

Limber Agile Methodology in Software Development

Shailendra Kumar Tiranga

Software Engineering, M.Tech, Noida International University, Uttar Pradesh

Abstract- Agile Development Methodologies have ascended in the earlier decade as an elective method for managing the work and movement of programming advancement improvement gatherings, with innumerable itemizing the determination and usage of advancement systems. The authorities of these method make extensive cases with respect to the benefits of their usage. Regardless, up to this point, only two or three these cases have been attempted in the investigation composing.

Now a day, based upon the change in the requirement, software development company will not be denied that they will not accept the requirement changes by which development process has effected or detour from their assigned work but It can never be avoided that It must be completed their task without effected to other modules. There is always a question has been arising in the software development team that how to overcome this process in the very precise and organised manner such a way Other new or old changes are Implemented without effected to the others module.

Based upon my experienced and knowledge, I have design a new methodology and given a name called – “Limber Agile Methodology in software development”.

Introduction

In this module, I have divided all the software development process in the 5 stages –which is as follows-

1. Software Development Phase-I – In general we can say that when the requirement received the development process start on that day Itself.

It Is the core part of the software development process in which CRS changes Into the SRS and make the blueprint of the software.

2. Backend Software Development – It has been divided into two submodules –

a). Infra and DB(database) design-In which we design, write Infra and data base code and If Its completed then we write the Unit test cases Its.

b). **Web Services (API)**- Write the Web services and If Its complete then write the unit test case.

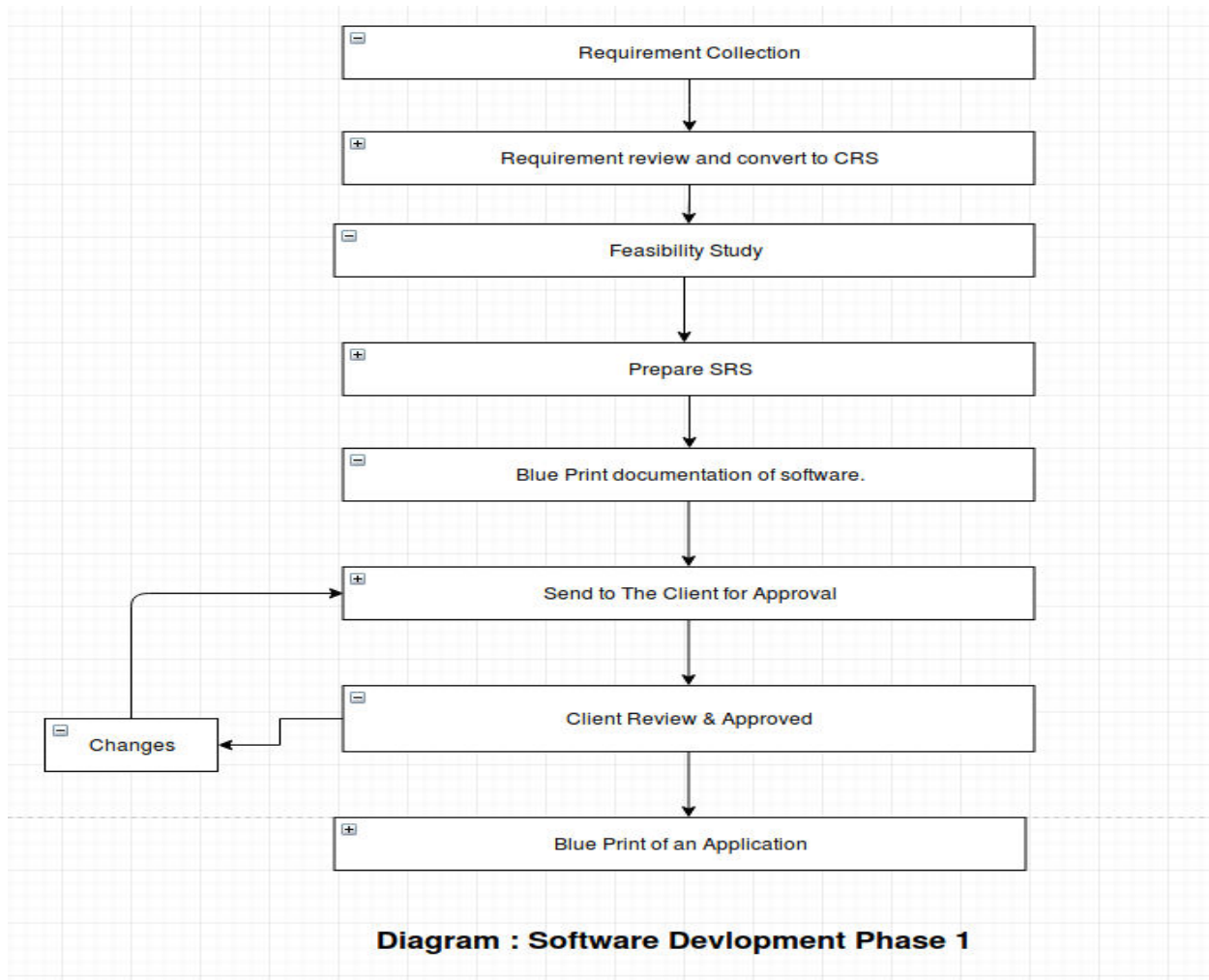
3. Infra, DB, API Testing- Writing Manual and Automation Test case, review these test cases and execute the test cases and share to their corresponding team to resolved the bugs if any found.

