

# Last Planner

TDT4290 Customer Driven Project

Group 2

*by*

Atiyeh Seifvand

Kishore Kosuri

Eirik D. Haukedal

Aleksander L. Waage

Tobias Laupsa Nilsen

Christian Skar

Ole Kristian Ekseth



## Abstract

# Contents

<b>Abstract</b>	<b>I</b>	<b>1</b>
<b>1 Project Directive</b>		<b>1</b>
1.0.1 Chapter Overview		1
1.1 Project Purpose		1
1.2 Project Mandate		2
1.2.1 Project name		2
1.2.2 Project sponsor		2
1.2.3 Stakeholders		2
1.2.4 Client: Iterate		2
1.2.5 Project Background		2
1.2.6 Project Goal		4
1.2.7 Similar Products		5
1.2.8 External Conditions		5
1.2.9 Duration		5
<b>2 Planning</b>		<b>7</b>
2.1 Project Plan		7
2.2 Team Organization		7
2.2.1 Templates and Standards		7
2.3 Risk Management		7
2.4 Quality Assurance		7
<b>3 Preliminary Studies</b>		<b>8</b>
3.1 First Impressions		8
3.2 Scrum		8
3.3 Study of Technologies and Software		8
3.3.1 Model - View - Controller		8
3.3.2 Python		8
3.3.3 Django Framework		8
3.4 Code Conventions		8
<b>4 Requirements</b>		<b>9</b>
4.1 Product backlog		9
4.2 Non-functional Requirements		9
4.3 Use Cases		9
4.4 Test Plan		9
<b>II Sprints</b>		<b>10</b>
<b>5 Sprint 1</b>		<b>10</b>
5.1 Spring Backlog		10
5.2 Implementation		10
5.3 Sprint Evaluation		10

<b>6</b>	<b>Sprint 2</b>	<b>11</b>
6.1	Spring Backlog . . . . .	11
6.2	Implementation . . . . .	11
6.3	Testing . . . . .	11
6.4	Burndown Chart . . . . .	11
6.5	Client Feedback . . . . .	11
6.6	Sprint Evaluation . . . . .	11
<b>7</b>	<b>Sprint 3</b>	<b>12</b>
7.1	Spring Backlog . . . . .	12
7.2	Implementation . . . . .	12
7.3	Sprint Testing . . . . .	12
7.4	Testing . . . . .	12
7.5	Burndown Chart . . . . .	12
7.6	Client Feedback . . . . .	12
7.7	Sprint Evaluation . . . . .	12
<b>8</b>	<b>Sprint 4</b>	<b>13</b>
8.1	Spring Backlog . . . . .	13
8.2	Implementation . . . . .	13
8.3	Sprint Testing . . . . .	13
8.4	Testing . . . . .	13
8.5	Burndown Chart . . . . .	13
8.6	Client Feedback . . . . .	13
8.7	Sprint Evaluation . . . . .	13
<b>9</b>	<b>Sprint 5</b>	<b>14</b>
9.1	Spring Backlog . . . . .	14
9.2	Implementation . . . . .	14
9.3	Sprint Testing . . . . .	14
9.4	Testing . . . . .	14
9.5	Client Feedback . . . . .	14
9.6	Sprint Evaluation . . . . .	14
<b>III</b>	<b>Evaluation</b>	<b>15</b>
<b>10</b>	<b>Overall System Architecture</b>	<b>15</b>
10.1	Overall Architecture . . . . .	15
10.2	Further Development . . . . .	15
10.3	Limitations . . . . .	15
<b>11</b>	<b>Project Evaluation</b>	<b>15</b>
11.1	Using Scrum . . . . .	15
11.2	Group Dynamics . . . . .	15
11.3	Learning Python.....? . . . . .	15
11.4	Risks . . . . .	15
11.5	Time Estimation . . . . .	15
<b>12</b>	<b>Future Work</b>	<b>16</b>

## List of Figures

1	Planning process of Last Planner . . . . .	3
2	Look ahead process involved in Last Planner . . . . .	3
3	Pull process of Last Planner . . . . .	4
4	Pull process of Last Planner . . . . .	4
5	CCSR Process Weekly Report . . . . .	5
6	CCSR Reasons . . . . .	5

## Part I

# Planning Requirements

---

## 1 Project Directive

This section discusses about the administrative plans of the project. This includes the results of the planning phase: project mandate, project plans and various templates, tools and routines involved with the project.

