HCI DEVELOPMENT PROCESS

1. Software Engineering and Software Life cycle.
   - Activities from initial concept to final phase of phasing out and replacement.
   - 2 parties to consider: Customer who use the product and designer who provides the product.
   - Waterfall-Model:
     i. Requirement specification: What the system is expected to provide.
     ii. Architectural design: How the system is expected to provide?
     iii. Detailed design: From the above high-level design, individual components are explicitly designed.
     iv. Coding and unit testing: Each individual components designed are coded and tested.
     v. Integration and testing: the above components are integrated and tested later.
     vi. Maintenance: involves correction of errors after the product release.
   - Validation and Verification: To ensure the high-level requirements of the customer are internally consistent.
   - Validation and verification proofs: Verification is concerned only within mathematical world but validation is concerned with both real-world requirements associated with corresponding mathematical world equivalence.
   - Management and contractual issues: wider aspects must be considered, like time constraints, marketability of system, availability of skilled personnel's, etc.. There should be timely agreement between customer and designer.
   - Interactive system and life cycle: In these systems, the life cycles are iterative. All the components must be designed again to complement each other.
   - It also must depend on the observational quantities and such must be used to refine the components iteratively.
   - User-centered design.

- Usability Engineering: Is a User-centered design. The actual user-system interaction must be considered.
- Usability measures: These also must be considered iteratively in software life cycle.
- Problems with Usability Engg: Early Usability metrics must be explicitly conducted for the usable systems. But this depends on user behavior during different situations which designer initially is not aware of.
- Prototyping:
  (a) Throw away: prototype is built and tested. The knowledge gained are implemented and this prototype is then discarded.
  (b) Incremental: Final product is built as a series of components.
  (c) Evolutionary: Initial prototype is not discarded, but serves as a basis for next iteration.
- Potential problems:
  (a) Time: This takes time and throwing away these seems expensive.
  (b) Planning: Most don't have experience in this field.
  (c) Non-functional features
- Techniques of prototyping:
  (a) Storyboards: snapshots of interface in interactive scene.
  (b) Functionality simulations: to demonstrate limited functionalities. These are simulated. These could be throw-away.
  (c) HyperTalk: using high-level special-purpose programming language.
- Drawbacks of iterative designs:
  (a) Real management issues.
  (b) Initial assumptions for the prototypes are usually wrong.
  (c) Must realize the reason behind the errors instead of finding the symptoms.
- Design rationale:
  (a) Design space analysis: set of questions related to major issues.

(b) Psychological design rationale: explicit consequences of design for the user.

INTERACTION PARADIGMS

- Terms of interaction: accomplish goals in a specific domain. An intention is an action to achieve this goal.
- Evaluation cycle:
  (a) Establish goals
  (b) Forming intentions
  (c) Specifying actions.
  (d) Executing these.
  (e) Perceiving the state.
  (f) Interpreting the state.
  (g) Evaluate the state.
- Interaction framework:
  (a) System
  (b) User
  (c) Input
  (d) Output
- User begins the cycle by formulation of goals and tasks to achieve these. This is by Input. This is transformed by system. System produces new state to user as output.
- Ergonomics:
  - ♦ Study of physical characteristics of the interaction (Human Behavior).
  - ♦ Arrangement: Functional (related functions and displays must be together).
  - ♦ Arrangement: sequential (must reproduce the order of interaction)
  - ♦ Arrangement: frequency (most frequent must be most accessible)
  - ♦ Physical Environment: (arrangement of machines, displays and size of the users)
  - ♦ Health Issues: (physical positions, Temperature, Lighting, Noise and time)
  - ♦ Color : They must be distinct.

- Interaction styles:
  (a) Command Line Interface: instructions to the computer directly. Difficult to remember.
  (b) Menus: Options are displayed. Hierarchically ordered.
  (c) Natural Language: Ambiguity. Restricted to domains.
  (d) Query dialog: easy to use but limited functionalities.
  (e) Form-fills and spreadsheets.: Useful for data entry and data retrieval.
  (f) WIMP: stands for windows, icons, menus and pointers. Used by Mac, Windows and Unix.
  (g) Point and click interfaces: Hypertext and web-browsers.
  (h) 3D interfaces: VR and 3D WIMP.3D workspaces (windows in front are larger)
- Understanding experience: constraints of designer
  (a) Ergonomic
  (b) Physical
  (c) Legal and safety
  (d) Aesthetic
  (e) Economic.