**REDUCING SEARCH SPACE IN BIG DATA**

**PROJECT PLAN**

**PRIYA SHARMA 1MS12CS144**

**SWEETY KUMARI 1MS12CS119**

**Project Model**

We are using **SCRUM** model to carry out our project. Scrum is one of the most popular frameworks for implementing agile.

With scrum, the product is built in a series of fixed-length iterations called **sprints** that give teams a framework for shipping software on a regular cadence. **Milestones**–i.e., the end of a sprint–come frequently, bringing with them a feeling of tangible progress with each cycle that focuses and energizes everyone. (" Continuous inspiration" for the win!) Short iterations also reinforce the importance of good estimation and fast feedback from tests–both recurring struggles in waterfall projects .Scrum methodology advocates for a planning meeting at the start of the sprint, where team members figure out how many items they can commit to, and then create a sprint backlog – a list of the tasks to perform during the sprint.

During an agile Scrum sprint, the Scrum team takes a small set of features from idea to coded and tested functionality. At the end, these features are done, meaning coded, tested and integrated into the evolving product or system.

On each day of the sprint, all team members should attend a daily Scrum meeting, including the Scrum Master and the product owner. This meeting is timeboxed to no more than 15 minutes. During that time, team members share what they worked on the prior day, will work on that day, and identify any impediments to progress.

The Scrum model sees daily scrums as a way to synchronize the work of team members as they discuss the work of the sprint.

At the end of a sprint, the team conducts a sprint review during which the team demonstrates the new functionality to the PO or any other stakeholder who wishes to provide feedback that could influence the next sprint.This feedback loop within Scrum software development may result in changes to the freshly delivered functionality, but it may just as likely result in revising or adding items to the product backlog.

Another activity in Scrum project management is the sprint retrospective at the end of each sprint. The whole team participates in this meeting, including the Scrum Master and PO. The meeting is an opportunity to reflect on the sprint that has ended, and identify opportunities to improve.

**Risk Identification**

**Risk** is the potential of gaining or losing something of value. Uncertainty is a potential, unpredictable, and uncontrollable outcome; risk is a consequence of action taken in spite of uncertainty.

*Risk identification is the process of determining risks that could potentially prevent the program, enterprise, or investment from achieving its objectives. It includes documenting and communicating the concern.***.**

1. Personnel shortfalls.
2. Developing the wrong software functions.
3. Gold plating.
4. Continuing stream of requirement change.
5. Straining computer-science capabilities.

**Risk-management techniques for each**

1. **Personnel shortfalls**

– Teambuilding

– Morale building

– Cross-training

– Pre-scheduling key people

– Job matching

1. **Developing the wrong software functions**

– Organizational analysis

–Mission analysis

–Ops-concept formulation

–User surveys

–Prototyping

–Early users' manuals

1. **Gold plating**

–Requirements scrubbing  
 –Prototyping  
 –Cost-benefit analysis  
 –Design to cost

1. **Continuing stream of requirement changes**

–High change threshold

–Information hiding

–Incremental development

1. **Straining computer-science capabilities**

–Technical analysis  
 –Cost-benefit analysis  
 –Prototyping  
 –Reference checking