### 1.0.1 Chapter Overview

The following sections can be found in this chapter:

1. Project mandate: This presents details about the customer and project.
2. Project organization: This states about our team organization and distribution of tasks and responsibilities.
3. Project plan: This part discusses on overall plan for the project, The choices we made for development and various pahses involved in the development process.
4. Templates and standards: This section describes about the various templates used while working with the project.

### 1.1 Project Purpose

The purpose of this project is to develop and deliver the Last Planner (planning the work at construction sites ) application for the Client "Iterate and Belief". The task of this report is to sum up the stages of development, creating the application, whom this report in addition describes.

## 1.2 Project Mandate

### 1.2.1 Project name

The name of the project is "Last Planner".

### 1.2.2 Project sponsor

The project sponsor is Iterate AS and Belief AS, one of Norway's leading consultancy companies.

### 1.2.3 Stakeholders

Representatives from Iterate AS and Belief AS:

1. Simen Fure Jørgensen
2. Lars Gullaksen

Group 2 in the course TDT4290 Customer-driven project:

1. Atiyeh Seifvand
2. Kishore Kosuri
3. Eirik D. Haukedal
4. Aleksander L. Waage
5. Tobias Laupsa Nilsen
6. Christian Skar
7. Ole Kristian Ekseth

Representatives from The Department of Computer and Information Science/Norwegian University of Science and Technology:

1. Gustav Aagesen
2. Ole Andreas Alsos

### 1.2.4 Client: Iterate

The client, named Iterate, is an international firm which provides software solutions for various businesses and organizations [1]. The main focus of iterate company is to find the best possible software solution fitting to needs of the customers. This is mainly achieved by becoming acquainted with the domain in which company is working on. The Lean construction is one such project in that direction.

### 1.2.5 Project Background

The consultancies Iterate and Belief are working together for developing software products for building industry. The building projects are getting benefited by following a method known as Last Planner (i.e. This is a framework for planning the building activities.)[2]. Presently, the Last Planner method is not fully developed. It is still in experimental stage. Iterate and Belief are exploring the possibilities to apply it to practical scenarios of construction thus increasing the productivity and saving lot of needless expenditures on waiting for activities to be done in the construction site.

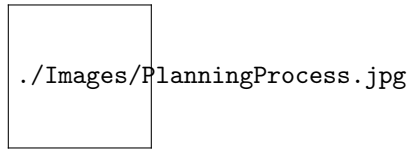


Figure 1: Planning process of Last Planner

### The Lookahead Process: Make Ready by Screening & Pulling

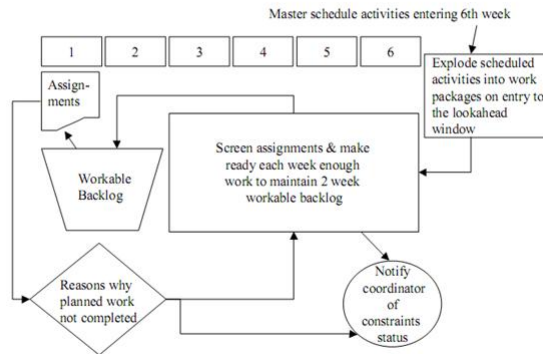


Figure 2: Look ahead process involved in Last Planner

**1.2.5.1 Last Planner** The Last Planner system was developed by Ballard[3] . The framework was initially developed for improving the quality management and productivity in construction industry. The processes involved in the construction process are to schedule tasks, make ready plans and working on the backlogs and finally weekly work plans and its execution. The following figure 1 illustrates the processes involved in the planning system.

Last Planner research focuses on improving the quality of assignments in weekly work plans (Koch Refinery Mid-Plants Project, 1993-45). The research process of last planner continued by adding a look ahead process to shape and control work flow (PARC, 19956 ; DMOS-6, 19967). This was extended further from construction to design (Nokia 8 and Hewlett-Packard9 , 1996). The look ahead process is way of assimilating the tasks for the next period for instance, 3-12 weeks. The weeks are decided upon the project characteristics and requirements such as materials, labor, and equipment etc. The figure 2 illustrates the look ahead process showing work flow from right to left. The activities for 6 weeks are looked ahead of its execution. Then they move into workable backlog, which means that all the constraints have been fulfilled and ready for execution. In other words, The Last planner pull system includes what can we do and what we should do in the planning process such that it Will produce the desired results. The figure 3 illustrates the pull mechanism involved in the last planner.



