1. To build framework from scratch we need to have any of the following:-

* regression pack
* sanity pack
* test case documentation
* feature files

if none of the above is there go ahead and start creating sanity pack - which has the major or important functionalities of the application.

1. constraints of automation :-

* we cannot automate dynamically changing values
* pdf content validation
* bar codes
* command prompts (non-GUI)

1. decide tools to be used :-

* IDE
* eclipse
* net beans
* language - java/selenium/c#/ruby

1. unit testing framework :-

* junit - based on cucumber so junit is best - simple and easy
* testNG - emailable reports

1. Reports :-

* log4j APIs
* junit - cucumber
* self designed
* cucumber - petty reports, json reports
* testNG - email, html reports
* if we use jenkins we need not have large html reports

1. Data flow :-

* hardcode
* from data sheets (excel) - data driven
* database - using jdbc
* properties file - property files which have all the default values - uat.properties, test.properties etc
* random APIs - for randomly generating data
* cucumber - examples
* data providers API - getting data from third party but that is heavy weight.
* data should be reversible - i.e able to clear data

1. CI server / VM

* Jenkins/team city/curise control
* selenium grid

1. build tool

* maven - run CI, configure tests etc
* ANT

1. repository :-

* SVN/Git - we go with this because it has inbuilt intelij, which maven also has.

1. Cross browser testing :-

* selenium grid - configuration
* parameterize the browser at run time

1. locators :-

* where to store locators
* load locators
* properties file
* object repository properties file
* cabybara - cucumber
* page factory
* hard code

1. designing framework :-

* folder structure, class structure
* automation framework :-
* page object model
* hybrid
* key word driven
* data driven
* cucumber
* functions driver

now designing in - intelij , maven, page object model, cucumber, junit

arch type - quickStart