❖ Using smart contracts to solve Agile methodology challenges:

The proposed solution aims to model a smart contract system to manage Agile software development projects. We intend to solve it using the following:

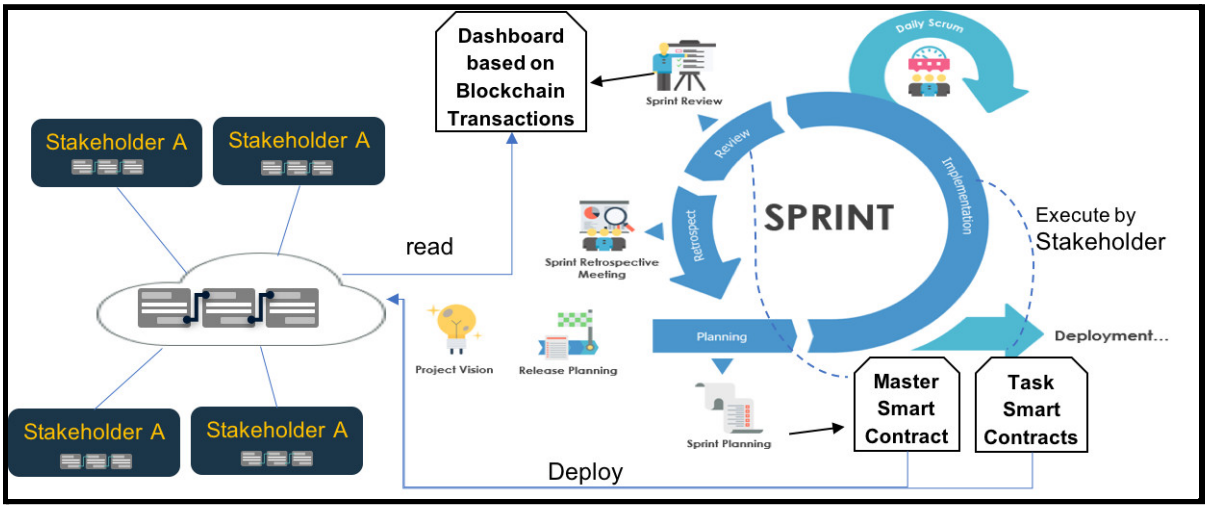
1. **Validating the User Stories:** Whenever the user stories are decided, they are stored in the master smart contract along with requisite conditions, then the stakeholders are notified, and upon their consensus and validation, the user stories are accepted for further development.