**Risk Matrix**

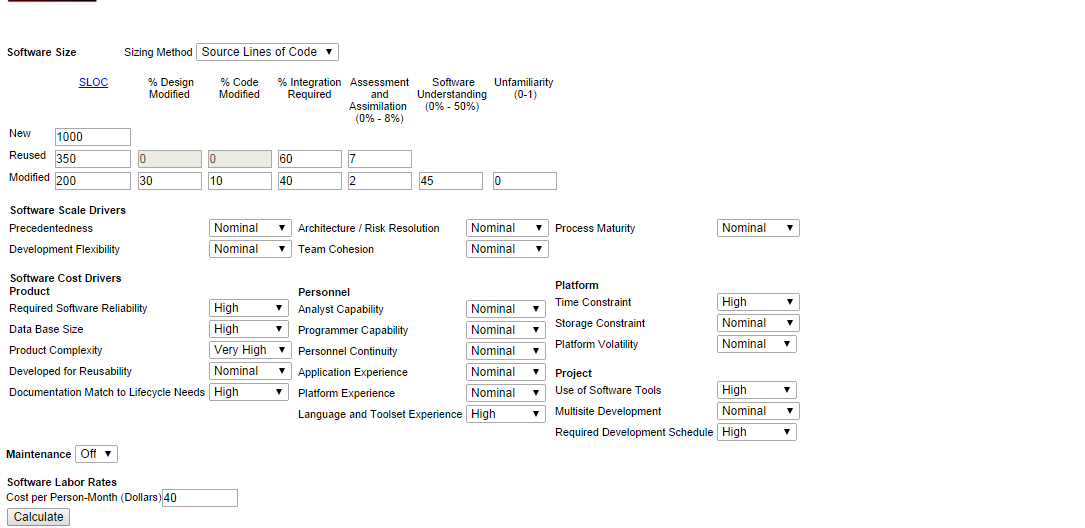
A Risk Matrix is a matrix that is used during [Risk Assessment](https://en.wikipedia.org/wiki/Risk_Assessment) to define the various levels of risk as the product of the harm probability categories and harm severity categories. This is a simple mechanism to increase visibility of risks and assist management decision making.

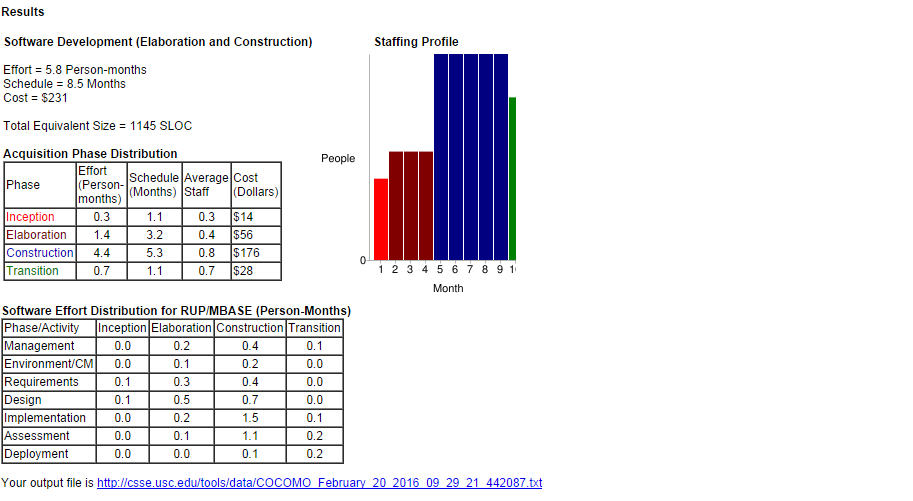
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk No.** | **RISK** | **I** | **Po (%)** | **R** | **Manage/Mitigate** |
| 1 | personnel shortfalls | S | 45% | M | Staffing with top talented, Job matching training |
| 2 | Unrealistic schedule & budget | S | 40% | M | Detailed cost & schedule estimation |
| 3 | Developing wrong softwrare functions | Mo | 15% | M | User survey,prototyping, user manuals |
| 4 | Developing wrong user interface | C | 30% | H | Prototyping & Task analysis |
| 5 | Gold plating | S | 45% | M | Requirment scrubbing,Prototyping |
| 6 | Continuing stream of requirement changes | C | 10% | M | Incremental devlopment, Information hiding |
| 7 | Shortfalls in externally furnished components | Mi | 45% | M | Benchmarking,Inspections, Comptability analysis |
| 8 | Shortalls in externally performed tasks | S | 30% | M | Teambuilding, Compititive prototyping,Reference checking |
| 9 | Real-time performance shortfalls | Mo | 20% | M | Study white paper, guidelines on performance, Modeling,Simulation |
| 10 | Straining computer-science capabalities | Mo | 15% | M | Technical analysis,Prototyping,cost-benefits analysis |
| 11 | Spoofing | S | 70% | M | Authentication ,Digital certificates |
| 12 | Unauthorized disclosure | C | 80% | H | Encryption |
| 13 | Unauthorized actions | S | 70% | M | Digital certificate ,Secured Sockey Layer(SSL) |
| 14 | Data alteration | S | 88% | M | Encryption |

**EFFORT ESTIMATION**

Software development effort estimation is the process of predicting the most realistic amount of effort (expressed in terms of person-hours or money) required to develop or maintain software based on incomplete, uncertain and noisy input.

