

2012

SEN Team #3

Team Leader - Avadhesh Gadia (200901028)
Chinmany Modi (200901039)
Arunangshu Bhakta (200901026)
Pankaj Bhambhani (200901047)
Unique Jain (200901036)
Anand Mudgerikar (200901056)
Harsh Chawda (200901052)
Mukesh Makwana (200901032)
Jay Oza (200901014)

SOFTWARE MODEL FOR ONLINE JUDGE SYSTEM

This document enlists the software model that we have used for our online judge system. It explains the model in general first and in what kind of situations is it primarily used, which then draws towards why this model fits in our project scheme.

Concurrent Development Model (CDM) is the most apt software development model for the software project undertaken by our group.

Explanation of the model

The concurrent process model can be represented schematically as a series of major technical activities, tasks, and their associated states. It makes use of state charts to represent the concurrent relationship among tasks associated within a framework of activities. It is represented schematically by a series of major technical tasks, and associated states. The user's need, management decisions and review results drive the over-all progression of the development.

The concurrent process model defines a series of events that will trigger transitions from state to state for each of software engineering activities, actions or tasks. This generates the event analysis model correction which will trigger the analysis action from done state to awaiting changes state.

The concurrent process model is applicable to all types of software development and provides an accurate picture of the current state of a project. Rather than confining software engineering activities, actions and tasks to a sequence of events, it defines a network of activities. Each activity on the network exists simultaneously with other activities, actions or tasks. Events generated at one point in the process network trigger transitions among the states.

Why the model fits in our project scheme?

The project in-hand will require to be broken down into smaller chunks. First and foremost a basic ground model should be setup and then different teams should evolve this basic model further according to the needs. While developing the project one may find the basic development to be flawed which may require the redesign and redevelopment of the basic model.

A large sized project, in order to finish the work in a short time span would need the entire project to be subdivided into smaller parts and small groups should work on these parts simultaneously.

Looking at the sizeability, scalability and the efficiency of the Concurrent development model, it seems to be the most suitable model for our project.

References

 Software Development Life Cycle, DePaul University

 Software Product Development (<http://productdevelop.blogspot.com>)

 Wikipedia (<http://www.wikipedia.org>)