4. Performance Testing – Performance testing is the very crucial part of the software development because performance decide the response time and stability of the software.

we test the performance testing of the software in different ways such as – Load Testing, Stress Testing, Volume Testing, Soak Testing etc.

5. Front End Testing –Front-end Testing is testing Graphical User Interface (GUI), functionality and usability of website or application.

Methodology



1. **Requirement Collection-** It Is done by the Business Analyst and product manager, In this stage business analyst goes to the customer place and gathering the requirement.
2. **Requirement Review and convert to CRS.-** It Is done by the Business Analyst and product manager, In this stage business analyst convert requirement to CRS(customer requirement specification).
3. **Feasibility study** - A feasibility study is a study, usually done by software Architect senior engineers, that establishes whether conditions are right to implement a particular project. Feasibility studies can be done for many purposes, and are sometimes done in IT in order to look at feasibility for software setup.
4. **SRS Preparation** - A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is

developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system.

5. **Blueprint documentation of the Software** -A software development blueprint is an essential part of the software development lifecycle. It is the combination of a set of integrated steps to develop robust software products. These steps take four principles into account. Agile Development, Process Automation, Analytics, and Security. It is very helpful in developing the software.
6. **Send It to client for the approval/Changes** - Before start developing the software, blueprint of the software document must be sent to the client for the approval, after approval developer start developing or coding. Its might be possible client needs some changes in this stage so we accept It and change It to all the document accordingly based up on the change in the requirement.
7. **Client Review and approved.** This is the Initial final stage of the client to reviewing the blueprint of the software once It approved then we start developing the software by referring the approved blue print of the software.
8. **Blueprint of the Software**- If any change required then refer this document for the analysis the feasibility. It's very useful If any new developer joins to the company then It's very easy to understand the requirement and the all workflow of the software.