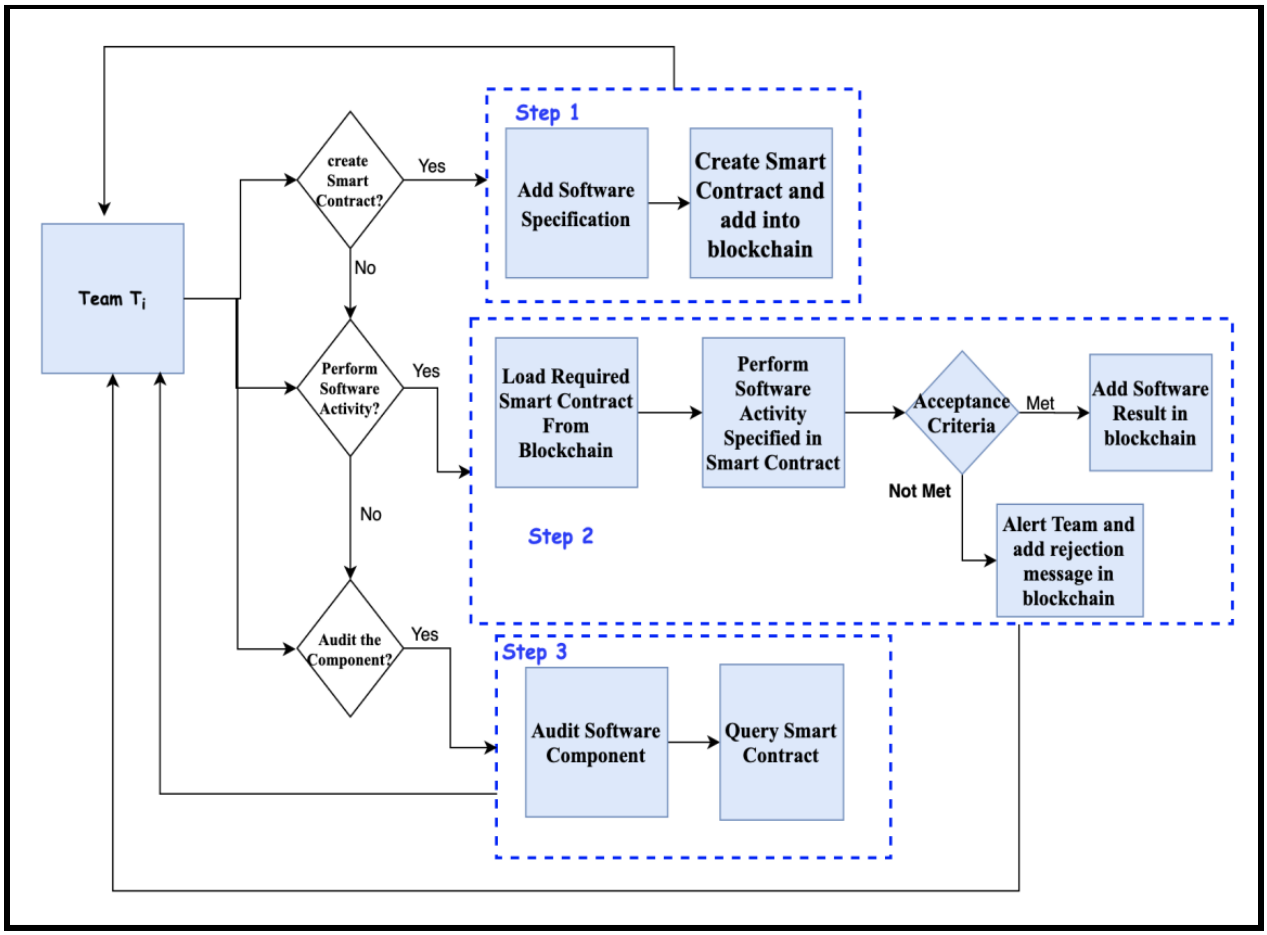
2. **Verify fulfillment of conditions:** The customer creates additional smart contracts where the product owner registers the validated user stories and specifies the acceptance test cases along with the endorsement parties and consensus protocols.
   So, once a task/user story is completely developed, its corresponding smart contract is executed to determine whether all the acceptance test cases are passed or not.
   Based on the delay in the submission of the task, the product owner gives penalty points to the developer which will later affect their performance in the organization.

3. **Accepting the specification:** Once all the tasks in the sprint are completed, the master smart contract is executed to validate all the conditions of the different stakeholders and after that, the particular software module is accepted.

4. **Payment:** Based on the validations and acceptance criteria of the software specification provided in the master smart contract, the client pays the vendor. While initializing the deal, the client and vendor agree upon a payment structure that comprises all the penalties and deductions (if requirements aren't matched or there's a delay) and this is already inputted into the smart contract.
   So, whenever a user story is completed then according to the payment structure, the payment is done directly to the vendor through ethers. This is helpful to reduce the payment delay.

❖ Model diagram:





❖ STEPS OF THE PROPOSED MODEL:

1. First, the stakeholders and the requirements gathering team will sit together to discuss the project vision and goal.
2. The requirements gathering team will then decide the requirements and form them into the user stories.
3. They will be then stored in the master smart contract.
4. This contract will be accessible to the stakeholders. They will then validate whether all their requirements are addressed in the user stories or not.
5. If all the requirements are met, then they will validate the same and approve them for the sprint planning stage.
6. If only some of the requirements are met, then they will be sent to the sprint planning stage. While the others will be given reviews by the stakeholders like, “useless”, “not meaningful”, “different target” etc and sent back to the requirement gathering team.

1. In validating and testing, the smart contract will have test conditions with different input values and expected answers in a JSON file. Only the test cases along with the tasks will be provided to the developers.
2. On completion of the development part, the developers will run the code on the given test cases and then generate the JSON file.
3. These files will be checked in the smart contract and if they give the same key then the task will be accepted.
4. As soon as the task gets accepted, the payment to the vendor is done in the form of ether.