

BlueMasons Online Service LLP

Technology Department-

Specifications and Standards

TABLE OF CONTENTS

|  |  |
| --- | --- |
| 1. INTRODUCTION | 3 |
| 1. SCOPE | 4 |
| 1. DEVEOPMEMT LIFE CYCLE | 9 |
| 1. FRAMEWORK ACTIVITY | 10 |
| 1. GUIDELINES | 12 |

1. **Introduction**

This document consist of the project/product development standards which need to be followed by the Technology Department. The mentioned procedure should be read thoroughly before starting the life cycle. The measures and guideline need to be followed by the individual’s including project managers and their respective teams.

1. **Scope**
2. **Specifications-**

The project will start with the detailed project plan and schedule. After that resource will be assigned to specific roles and committees, and tool used for the project development will be acquired.

The project life cycle will follow particular methodologies to cover the development in estimated cost respectively. The methodologies includes all stages start from planning till launching of the project.

1. **Planning-**

The planning of the software development will includes the following steps-

**Step 0: Select project**

**Step 1: Identify project scope and objectives**

**Step 1.1: Identify objectives and practical measures of the effectiveness in meeting those objectives-**

In this objective of the project will be elicited on the basis of the requirement from the stakeholders.

**Step 1.2**: **Establish project authority-**

It will have the information of the project team who is responsible for the development cycle including the project manager.

**Step 1.3**: **Identify all stakeholders in the project and their interest.**

It will include the stakeholder information related to the project.

**Step 1.4**: **Modify objectives in the light of stakeholder analysis**.

Adding or deleting features to the project without affecting modules.

**Step 1.5**: **Establish method of communication**

This will involve setting up the Git repositories and updating every team members on the server and creating chat rooms for the development life cycle.

**Step 2: Identify project infrastructure**

**Step 2.1**: **Identify relationship between the project and strategic planning-** The elicitation of need of the project

**Step 2.2**: **Identify installation standards and procedures-**

The user manual for the users should be provided. And the works done by the development team should be recorded and monitored by the project manager.

**Step 2.3: Identify project team organization-**

Team Leader will assign the roles to the team members

**Step 3: Analyse project characteristics**

**Step 3.1: Distinguish the project as either objective- product driven**

**Step 3.2**: **Analyse other project characteristics ( including quality –based ones)-** it will identify remaining characteristics of the project

**Step 3.3**: **Identify high level project risks –** identifying risks

**Step 3.4**: **Take into account user requirement concerning implementation**

**Step 3.5: Select development methodology and life cycle approach-**

Choosing the life cycle to develop.

**Step 3.6**: **Review overall resources estimates-**

Estimating the cost of the project.

**Step 4: Identify project products and activities**

**Step 4.1**: **Identify and describes project products ( or deliverables )-**

Development of the product breakdown structure.

**Step 4.2**: **Document generic product flows-** making of the product flow diagram

**Step 4.3**: **Record product instance-** count of the modules.

**Step 4.4**: **produce ideal activity network-** It describes the activities and transformation.

**Step 4.5**: **Modify the ideal to take into account need for stages and checkpoints-** setting the checkpoints

**Step 5: Estimate effort for each activity**

**Step 5.1**: **Carry out bottom-up estimates** **Effort-** staff efforts for the project

**Step 5.2: Revise plan to create controllable activities-** dividing project into sub-sub task

**Step 6: Identify activity risks**

**Step 6.1**: **Identify and quantify activity based risks-**

The risk identification part is employee dropout or changes of employee and uncertain user requirements.

**Step 6.2**: **Plan risk reduction and contingency measures where appropriate-** planning based on the risk.

**Step 6.3**: **Adjust overall plans and estimates to take account of the risks-** estimating efforts on the basis of the risk.

**Step 7: Allocate resources**

**Step 7.1**: **Identify and allocate resources –** allocation of the responsibilities to the team members.

**Step 7.2**: **Revise plans and estimates to take into account resource –constraints-** In order to avoid this we should have alternate staffs for the important phases.

**Step 8 Review / Publicize plan**

**Step 8.1**: **Review quality aspects of the project plan-**

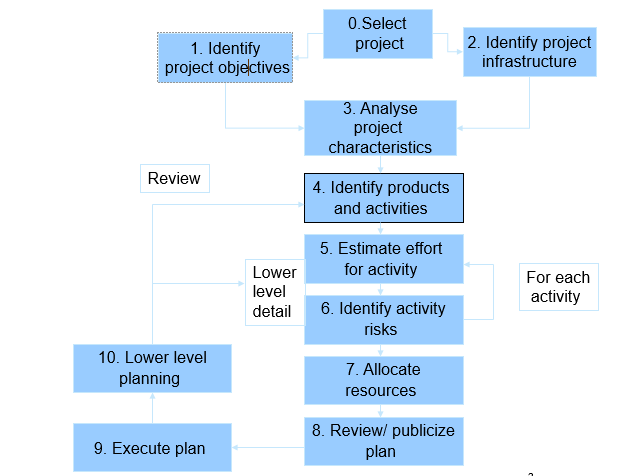
If any of the work remains uncompleted in any phase that will lead to a great loss, so we should review each and every phase works weekly or monthly.