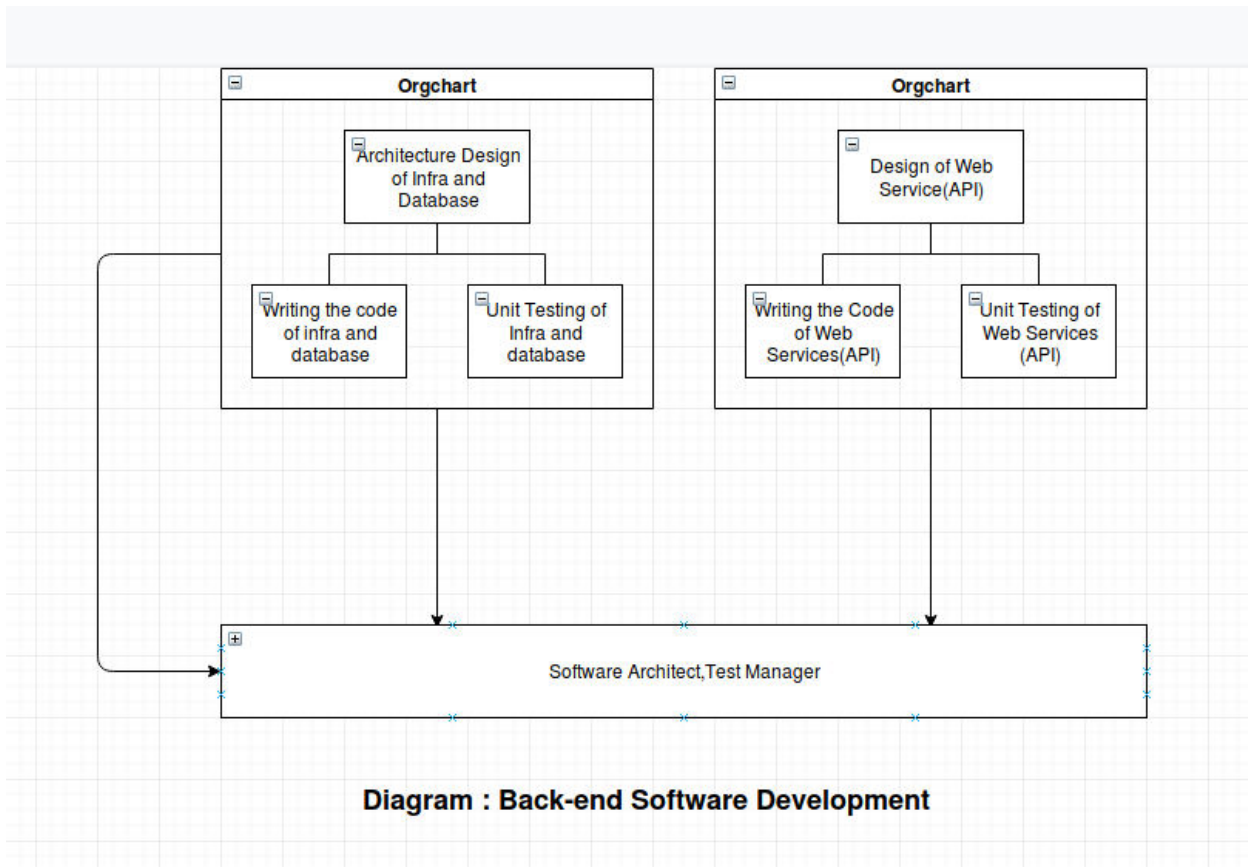
Backend software development is divided into two sub parts, which is described below-

2.1 Indra and DB design:

2.2 Design of Web Services (API):

2.1 Indra and DB design: The enabling idea of Infrastructure as Code is that the systems and devices used to run software can be treated as if they, themselves, are software. ... Quality is not a separate practice from development, a testing activity that happens after a system has been built.

The application infrastructure is a platform, which integrates diverse computers, networks using multiple operating systems and software packages.



2.Back-end Software Development

2.1.1 Writing the code of Infra and Database -

Infrastructure as code. Infrastructure as code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.

Thus we have concluded from the above diagram and description that infra and DB design is nothing but the high level writing the code by the Software Architecture and Database architecture by keeping all feasibilities of present and future changes applicable also.

It can never be avoided that architect design the workflow of network and DB by sub module level so that any sub module can be implemented or can be remove based upon the requirement change in the future.

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database

2.1.2 Unit Testing of Infra and Database

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

If anything changes in any module so no need to test It manually we can run our test suit of infra and DB and find the error or where change is required then we can change that particular module no need to test entire module of infra and the DB.

It is also a point to note that if Unit Test suit is ready then we can run it again and again to find out the error cause and implementation approach .

