Repository Process

Human Resources Allocation:

1-Each group will have a *GitHub leader*: he will be responsible about their master fork in the original tree.

Group1 → Alex Egan

Group2 → Mohammad Bamogaddam

Group3 → Ken S'ng Wong

2-Responsibility about the origin (the default upstream repository) will be rotated between the *GitHub leaders*. By that, they will provide backup for each others.

Setup & Creation:

*1-*At first, the *GitHub manager* (the current *GitHub leader* who is responsible about the main repository) should fork the origin from RSP repository.

2- Then, *GitHub leaders* should fork their masters from the origin (i.e. *mfb82/earth*). So, we will have: *leader1/earth, leader2/earth and leader3/earth.*

3-Each leader should give all his group members privileges to push to their group's fork.

4-Then, each group member should only clone the group's master fork to his local PC (i.e. *leader1/earth, leader2/earth and leader3/earth*).

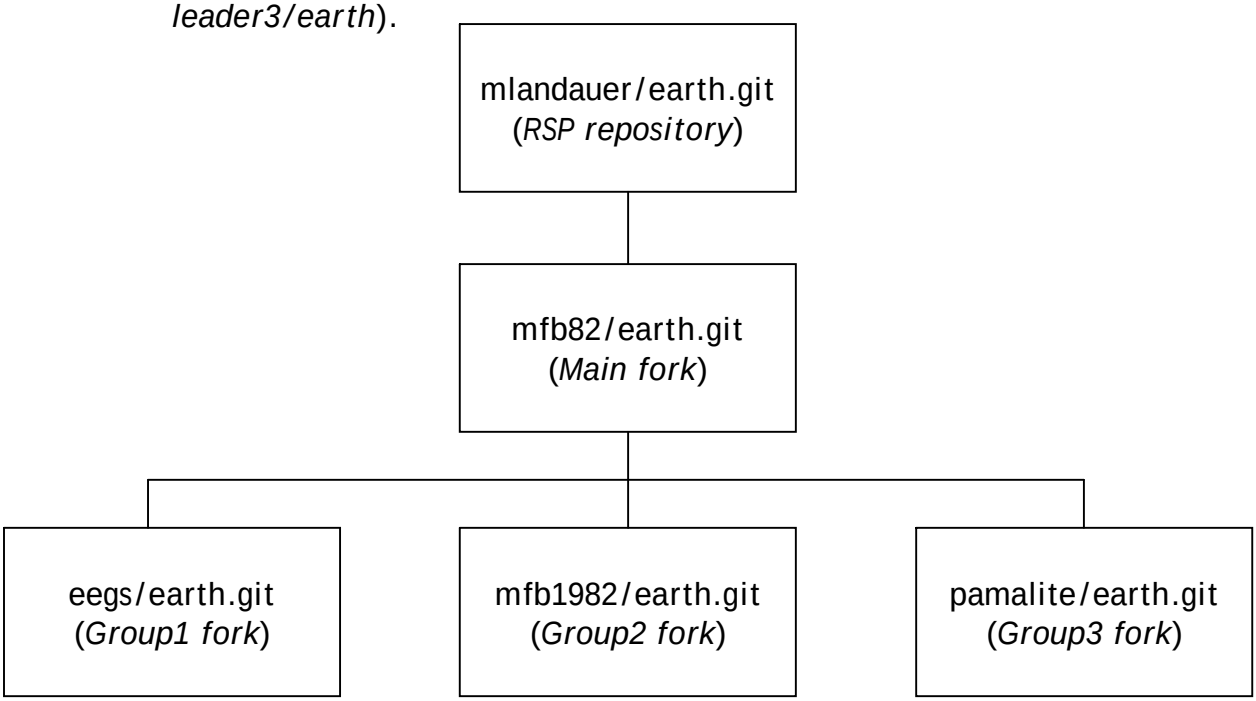

Figure 1

When a group member finishes a task:

1-He pulls the latest changes from the group's master.

2-Test his task locally.

3-Pushes the changes.

4-Sends a pull request to the group *GitHub leader and/or all other group members*. This request should include (at least):

a. Related task number or description

b. Files changed

c. Related ticket(s) numbers.

When a group GitHub leader receives a pull request from a group member:

1-The group GitHub leader reviews the changes in terms of: documentation standards, etc.

2-Performs the automated test to make sure that nothing is corrupted.

3-If the changes are approved: Sends a pull request to: all group members & to the current *GitHub manager* about the changes:

i. related task number or description

ii. files changed,

iii. related ticket numbers and

iv. Person responsible.

4-If the changes are not approved:

a. Reverts back the group's master to its status before these changes are committed.

b. Sends an email to the group member clarifying the reasons of rejecting the changes.

When the GitHub manager receives a pull request from a GitHub leader:

1-The Github manager pulls the latest changes from that group's master. (this could includes any necessary merges)

2-The GitHub manager performs an automated final test. (should be specified)

3- If it passed the test successfully, sends a pull request to all other *GitHub leaders* so that they pull the latest changes to their forks. (He could mark a version??)

4-If it does not pass the test:

a. Reverts back the origin to its state before these changes. (or from the beginning, make a fake fork or branch for testing)

b. Inform the group GitHub leader by an email about the reasons.

When a milestone has been achieved:

1-The GitHub manager marks a version/release

2-He sends a pull request to Earth people. This should contain: Milestone features.

Further Readings:

For every single developer, a good method to do the development is explained on the following tutorial:

http://www.kernel.org/pub/software/scm/git/docs/tutorial.html

Read this part: *Using git for collaboration*

Prepared by:

Mohammad Bamogaddam