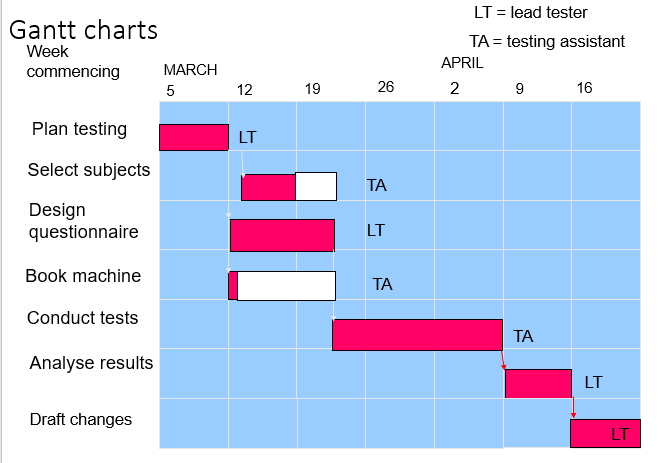
**Step 8.2: Document plans and obtain agreement-**

Proper Documentation of each and every work should be properly made and the person who is going to read this should understand clearly in this way the document should be made.

**Step 9 & 10: Execute plan / lower level of planning**

Execution of the plan





1. **Cost Estimation-**

Agreed functionality on time at the agreed cost with the required quality.

Where estimation are done?

* **Strategic planning**: Project portfolio management involves estimating the costs and benefits of new applications in order to allocate priorities.
* **Feasibility study**: This confirms that the benefits of the potential system will justify the costs.
* **System specification:** Effort needed to implement different design proposals will need to be estimated. Estimates at the design stage will also confirm that the feasibility study is still valid.
* **Evaluation of suppliers proposals**
* **Project planning**: planning and implementation of the project progress to greater levels of detail, more detailed estimates of smaller work components will be made.

1. **Development Life Cycle-**

Illustration of all the activities which should be involved in any development life cycle.

**• Software project tracking and control**: assess progress against the plan and take actions to maintain the schedule.

• **Risk management**: assesses risks that may affect the outcome and quality.

• **Software quality assurance:** defines and conduct activities to ensure quality.

• **Technical reviews**: assesses work products to uncover and remove errors before going to the next activity.

• **Measurement:** define and collects process, project, and product measures to ensure stakeholder’s needs are met.

• **Software configuration management:** manage the effects of change throughout the software process.

• **Reusability management**: defines criteria for work product reuse and establishes mechanism to achieve reusable components.

• **Work product preparation and production**: create work products such as models, documents, logs, forms and lists.

1. **Framework Activities**
2. **Communication-**

Acquiring the Business Logic from stakeholder perspective.

1. **Planning-**

Based on the requirement elicited the wireframes and prototypes will be made for further approval of the product.

1. **Modeling-**
2. **Analysis of the requirements-** understanding the requirement to implement further.
3. **Designing-**

**Stages of the Design=**

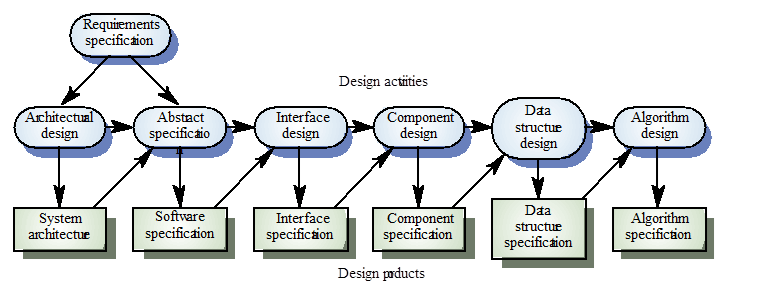
**Problem understanding**-Look at the problem from different angles to discover the   
design requirements.

**Identify one or more solutions**-Evaluate possible solutions and choose the most appropriate depending on the designer's experience and available resources.

**Describe solution abstractions**-Use graphical, formal or other descriptive notations to describe the components of the design.

**Repeat process for each identified abstraction** -until the design is expressed in primitive terms.

**Phases of the Design-**



1. **Construction**
2. **Development of the UI**
3. **Backend**
4. **Coding**
5. **Testing**
6. **Deployment**
7. **Guidelines-**
8. Updating the version of the designing and coding in the git repository regularly.
9. Maintenance of the project versioning by the project manager.
10. Integrate and build the system many times a day – every time a task is completed.
11. Work no more than 40 hours a week as a rule
12. Programmers write all code in accordance with rules emphasizing communication through the code
13. Programmers continuously write unit tests; customers write tests for features
14. Best architectures, requirements, and designs emerge from self-organizing teams
15. Daily meetings to discuss what to do next and what has been done.
16. Face-to-face conversation is the best form of communication (co-location).
17. Approvals after every stage project head.