It's very helpful for implementation any new modules or remove any existing module to find out how many test case is falling either we remove the existing feature or implement and new feature .

2.1.3 Design of Web Services (API) -

A web service is a software system designed to support interoperable machine-to-machine interaction over a network.

For example, a client invokes a web service by sending an XML message, then waits for a corresponding XML response.

Design of the web service should be such a way that

2.2 Writing the Code of Web Services(API)-

API developer(s) writing the code of API such a way that its full fill all the requirement and writing of API code should be -

- **Easy to learn**
- **Easy to use, even without documentation**
- **Easy to read and maintain code that uses it**
- **Sufficiently powerful to satisfy requirements**
- **Easy to extend**

2.2.1 Unit Testing of Web Services (API)

Unit Testing of API is written by API developer to verify the functionality of API is working or not as per requirement .

They written the test case such a way that -

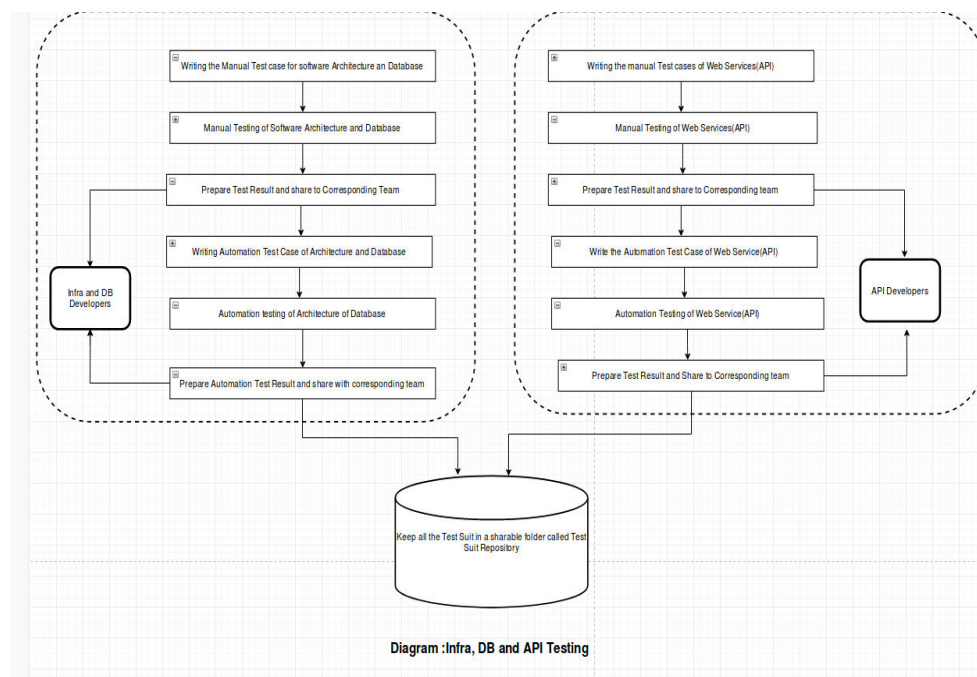
Test cases should be grouped by test category.

1. On top of each test, you should include the declarations of the APIs being called.
2. Parameters selection should be explicitly mentioned in the test case itself.
3. Prioritize API function calls so that it will be easy for testers to test.

Test Cases for API Testing:

- Return value based on input condition: it is relatively easy to test, as input can be defined and results can be authenticated
- Does not return anything: When there is no return value, a behavior of API on the system to be checked
- Trigger some other API/event/interrupt: If an output of an API triggers some event or interrupt, then those events and interrupt listeners should be tracked
- Update data structure: Updating data structure will have some outcome or effect on the system, and that should be authenticated
- Modify certain resources: If API call modifies some resources then it should be validated by accessing respective resources

It is also a point to note that all the database infra and Web Services(API) should be shared to the software architect and Sr. Application Manager to verify these. All these files must be kept in a shareable repository so that if any changes are required then they can be changed easily or it would be easy to monitor the modification.



