# Circle Language Spec Plan Planning + Black Box Spec Part A Startup

***-***

*Author: JJ van Zon*

*Location: Oosterhout, The Netherlands*

*Date: June 22, 2009 – June 24, 2009*

*Status: Finished*

Version: XXXX-XX-XX XX

Template: Software Development Startup, Planning & Design Plan Template.doc

## Purpose of This Document

This document is a checklist for software development-related work, for a good approach and to not forget anything. It is not to be read over litterly. Later it serves as a reference for looking up how exactly something was done.

## Rough Plan

- Setup project folders **(1)**

- Time planning for super-project **(3)**

- Pick theme **(0)**

- Make goal description **(2)**

+ Other startup issies

## Orientation Plan

- Estimate how long you will be working on the orientation phase.

- Possibly communicate how long you will be working on the orientation phase.

- Do request reflection

- Determine goal

{ in document Software Development Goal Description }

- Describe goal

- After that give a short description of the means to achieve that goal.

- Look over the described goal, to see if you are not actually only describing the means to achieve the goal.

- In case of a reinitiated software development trajectory, reflect on the previous software development trajectory.

- Describe the goal concisely, but if further elaboration is necessary, put it under a sub-section in the goal description.  
Consider whether that elaboration is supposed to be part of the system documentation (so is not project-specific, but also relevant *after* the project as a description of the delivered system.).

- Also fill in the other headings in *Software Development Goal* document.

- Figure out workings of the process

> Do that as you go along

- Analyse existing material:

You have to orient yourself into the idea box and the existing documentation in order to really know the extent of the work and to really go into details about the approach.

- Idea Box

- Existing Black Box documentation

- Consider the ‘writing efficiently’ rules in the project documentation of *New Computer Language Functional Design*.

- Brainstorm about solutions

- Determine program elements  
(in the document *Software Development Elements & Time Estimation*)

Intermediate planning:

- Now you can say how much time the *research phase* is going to cost.

- You can also give a rough estimation of the time-cost of the *whole project* based on the program elements.

- Make a rough time estimation, based on the elements of the project (in the document *Software Development Elements & Time Estimation*).  
You can really only make a good estimation of the time-cost after the research stage, but the research stage may take too long to deliver a time-cost estimation only after that.

## Research Plan

- Investigate conditions that need to be met to make the project succeed

- Make test-output files for other parties

- Brainstorm about solutions

- Investigate the tools & methods to use

- Possibly make a technical design or adapt a technical design

- Possibly adjust the rough time-estimation based on this investigation.

- Possibly build something improvised

(only if it is not clear yet what elements the program has to consist of)

## Planning & Design Plan

- Make an implementation plan

- Think of approach

- Adapt list of program elements

- Make implementation plan *Reach Goal* (see document *Software Development Implementation Plan Reach Goal*)

- Make implementation plan *Finishing Touch* (see document *Software Development Implementation Plan Finishing Touch* )

- Already postpone some things indefinitely

- Possibly divide into multiple successive projects

- Make projects as small as possible, or at least: small.

- Create project subdivision for time-planning (spreadsheet)

/ Create technical design / adapt technical design

- Final time estimation