Decision Tree Engine App

Software Engineering Project Teamwork

ROBOT
APPLICATION
CENTER

Group 3

Presentation overview

- Team organization and roles
- Project overview
- Project goals
- Client overview
- Planned effort
- Collaboration plan
- Supporting tools
- Requirements prioritizing
- Non-functional requirements
- Questions

Team Organization and Roles

Role	Person(s)
Team:	Nathan Chape, Eric Johansson, Niklas Sjöqvist, Valerio Lucantonio, Anna Enbom, Tamara Dancheva, Julieth Castellanos.
Project manager:	Valerio Lucantonio
Customer relations:	Anna Enbom
Github maintenance:	Tamara Dancheva
Documents’ responsible:	Nathan Chape

Project Overview

Robot Application Center developed a decision tree engine to help partners deciding if a particular investment is profitable or not. Currently they use a desktop application to generate surveys, and through this application they deploy a final-user application to answer these surveys.

Problems:

- Client want a license independent framework
- The client has to re-deploy the app each time the content changes.

Project Goals

Develop two applications:

- An admin desktop application to create general decision trees.
    - Allow admin to create/modify/delete decision trees.

- An end-user mobile app to let final users answer these trees.
    - Allow end-users to generate a summary to evaluate if their investments will be profitable.

Client overview

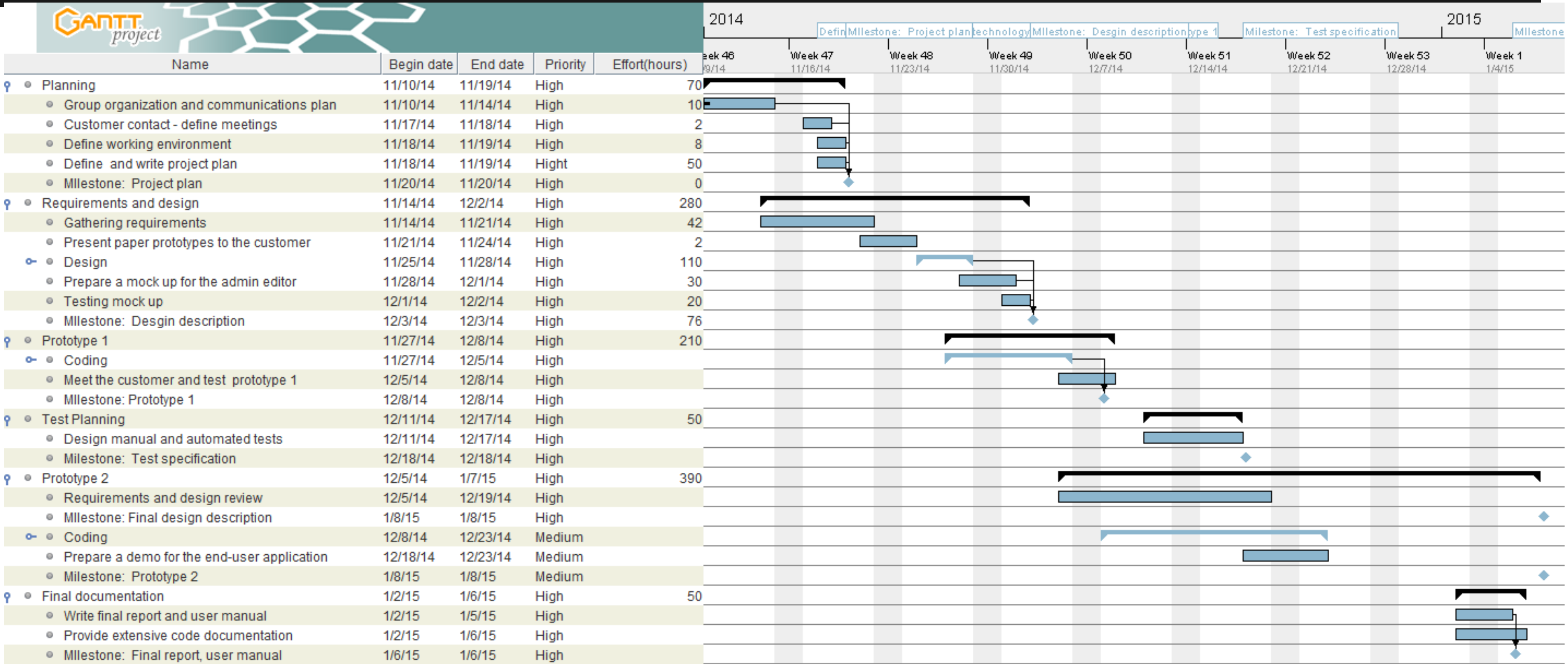
Client details:

