2. Also uses mathematical, economical, social and behavioural sciences.

Purpose of game – Winning or Loosing

The purpose of the game is served by the action to increase the profit (as in score)

For Instance: No.of kills in PUBG , score in mario etc.

The course of action performed by the player and by the other players leads to the fulfilment of the missions.

What is Prisoner's dilemma?

- It is a game theory framework which illustrates how lack of trust leads people to compete with one another even in a situation where they might gain more profit by team cooperation.
- Eg: Chakde India (where Shah Rukh leads a women hockey team and figures out that there's no trust among the team and tries to teach them what it means to build a trustworthy team and how one can achieve success being in a team)

Acc to prisoner's dilemma –Although people gain more by the action of cooperation, they tend to compete instead of cooperating. The main dilemma arrives when the behaviour of the person/partner being an unknown factor.

For eg : Consider 2 employees A & B in a team where A doesn't know about B's behaviour and intentions so he don't trust the actions of B and vice versa . This leads to competition among both the parties which is quite common for all human beings in some situations. Here, the competition overpowers the cooperation.

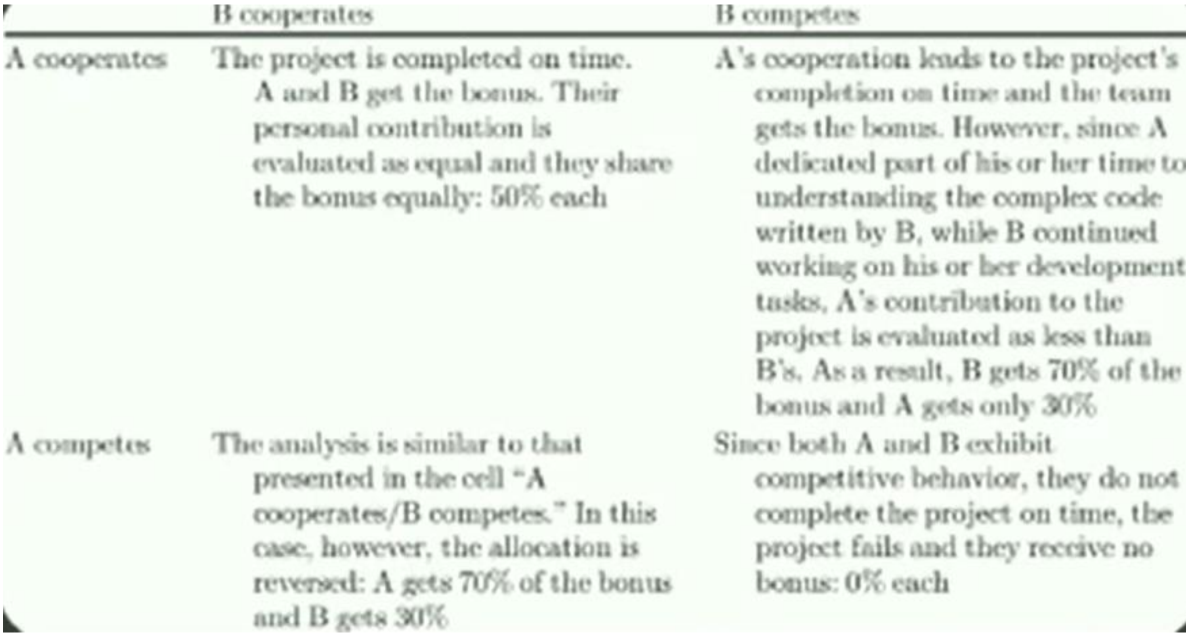
Analysing software environment using prisoner's dilemma:

1. Cooperation can be expressed in many ways such as information sharing, using coding standards, writing clear and simple code etc. (The opposite of this is considered as competition)

2. The chances of succeeding the project increases with expression of cooperation.

Case Study: Consider there's a software team with A&B and it's promised that after completion of the entire project on time a bonus will be distributed among each other.

Prisoner's dilemma table: (explain about all the 4 cases)



	B cooperates	B competes
A cooperates	The project is completed on time. A and B get the bonus. Their personal contribution is evaluated as equal and they share the bonus equally: 50% each	A's cooperation leads to the project's completion on time and the team gets the bonus. However, since A dedicated part of his or her time to understanding the complex code written by B, while B continued working on his or her development tasks, A's contribution to the project is evaluated as less than B's. As a result, B gets 70% of the bonus and A gets only 30%
A competes	The analysis is similar to that presented in the cell "A cooperates/B competes." In this case, however, the allocation is reversed: A gets 70% of the bonus and B gets 30%	Since both A and B exhibit competitive behavior, they do not complete the project on time, the project fails and they receive no bonus: 0% each

In general, software team members are always asked to cooperate, however if the development process is not clear they are unable to ensure that their cooperation might work in bringing the profits. In these cases even if there's a desire to cooperate as indicated by the prisoners dilemma, each team member will try to compete.

From game theory perspective – Agile practices like Refactoring and Test driven development leads to establishment of development environment.

TDD (cooperation): In TDD cooperation is ensured by making each and every person in the team test the code and verify that there are no active errors.

TDD (competition): In TDD, competition is the practice where all the team members doesn't verify the code and doesn't check if the code is fully tested or not!

Since agile team is continuous development environment, all the team members are supposed to apply their practices in particular test driven development (all team members are committed to verify that their code is fully tested)

Refactoring (cooperation): All the development team has to make sure that they stop the task from time to time and increase their existing code's readability and clarity.

Refactoring (competition): The members in the development team doesn’t care (pay attention) about the code readability and clarity. They don’t invest any time for refactoring.

The concern regarding to the unknown behaviour of the team members presented in the prisoners dilemma will be getting clarified due to this TDD and Refactoring practises.

The team members doesn’t face any situations of dilemma whether to extend their cooperation or not because they all cooperate and refactor the code when its needed without any doubts about how their cooperation’s gonna work and benefit the project. For the purpose of software quality and maintainability it’s(code) required to be refactored when needed. Therefore all the team members will benefit more from this practise that even if they had chose to compete with one other. This is the overall process on how to build trust and cooperation among team members.

Key Words:

Trust, behaviour, prisoner’s dilemma, cooperation, competition, team work, refactoring, TDD