**Meeting #1(03/05/18)**

Core Developer Requirements:

* Proof of concepts of the below to **ensure it is a deployable product**
  + C# Kernel in Azure
  + Intellisense
  + Syntax highlighting
  + Attractive educational notebooks
* Open source Project
* Communicate through slack
* Cadence set initially to once every 2 weeks, but could become more frequent as we approach the end
* Cadence should include demo of progress over the 2-week period
* Start with icsharp as base or start from scratch on top of Roselyn
* Roselyn and Jupyter alignment
* Identify production environment to use for testing, but need to have IPs whitelisted by Dino to use

Goals for Next Meeting:

1. Get the environment setup, on everyone’s system after it has been identified
2. Identify Jupyter, Azure and other application dependencies and relationships
3. ***icsharp*** Kernel Evaluation based on a Tabular comparison of ***ipython, icsharp and ifsharp*** on aspects such as:
   1. Features
   2. Code quality
   3. Container structure – DOCKER?
   4. How is it deployed?
   5. Ifsharp has superior experiences other than the experience Jupyter offers
      1. Code completions with attractive UI
      2. Inline error messages
      3. Squiggly line highlighting
4. Our Recommendations of how to extend C# capabilities and feature
5. Thoughts on concept of ‘NorthStar’ notebook that showcases the C# kernel to users:
   1. Prepare educational C# documentation for inexperienced C# programmers
      1. Use yellow book (provided by Microsoft) for guidance
      2. Create attractive examples similar to the interactive maps of the F# notebooks
6. Market Research through distributing survey using Microsoft Forums (Lee Stott can help with distribution)
7. Define project scope and milestones
8. Define project roles and responsibilities

Potential Pitfalls:

* Don’t develop on desktop CoR and try on Linux
* Stack-handling bug that’s difficult to solve on F# kernel
* Ensure to test on MONO (development environment)
* **WINDOWS:** Run MONO or DOCKA container , **MAC/LINUX:** Run MONO