MUDANO

RECRUITMENT STAGE 2 – ACTIVITIES

Matteo Gilardoni

January 2015

MUDANO

ENGINEER RECRUITMENT STAGE 2 INFORMATION

Many thanks for your continued interest in joining Mudano. To understand if you are a good fit for Mudano, and if we are a good fit for you we would like to work together on a test assignment. You should work with us in the style you normal would if you already worked here. The assignment is as much about testing our ability to work together as it is testing your ability to produce great code. Please take the opportunity to schedule time with us to work in your normal style.

Please complete the 2 activities we have included. These activities should be returned to us within 1 week (before 28/01/2015), if this timeline isn't feasible please let us know and we are happy to reschedule. We are of course happy to receive earlier submissions.

The activities are:

1. Question on implementing delivery paths in a web app environment
2. Practical task to develop a small absenteeism tracker component

Your recruitment contact is Sandi Chanda, if you have any questions about the process please contact him on sandi.chanda@mudano.com

ACTIVITY 1

This question is aimed at testing your ability to respond to a high level design objective with:

- The steps required to deliver from design to build
- The key technologies required
- An initial estimate of the effort to deliver
- The approach you would use to scope this and start to undertake this work

We would like you to review the information below which introduces the Mudano Delivery Path and respond covering the points above.

THE USER PROBLEM

A key area of Mudano's business in managing large scale change programmes. These projects typically last between 6 months and 2 years and will have anywhere between 10 and 100 people working on them.

It is imperative to the success of large projects that everyone in the team understands and is contributing to the delivery plan. This can be very difficult using traditional techniques. Gantt charts are too high level to accurately represent the complexities of the dependencies. Detailed Microsoft Project plans are very difficult to discuss or understand – usually meaning they are only ever understood by their author.

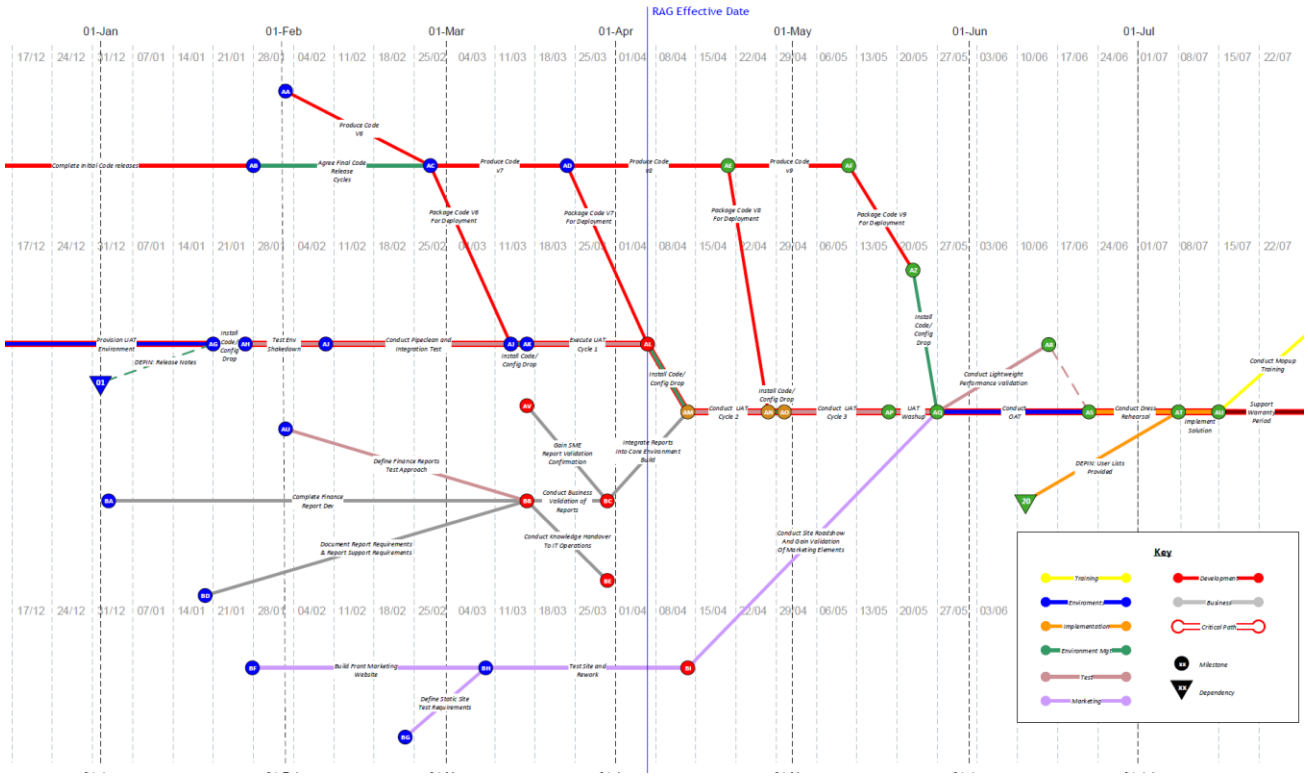
To address these issues we use a visual representation of the project plan called a Delivery Path. This representation makes it very easy to see the dependencies between tasks, their duration and current status. They are used to aid discussion, ensure a consistent understanding of the priorities and identify and address plan issues. A better understanding of the plan and the team priorities is key to enabling the team to succeed.