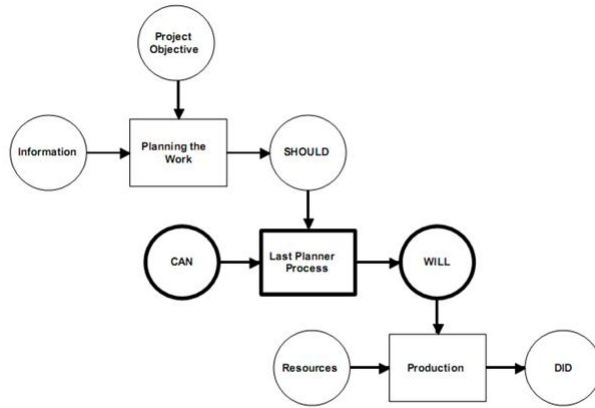


Figure 3: Pull process of Last Planner

Week	12/24/97	12/31/97	1/6/98	1/14/98	1/18/98	1/25/98	2/3/98	2/10/98	2/17/98	2/24/98	3/3/98
PPC	56%	88%	57%	87%	73%	75%	50%	53%	74%	44%	70%
Tasks Completed	5	8	8	10	11	18	7	10	23	19	14
Tasks Planned	9	7	14	15	15	24	14	19	31	43	20
Rain			1		1	3	8	2	2	1	2
Pre-Requisite			2	2	1			7	2		14
Design			1			1			4	2	8
Submittal				2	2	2					8
Other		1					1			1	4
Space										1	2
Equipment			2								2
Labor										1	1
Materials				1							1
Contract											0

Figure 4: Pull process of Last Planner

The last planner provides various possibilities to find the PPC ('percent plan complete'; i.e., percentage of assignments completed) , weekly progress The following are the graphs generated from CCSR project by implementing last planner process. The CCSR Project was to build laboratory for Stanford University for which the general contractor was Linbeck Construction.

### 1.2.6 Project Goal

(The application should provide features for planning and reporting the activities.support tasks like planning and reporting, but must be easy to use. Usability is an important aspect for this application. Based on findings in the early stages of the project, the students will further develop an existing software

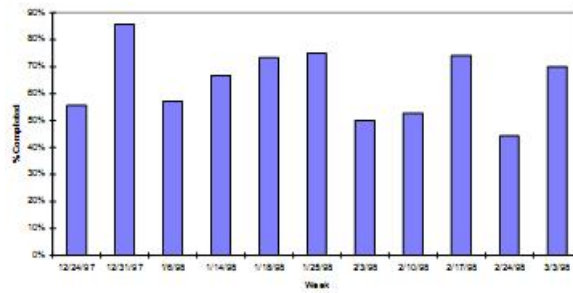


Figure 5: CCSR Process Weekly Report

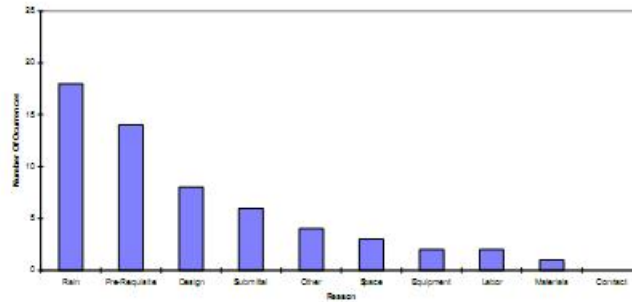


Figure 6: CCSR Reasons

prototype.)... To be done

### 1.2.7 Similar Products

Presently, the activities at the construction sites are manually maintained by recording the activities on paper. It becomes difficult for Foreman to arrange and attend the meetings with all the workers when working in the construction sites. It is rather less efficient way of recording and maintaing the activities as most of the time is wasted in waiting for others.The commonly followed procedure is Sticky notes or Papers on wall.

### 1.2.8 External Conditions

People and Companies The client: A representative from a group should regularly (i.e. Weekly or twice in a week )contact the client via email or any video conference. The meetings should focus upon updating the status of the project and taking inputs and requirements needed for the project. The supervisors: The team should meet the supervisors weekly and update the status of the report and take suggestions if needed. The group: See section xxxxx for details about the roles of the team members

Computers : The team members will use their personal laptops for the development. The code will be stored in the already existing GIT repository provided by the client. Each team member will use their personal laptop for the project. The final product will be used in the construction field. The application can be used either weekly or daily (depending upon the users who are using it)to monitor the activities at the construction site.

### 1.2.9 Duration

The work load for each member is expected to be 312 working hours. In our case, we have 8 members in our team this constitutes 2496 working hours.

- Project start: 31st of August, 2010.
- Project end and Final presentation: 25th of November, 2010.

