## GUI Summer 2020

Assignment – Self Study – 2 (SS2)

Zebo Xiong – A04907051

**Below summary is based on paper**:

* i. “*An Effort-Based Framework for Evaluating Software Usability Design*”
* ii. and paper “*Detecting Software Usability Deficiencies Through Pinpoint Analysis*”.

1. The definition of Usability, Learnability, Operability, Understandability, and Pinpoint analysis

* Usability: The IEEE standard glossary defines software usability as “the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component”
* Learnability: Learnability is the ease with which a user learns to use the software.
* Operability: is the capability of a user to use the software to accomplish a specific goal.
* Understandability: Understandability is the ability of a user to understand the capabilities of the software.
* Pinpoint analysis refers to identifying and locating software usability deficiencies and correlating these deficiencies with the UI software code.

1. How would you measure Learnability, Operability, and Understandability

* Learnability: It is possible to measure learnability by plotting either the average Time-On-Task (TOT) or the average Effort-On-Task (EOT), that is, the average effort expended by a group of subjects for a task, and then fitting the subjects’ average performance for each task in a set to a Power Law curve. When there is a tight fit to the Power Law curve, then it is possible to say that learning has occurred
* Operability: can be derived from the learning model. Base on this equation: 

Several correlated parameters derived from this equation can be used to denote operability

* Understandability: One method of evaluating understandability is to compare the average subjects’ performance on a specific set of tasks to the baseline performance, such as the designer or a person accepted as an expert.

1. How would you test Learnability, Operability, and Understandability.

* To provide interface designers and developers with a methodology to evaluate their designs as they are completed, and a usability testing technique that is applicable to both the verification and validation
* One of the first steps in constructing a usability test is to establish the usability requirements for the software under evaluation. At a minimum, clients should provide a profile for each user of the application and requirements for the “In Use” Quality characteristics and learnability. The user profile should include characteristics such as education, experience with user interfaces, skills with a rating of expertise, etc. describing the systems functionality using Unified Modeling Language (UML) use cases provides a focus for both

specifying requirements and evaluating the software.

* After establishing requirements for each use case, the next step is to design a set of goals or tasks to evaluate a specific use case. The current method for constructing a usability tests concentrates on real world situations and uses them as the basis for designing tasks. In light of the experience gained from developing this framework, two more components are required from a test suite: 1) It has to contain tasks that allow the subject to master the use of the system before making measurements of usability. 2) It has to enable a software engineer to diagnose issues.
* For this end, an approach that uses a set of test cases or tasks from a scenario based test design technique utilizing a use case diagram is adopted. It provides the developer with a known focus, so that issues identified in the test trace to a specific set of interface components. Designing tasks based on use cases also insures complete coverage of the application.

1. How would you define Learnability, Operability, Understandability.

* Learnability: is the capability of a software product to enable the user to learn how to use it. Learnability may be considered as an aspect of usability, and is of major concern in the design of complex software applications. For example, when I start using Unity, the software designed tutorial which guide me to learn the layout and buttons, etc. It is a good example.
* Operability: using the same example of Unity. Operability here means what kind of software and how complicated the software I can make with Unity. Does it support 3D? Does it support AI function? Operability mean the extend the software can reach.
* Understandability: the attributes of software that bear on the users' efforts for recognizing the logical concept and its applicability.

1. How would you perform Pinpoint analysis.

* Pinpoint Analysis requires accurate means through which an engineer can identify and pinpoint issues in the software or the interface

1. inter-pinpoint analysis deals with identifying issues with tasks performed by the users in a specific system,
2. whereas intra-pinpoint analysis refers to identifying issues within tasks in a specific system.

* For example, outlier tasks might be identified through inter-pinpoint analysis and used for intra-pinpoint analysis. This analysis can help graphical user interface (GUI) designers to make decisions about element placement on displays and determine the level of effort that is related to different widgets