- Andreas Runfors - Primary client (Application owner)
- Erik Hellström - Secondary client/proxy (Robotic Application Center)

Project Team/Client Interaction:

- Meetings - questions prepared beforehand
- Email

Planned effort



Name	Begin date	End date	Priority	Effort(hours)
Planning	11/10/14	11/19/14	High	70
Group organization and communications plan	11/10/14	11/14/14	High	10
Customer contact - define meetings	11/17/14	11/18/14	High	2
Define working environment	11/18/14	11/19/14	High	8
Define and write project plan	11/18/14	11/19/14	Hight	50
MIlestone: Project plan	11/20/14	11/20/14	High	0
Requirements and design	11/14/14	12/2/14	High	280
Gathering requirements	11/14/14	11/21/14	High	42
Present paper prototypes to the customer	11/21/14	11/24/14	High	2
Design	11/25/14	11/28/14	High	110
Prepare a mock up for the admin editor	11/28/14	12/1/14	High	30
Testing mock up	12/1/14	12/2/14	High	20
MIlestone: Desgin description	12/3/14	12/3/14	High	76
Prototype 1	11/27/14	12/8/14	High	210
Coding	11/27/14	12/5/14	High	
Meet the customer and test prototype 1	12/5/14	12/8/14	High	
MIlestone: Prototype 1	12/8/14	12/8/14	High	
Test Planning	12/11/14	12/17/14	High	50
Design manual and automated tests	12/11/14	12/17/14	High	
Milestone: Test specification	12/18/14	12/18/14	High	
Prototype 2	12/5/14	1/7/15	High	390
Requirements and design review	12/5/14	12/19/14	High	
MIlestone: Final design description	1/8/15	1/8/15	High	
Coding	12/8/14	12/23/14	Medium	
Prepare a demo for the end-user application	12/18/14	12/23/14	Medium	
Milestone: Prototype 2	1/8/15	1/8/15	Medium	
Final documentation	1/2/15	1/6/15	High	50
Write final report and user manual	1/2/15	1/5/15	High	
Provide extensive code documentation	1/2/15	1/6/15	High	
MIlestone: Final report, user manual	1/6/15	1/6/15	High	

Collaboration Plan

- Meetings
  - Internal (Bi-weekly) - Progress discussion, planning, etc.
  - Steering group (Weekly) - Progress report
  - Client Meetings (fortnightly) - Prototype evaluation, Requirements consolidation.

- Assorted collaborative tools.

Supporting Tools

- Gmail: communication among the team
- Asana: backlog tool
- Google Drive: sharing documents
- GitHub: code repository
- Facebook: short communications
- Dropbox: document sharing with customer

Requirements Prioritizing

Functional Requirements Prioritizing: the more “market” value a task adds to the final product the higher it will be prioritized. All tasks are prioritized in accordance with the client.

Non-Functional Requirements Prioritizing: for the admin application, prioritization will be done according to the client’s desires . For the end-users’ app prioritization will be done based on what the app market require.

Non-Functional Requirements

- **Usability:** the applications should be simple, highly intuitive and easy to understand. User-friendliness of the applications is the most important non-functional requirement.
- **Pricing:** restricted to non-proprietary software.
- **Portability:** the end-user app will be deployed for Android and IOS.
- **Documentation:** In order to assure reusability, maintainability and extensibility extensive documentation has to be provided.

Questions

Decision Tree

http://en.wikipedia.org/wiki/Decision_tree