We are using **COCOMO model** for Effort Estimation. The **Constructive Cost Model** (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics.

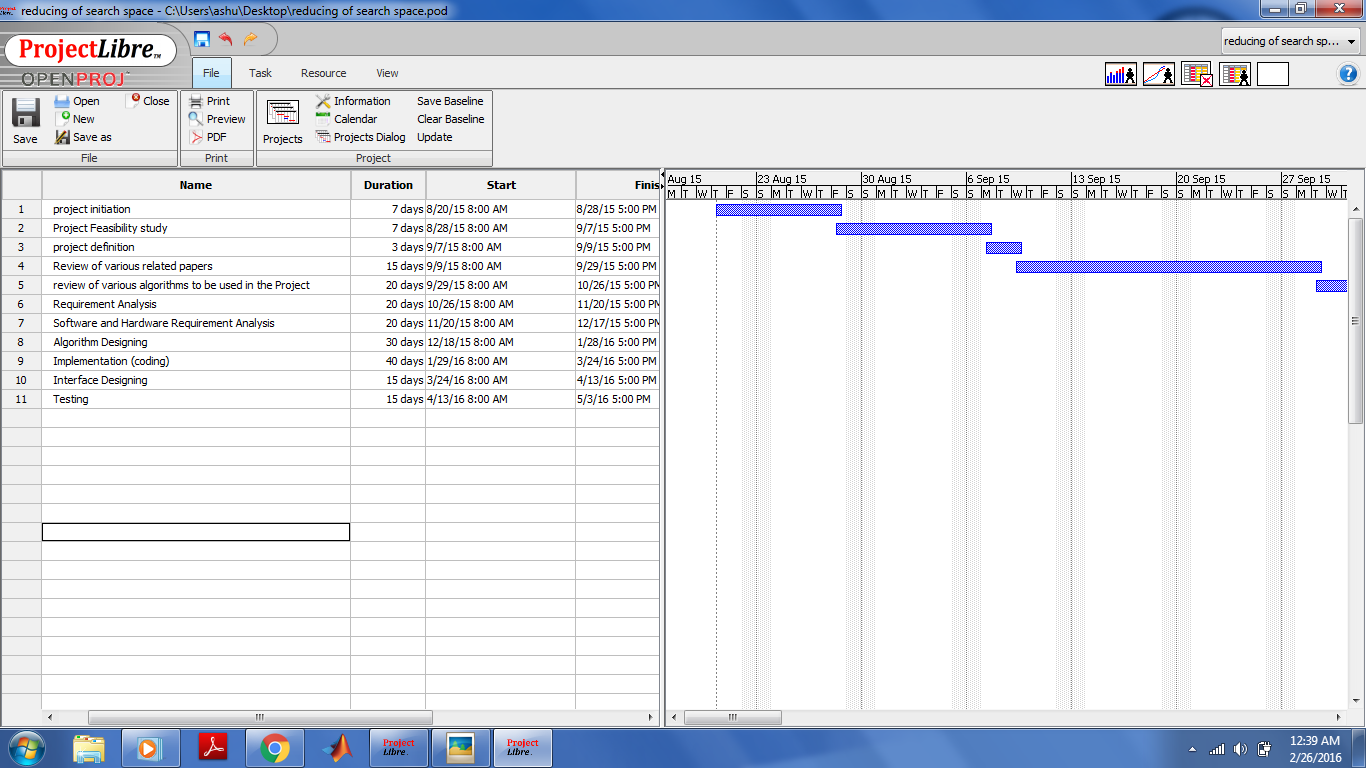


**OUTPUT:**



**GANTT CHART**

A Gantt chart is one of the most popular and useful ways of showing **activities** (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity.

****