3. Infra, BD and API Testing

3.1 Writing the Manual Test cases of Software Architecture and database

In this stage, sr manual tester write the test case of Software Architecture and database by cover all the possible test positive and negative test case.

After writing the test case they review it Internally or externally based upon the company policy and once review process completed then It share to the Test manager to review once at a glance and once It reviewed then Its kept in a sharable folder called Manual Test case Repository.

3.2 Manual Test Execution of Software Architecture and database.

In this stage Manual tester open the manual test case and start executing the positive test case and Negative test case by putting the test data .

3.3 Prepare Manual Test Result and share to corresponding team.

Once the test execution completed then share test case report to the developer(s) to fix the bugs and test manager to analysis the test result and assigning the priority and severity of the bugs. Most of the customer want to analysis the test result of their software to determined the application is stable or not to see the number of critical bugs, majorbugs, minor bugs.

3.4 Writing the automation Test cases of Software Architecture and database

In this stage, automation tester(s) write the test case of Software Architecture and database by cover all the possible test positive and negative test case.

After writing the test case they review it Internally or externally based upon the company policy and once review process completed then It share to the Test manager to review once at a glance and once It reviewed then Its kept in a sharable folder called automation Test case Repository.

3.5 Automation Test Execution of Software Architecture and database.

In this stage Automation tester(s) open the automation test case and start executing the positive test case and Negative test case by putting the test data .

3.6 Prepare Automation Test Result and share to corresponding team.

Once the test execution completed then share automation test case report to the developer(s) to fix the bugs and test manager to analysis the test result and assigning the priority and severity of the bugs. Most of the customer want to analysis the automation test result of their software to determined the application is stable or not to see the number of critical bugs, majorbugs, minor bugs.

3.7 Keep Software Architecture and database test suit and execution report in the test Suit Repository.

In this test suit repository, we store the manual test case ,automation test case and Its execution report and based on up uses we reopen it to retest, change in the test case based upon the changes in the requirement

etc.

It is nothing but a asset of the software company .

4.1 Writing the Manual Test cases of Web Services(API).

In this stage, API tester(s) write the test case of Web Services by cover all the possible test positive and negative test case.

After writing the test case they review it Internally or externally based upon the company policy and once review process completed then It share to the Test manager to review once at a glance and once It reviewed then Its kept in a sharable folder called Web Service(API) Test case Repository.

4.2 Manual Test Execution of Web Services(API).

In this stage Manual tester(s) open the automation test case and start executing the positive test case and Negative test case by putting the test data .

4.3 Prepare Manual Test Result and share to corresponding team.

Once the test execution completed then share manual test case report to the developer(s) to fix the bugs and test manager to analysis the test result and assigning the priority and severity of the bugs. Most of the customer want to analysis the manual test result of their software to determined the application is stable or not to see the number of critical bugs, majorbugs, minor bugs.

4.4 Writing the automation Test cases of Web Services(API).

In this stage, automation tester(s) write the test case of Web Services(API) by cover all the possible test positive and negative test case.

After writing the test case they review it Internally or externally based upon the company policy and once review process completed then It share to the Test manager to review once at a glance and once It reviewed then Its kept in a sharable folder called automation Web Services(API) Test case Repository.

4.5 Automation Test Execution of Web Services(API).

In this stage Automation tester(s) open the automation test case of web services (API) and start executing the positive test case and Negative test case by putting the test data .

4.6 Prepare Automation Test Result and share to corresponding team.

Once the test execution completed then share manual test case report to the developer(s) to fix the bugs and test manager to analysis the test result and assigning the priority and severity of the bugs. Most of the customer want to analysis the manual test result of their software to determine the application is stable or

not to see the number of critical bugs, majorbugs, minor bugs.

4.7 Keep all test suit and execution report of web services(API) in the test Suit Repository.

In this test suit repository, we store the manual test case, automation test case and Its execution report and based on up uses we reopen it to retest, change in the test case based upon the changes in the requirement etc.

It is nothing but a asset of the software company .