## **2 Planning**

### **2.1 Project Plan**

### **2.2 Team Organization**

#### **2.2.1 Templates and Standards**

### **2.3 Risk Management**

### **2.4 Quality Assurance**

## 2.5 Preliminary Studies

The aim of the preliminary studies is to give an overview of the information needed before the development of the project starts. The following elements are covered by this preliminary study:

**Limitations:** The observed limitations the frame of this project, in a general term, sets.

**Scrum:** The methodology of the agile project development used.

**Study of technologies and software:** Analyzing the tools, and the constraints they imposes.

### 2.5.1 Limitations

As a software to be developed for a commercial firm, there are certain limitations set. This, in order to avoid work who is outside the scope of the project. Those limitations are:

**Response time:** Complexity of queries to the database system must be less than 0.1 seconds.

**Browsers:** Will not support backward compatibility for browsers who is not updated with the newest release. (Exception is for Microsoft Explorer, whom the web page will be working from 7 and newer).

**Handheld devices:** Handheld devices with a touch screen will not be specifically supported.

### 2.5.2 Scrum

Scrum is the methodology of the agile project development used. Initially, scrum was proposed as the preferred methodology by the customer, and, during the initial stages of this project, verified as the preferred methodology in order to reach the goals, set in the guidelines for this project.

Why is scrum the preferred methodology? The main focus is this project is usability. Does usability corresponds to a development cycle, specified by the scrum Methodology? – The scrum methodology focuses on short development cycles, giving workable program for every sprint, making the road open for a user-feed-back-responed track of development, implying that, ref figure above, every sprint adds some functionality to the system.

Daily Scrum Gaining understanding of the other team members work, and the motivating effect about making commitments to tasks, is, by the group, regarded as a key factor in a project development. Therefore, a daily scrum meeting isto be held. The three questions given to each team member is: 1. What did you do yesterday, 2. What will you do today?, 3. Are there any impediments in your way?.

Scrum Review Meeting At the end of each sprint, a presentation will be given to the customer by Skype, together with the resulting code, uploaded to their test facility.

### 2.5.3 Study of technologies and software

Analyzing the tools, and the constraints they imposes, opens the horizons, and enhances the understanding. Selecting the best matching tools for the task, reduces the risks of uncertainty, increases the productivity and opens a path through the uncertainty.

Model View Controller In the limit of requirements for the system, specified by the client, was the specification of using the pattern of a “Model View Controller”(MVC) principle. The logic behind the MVC principle is to layer the code, and thereby enable modifications on one layer, without the need of cascading changes in the other parts of the code. (Ref <http://martinfowler.com/eaDev/uiArchs.html>).

The isolation of the data, as described above, enables:

**Model:** Storage keeping of data, in order to render information in the view.

**View:** Modifications of the user interface (i.e. how the data is displayed), without a change in the functionality, or the structure of the model used, itself.

**Controller:** Modifications of the functionality (i.e. the speed of execution), without a change in the user interface, or the structure of the model used, itself.

Making a strong separation between the model and the presentation (view and controller), decreases the complexity, whom this model causes, enables rapid changes in code, due to the separation caused by the layers. Using the MVC principle, a well known, de facto, standard, eases the time-to-understand curve for future developers, reducing the cost for reusing the code for future applications.

Python The framework given to the team was written in Python, a programming language. This section includes a discussion around why the team concluded that the language would solve the task well.

Curve of learning Due to Python's semantic structure, the curve of learning is higher, compared to other languages, like PHP. This potential upside might, not necessarily, be a good thing, due to its potential of creating excessive maintenance costs in the future. Thereby, in order to avoid this pitfall, documentation would need some high light during the process of working.

The history has it that Python is not a Web language, compared, for instance, to PHP. In recent years, this has changed, thanks to the introduction of frameworks, removing the hindrances, opening the road, and thereby the ability creating complex web pages with the language Python.

The risk of an open source language languish Python is an open source language. Most open source languages has a tendency of languishing. Due to the widespread use of the language, among developers on the web, the tendency is that it will last for some years to come. Thereby, the team evaluates the risk in the language python to languish as negligible.

Library, the availability of tools Due to the widespread usage of the language, libraries, like load testing tools and web frameworks, has been developed, with the implication that a community, giving response to the question arising, exists.

Django Framework As for Python, the Django Framework was chosen by the customer as the content management system (CMS). In order to verify that Django filled the needs whom the requirements of the project builds upon, a review was made. This section includes an a discussion about why the Django environment fulfills the need of the tasks set during this project.

