# **Managing and storing our software**

Drafted by IW, 19/12/2018

Revised by KM (including EMZ notes’ of meeting) after discussion 04/03/2019.

## Needs

1. With the increased focus on software development in the MRC Unit, there is a need to formalize our software testing and standardize the process of our software development. We will predominantly be focussing on the development and dissemination of Stata programs.
2. 4 levels of storage
   1. Local: Working copy held by person currently working on it
   2. Master: CTU copy
   3. Internal release: all of CTU
   4. External release: Public copy
3. Independent testing should always happen before release (= promotion from b to c/d)
4. Testing must be documented & stored.
5. Each program should have a “lead developer” (or lead developers?) who would control its development. (Better term than “owner” though maintainer is the word used in github documentation.)
6. Version Control As ssc does not provide a utility for storing previous versions of code, it is thought that Github could be used as a way of implementing version control and providing access to previous versions, as well as the testing files and documentation of our programs.
7. Backup
8. Store communication with users (bugs, clarifications, new features)

## Provisionally agreed model: github

CTU copy held in UCL github.

Individuals can check this out (by forking a branch from the master branch on Github) to their own filespace locally ( pulling onto Git), and work on it; owner makes changes / others propose changes, the edited version is saved (committed) locally and pushed onto their Github branch. They submit a pull request to merge their updated branch to the master branch. This is reviewed by the project maintainer (possibly plus others who either rejects (giving reasons) or accepts the updates. Once accepted the owner should delete their branch as the edits and reasons for them are now saved on Github.

Multiple users can work on the same project in Github by using branches. There will be one main branch of the project, the ‘stable’ branch, which is released publically (and will contain the associated program published on ssc). The stable branch will not change until the next version of the program is ready for public release.

## Creating a new project / private to public

It was suggested that with our group’s new software, we keep the project private with one dedicated owner to accept changes to the master/stable branch. Once the project has reached a certain level of maturity and undergone testing, it can then be made public where every change/commit will be available for the wider Github community to view.

A private Github repository can be made public through the settings tab.

Further software development that needs to be private for a certain time - for example classified research or protecting from other groups publishing first - can be achieved by creating a private fork on the main directory before this is later pushed in to the master branch.

***Publishing software***

This is our main goal. Release would primarily be to SSC but could also be to CTU website and/or to public area of UCL github.

In terms of linking to Stata, the Stata program help file could include a line which points to the github directory.

## Where we are

We have started to learn how to implement this plan in Github. We have started by creating 2 real projects (Ella’s and Kevin’s projects with Andrew Copas) and one on testing admetan (with David Fisher). The initial aims were

1. Understand basic mechanics of creating repository & pushing it to github
2. Understand how to make a github pull request
3. Agree a suitable directory structure for programs, test specifications, test programs, test logs, etc.

## Next steps

1. Progress the projects described in the previous section including publishing the CRT project
2. Set-up a Github repo to store documentation on processes, agendas and meeting notes.
3. KM recommends that any researcher who writes code to incorporate version control (using git) as a matter of routine on their local workspace.
4. Action: TM offered to find out which licence the MRC would like us to use.