MUDANO DELIVERY PATH

The following diagram is an extract from an example Delivery Path.

Delivery Path description:

- Each line is an activity with a start date and end date, predecessor and successor
- Start dates are computed based on the end date of predecessor tasks
- End dates are computed based on the calculated start date + the task duration
- The colours of each activity map to a workstream (a project team) conducting that activity
- The coloured circles are Nodes where one or many activities terminate allowing dependent activities to begin
    - The Nodes are colour coded by status (Blue: Complete, Green: On Track, Amber: May Slip, Red: Slipped)
    - The Node takes the colour of the worst status of any of its predecessor tasks



OUR OBJECTIVES

- To allow the user to draw delivery path diagrams through the web interface
- Dependencies drawn in the user interface would be translated to plan dependencies used to determine project timelines and critical path
- Each element of the diagram (nodes, activities) should store data in the database and have all the features of a project plan (start dates, end dates, predecessors and successors)
- Users are able to update the status of individual milestones and this is reflected in the Delivery Path

TASK CONTEXT

We would like you to outline how you would prepare for this project. Assume you are working in the team and we asked you to investigate creating this functionality. In a maximum of 2 pages of A4 plus any supporting diagrams (or rough equivalent if using a different format) please outline how you would structure this project.

This task is designed to allow you to show your ability to take a loose subject area and conduct the initial investigation to help structure and define the work. We do not expect you to conduct a full analysis of available technology. We are looking for you to conduct the initial investigation, explain your thought process and initial findings and suggest the next steps you would take if this was a full assignment. You can structure your response however you wish. We would recommend you touch on each of the 4 points in the introduction.

ACTIVITY 2

CONTEXT

A common problem on large scale delivery projects is clashes in planned absence resulting in a number of people with similar skills being absent at the same time or having people working on critical path activities unavailable at key times. In addition, the lack of visibility of upcoming absenteeism means that plans are not always put in place to mitigate risks of the absence, often resulting in either slippages to the plan, or people being asked to cancel time away from work.

A better understanding of absenteeism would allow team members to schedule absenteeism more easily and avoid (or identify earlier) issues with may result in clashes of absenteeism with other team members.

The data from this module will also be combined with other modules such as plan management, risks and issues management, meeting actions and financial control to allow for more accurate forecasting and capacity management.

REQUIREMENT OVERVIEW

To allow planned absences to be managed better a user friendly interface allowing team members to enter their planned absence is required. The interface should allow users to view other team member's planned absence and highlight to them any clashes with other team members of any absence they enter. The entered absence, once confirmed by the team member should then be passed to the server to be logged in the database.

REQUIREMENTS

Time Categories and Units

- The user should be able to mark time as "Present", "Vacation", "Public Holiday", "Training"
- All categories other than "Present" are classified as Absent
- Weekends should be ignored by the system (not displayed and no classification should be given)
- The system should work on units of half days
- The default for days should be "Present" other than Public Holidays
- Public Holidays should be defaulted to "Public Holiday" however should be able to be changed to any of the other categories

Interface

- The user interface should be intuitive and require no user training
- The user should be able to see the records for other team members as they make their selections
- The interface should be designed to be able to deal with projects with up to 40 people
- The interface should highlight clashes with other users. A clash is defined as:
    - Absenteeism overlaps with another user
    - Absenteeism is adjacent to another user
    - Absenteeism is within 4 days of another user
- The user interface should favour graphical representation over form input
- The interface should deal with a period of up to 12 months in the future
- It is not required that all 12 months must be shown at the same time and the interface should be optimised for the most common use case (Vacation and training to be added within the next 3 months)

- The interface should be designed to reduce the required interactions (clicks, scrolls, searches, etc.) from the user to be able to
    - Select a relevant time to book absence
    - Input absence
    - Get feedback on clashes of input absence (can be automatic if required)
    - Save (can be automatic if required)

REQUIREMENTS CLARIFICATION

The requirements of this module are owed by our Head of Delivery, Ed Broussard. If you would like further clarity of requirements or have any questions Ed is happy to make time available as requested by you. Please consider this activity in the way you would if you were working as part of the team already and feel comfortable working in your normal style. Ed's email is ed.broussard@mudano.com if you wish to schedule some time or have questions.

TASK CONTEXT

This activity is designed to allow you to show your ability to understand requirements and translate them into a workable solution. As such to reduce overhead please note:

- There is no need to build the user login, you can hard code the logged in user as assume the component you build sits behind any user authentication
- You do not need to include the server side handling or database connection. A csv fie has been supplied with the database data, you can use this as input. Feel free to change its structure based on your requirements if needed
- Concentrate on the front end design and build, there is no need to build the calls back to the server once the update is made. Updating the local object and front end is sufficient. For convenience please log the object containing the user's updates that you would have sent to the server to the Console preceded by the text "EXAMPLE SERVER UPDATE REQUEST"

SUPPLIED DATA

A csv file of absenteeism is supplied for your use. The format is defined in line with the requirements above however we understand as your design develops the format may not suit your needs. Feel free to reformat as required. This is test data only so please do not let it become a restriction for you, if you wish to create your own test data this is not an issue.

The columns are

- userid: Unique identifier for the user
- name: Display name for the user
- date: Day of the record (yyyy-mm-dd)
- unit: Morning (AM) or afternoon (PM) section of the day
- value: Value for the user in that period; Public Holiday (P), Vacation (V), Training (T). There is no entry for present as this is the default value