MVC support in the Django environment Django imposes the MVC principle on the programmer, or rather MTV framework, as defined in (ref <http://docs.djangoproject.com/en/dev/faq/general/>), thereby preventing the programmer from the pitfalls, whom normally delays a software development project. The implementation of the MVC principle, may be defined by the following stages :

**Model:** The data, whom the model represents, is defined in the “models.py” file. This is done by using the Django ORM models (FIXME: Whats this??).

**Template:** A set of HTML templates. Those templates are based on the CSS description language. As the definition in the section describing the MVC principle above states, only lightweight processing- and decision making is legal at this stage. The link between the view and the controller (below), is defined in the “urls.py” file, mapping the URLs to the view function.

**View:** Describes the presentation of the data for the user, but not how it is presented. Instead, it view describes which data is presented. Thereby, the view is a python callback function for a particular URL. Also defined as the “view function”, is the stage in responsibility of the processing, making use of the models, and methods in the models.

In addition, the Django framework provides a large and active community, includes an automatically build admin and thorough documentation.

The django framework supports the model view controller ...

Version Control Working on multiple files creates the need for a file merging- and conflict handling tool. The git tool was the chosen tool from the customers point of view, but in order to verify that it

as well fulfilled the need of this project, a discussion was made. The conclusion, comparing the svn and the git tools, ref <http://www.looble.com/git-vs-svn-which-is-better/>, of the subsections below, were that software called Git fits the need of the project, and therefore became the version control system in use.

Looking first upon the alternative system, the SVN system, it is described as a system with one central repository, thereby creating the need of connecting to the server in order to commit their changes.

Compared to SVN, git, ref <http://git-scm.com/about>, is a distributed system, implying that every group member has a local copy of the repository, using the server as a synchronization tool.

Git was verified as the best tool to use, due to the factors listed below:

**Backup:** The delay of getting a backup, if the system crashes, is avoided by keeping the same backup of the entire repository on several machines. Therby, the status of the server will not cause uncertainty during the development of the project.

**Brancing and merging:** The support Git gives of rapid branching and merging, enables multiple users working on the same files.

**Offline working:** The opportunity of working offline, without loosing the advantages of the features of having a version control tool, is provided by Git.

Text editors The text editor for the report writing chosen, was latex.

Document compiling As a methodology for producing, editing and compiling the documentation, the make utility of linux were chosen.

#### 2.5.4 Naming conventions

In order to follow hte standards, the naming conventions used by the code are those specified for the python language, ref <http://www.python.org/dev/peps/pep-0008/>.

## **3 Requirements**

**3.1 Product backlog**

**3.2 Non-functional Requirements**

**3.3 Use Cases**

**3.4 Test Plan**



## Part II

# Sprints

## 4 Sprint 1

### 4.1 Spring Backlog

### 4.2 Implementation

### 4.3 Sprint Evaluation

## 5 Sprint 2

5.1 Spring Backlog

5.2 Implementation

5.3 Testing

5.4 Burndown Chart

5.5 Client Feedback

5.6 Sprint Evaluation

## **6 Sprint 3**

**6.1 Spring Backlog**

**6.2 Implementation**

**6.3 Sprint Testing**

**6.4 Testing**

**6.5 Burndown Chart**

**6.6 Client Feedback**

**6.7 Sprint Evaluation**

## 7 Sprint 4

7.1 Spring Backlog

7.2 Implementation

7.3 Sprint Testing

7.4 Testing

7.5 Burndown Chart

7.6 Client Feedback

7.7 Sprint Evaluation

## 8 Sprint 5

8.1 Spring Backlog

8.2 Implementation

8.3 Sprint Testing

8.4 Testing

8.5 Client Feedback

8.6 Sprint Evaluation

## Part III

# Evaluation

## 9 Overall System Architecture

### 9.1 Overall Architecture

### 9.2 Further Development

### 9.3 Limitations

## 10 Project Evaluation

### 10.1 Using Scrum

### 10.2 Group Dynamics

### 10.3 Learning Python.....?

### 10.4 Risks

### 10.5 Time Estimation

## 11 Future Work

## 12 Acknowledgement



## References

- [1] <https://www.iterate.no/no>
- [2] THE LAST PLANNER SYSTEM OF PRODUCTION CONTROL by HERMAN GLENN BALLARD, May 2000
- [3] Ballard, Glenn and Koskela, Lauri (1998). On the Agenda for Design Management Research. Proceedings of the 6th Annual Conference of the International Group for Lean Construction, Guaruja Beach, Brazil, August, 1998.