1) BDD is about implementing an application by describing its behaviour from the perspective of its multiple stakeholders

2) BDD Story Format:

Title(one line describing the story)

Narrative: Described as a User story

In order to[benefit]: Uses a Ubiquitous language developed by the team

As a [role] : benefit is the rationale, might b testable

I want to [feature] :should fit in an sprint, otherwise break up into smaller stories

Acceptance Criteria:

Scenario: Title

Given [Context]

And [add some more contet]

When[event]

And[another event occurs]

Then[outcome]

And[another outcome]

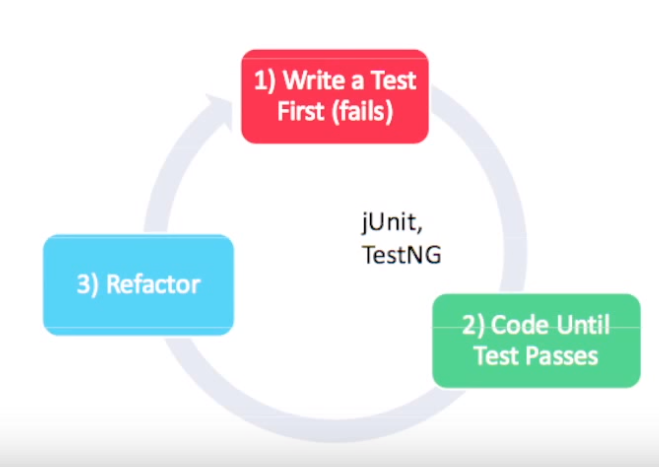
3) Pull based and push based development

pulling is more efficient than push based and enables scaling

Stories are prioritized by business value

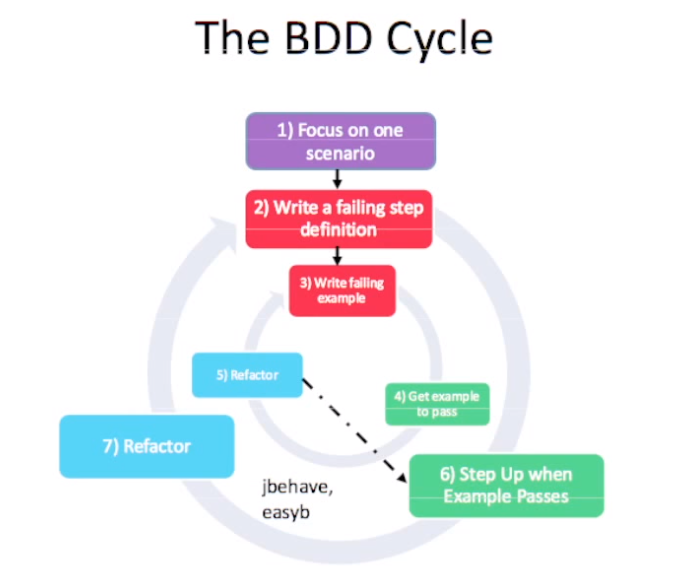
Team commits to a story delivery based on priority and capacity

Creating sub stories when required, with deep backlog more than one team can help

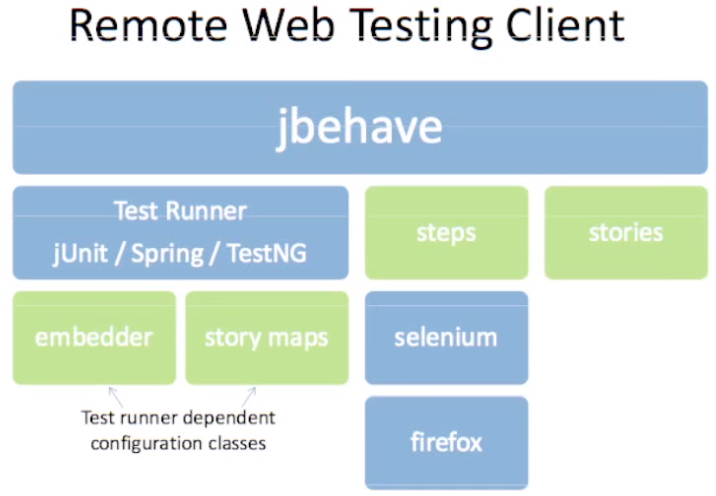
4) TDD cycle  


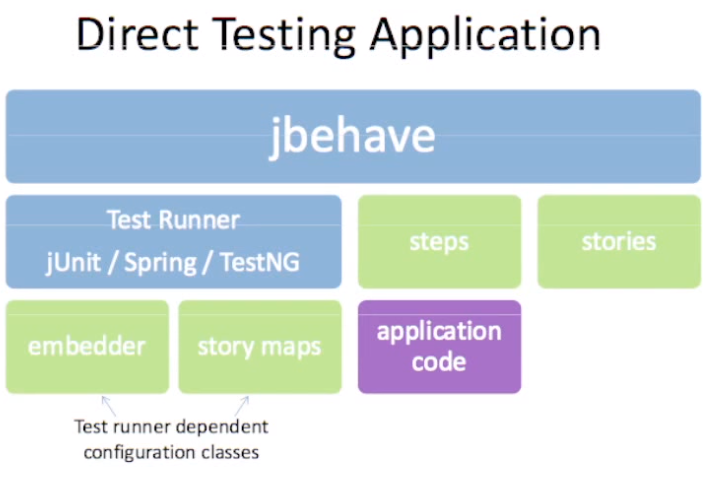
5) junit and testing-> unit testing tools  
BDD testing is not unit testing

Junit was written for developers  
Test runners assume one test per object life time

6) 

7) Jbehave -> a junit replacement for bdd testing  
 Automates story testing  
 -parses stories written in text, odt and google docs  
 -executes stories synchronously and asynchronously  
 -enables outside –in development  
Integrates with ant,maven, eclipse,junit, testing

8) 

9) 

10) Jbehave Usage  
 Write a story  
 Configure Embedder and story mapper  
 Links stories, steps and report together  
 Implement Steps using BDD cycle  
 Annotations associate steps with methods  
 Run the Story  
 View generated report

11)