## Requirement input

1. Gui interface for interaction.
2. resources input.
3. Resources generation.
4. Compiler.

## Feasibility study

1. KITT should be a Eclipse plugin as i don’t have any experiemence with swing or AWT programing.
2. KITT should own a way for input. E.g. code editor etc.
3. KITT should own ability for generating resources based on input.
4. KITT as a TTCN tool, it should own ability for syntax checking, compilation, running.

## High level user stories defination

1. As a designer, i want KITT own a GUI for interaction, also could be setting for opening, close, preference setting.
2. As a designer, i want KITT could handle input resource ( TTCN code, etc.).
3. As a designer, i want KITT could do outputting. Generating resources e.g. message sequence diagram, comments etc. Which based on input(Builder funalities).
4. As a designer, i want KITT could be a compiler, ownning syntax checking, compilation, running functionalities. (Builder funalities).

## Module design

1. GUI



1. eclipse-plugin provide interface that interact with Eclipse.
2. KITT is an eclipse plugin for TTCN. It provides functionalities of writing TTCN code, sytax checking and so on.
3. Code



1. KITT code should interact with eclipse, and management for task scheduling, component reuqest and response//TODO
2. Task scheduling should be a scheduler for scheduling kinds of task//TODO
3. task queue could queue tasks based on task type and priority.//TODO
4. task is a abstract for taskes.//TODO
5. user task is basic user input, e.g. coding and other basic operation except system level operations.(editor functionalities)
6. system task is a system operation, e.g. KITT setting, pereference setting.(eclipse reference view)
7. deamon task is not visiable for user, e.g. automatic syntax checking, building.(delegated to eclipse builder)
8. editor is a compment, it could receive user input(coding) etc.(editor functionalities)
9. compiler is a component for syntax checking, compilation , running request//TODO
10. system shell is a component for user setting of KITT.(other view functionalities)
11. for extension. extension should be a interface for interact with KITT core.(scalable functionalities)

## Software design

1. KITT should own a way for handling resources , e.g. ttcn source code bundle(TTCN project); create, import TTCN code. PRIORITY : 1
2. KITT should own ability for generating resources based on input (generated resource). PRIORITY : 3
3. KITT as a TTCN tool, it should own ability for syntax checking, compilation, running. PRIORITY : 4
   1. As a compiler, KITT should have input.
   2. As a compiler, KITT should own syntax defination.
   3. As a compiler, KITT should be implemented based on FSM.
   4. AS a compiler, KITT should do syntax checking based on FSM, syntax defination, input.
   5. AS a compiler, KITT should do compilation.//TODO
4. KITT should own a input interface, TTCN code editor, resources (par file) editor and so on. PRIORITY : 2
5. KITT should handle resources output, generating resources, compiling code, check syntax etc. PRIORITY : 5

## Functionalities design

### Input handling

Input definition:

* + KITT Project
  + File
    - TTCN file
    - Parameter file
      * XML
      * Par file
    - Other file
      * ASN.1 file
      * Configuration file

US definition:

1. KITT could create KITT project
2. KITT could handle KITT project ,e.g. navigation, delete, modify, configure KITT project
3. KITT could handle KITT project import, export
4. KITT could create TTCN file
5. KITT could handle TTCN file, e.g. delete, edit TTCN file etc.
6. KITT could create par file
7. KITT could handle par file, e.g. delete, edit par file etc.
8. KITT could open handle other file //TODO

### Output handling

Output definition

* Basic level outputother
  + - Dependency structure
      * Call hierarchy
      * Type hierarchy
      * Out line
      * Build path
      * Lib dependency
      * Code dependency
      * Project dependency
    - Syntax high light
    - Semantic markers
    - Problem, warning output
    - Other markers
* High level out put
  + - Compilation to target code
    - Message sequence
* Other//TODO

US definition：

//TODO

### Configuration handling

* Basic configuration:
  + - Font style
    - Font color
    - Code style
    - Clean up
    - Keys
  + High level configuration:
    - Content assistant
    - Build path
  + Others //TODO

US defination

### Other functionalities handling

* Searching
* //TODO