Prototyping
Agenda for discussion

- Define Prototyping
- Need/Importance for prototyping
- Types of prototypes
- Prototyping as methodology
- User interface prototyping
- Advantages and disadvantages of prototyping
- Lessons learnt
Prototyping Defined

**Prototyping** is the process of quickly putting together a working model (a prototype) in order to test various aspects of a design, illustrate ideas or features and gather early user feedback.- Wikipedia

**IEEE** defines **prototyping** as “A type of development in which emphasis is placed on developing prototypes early in the development process to permit early feedback and analysis in support of the development process.”
Need for prototyping

- Enables us to explore the problem space with the stakeholders.
- As a requirements artifact to *initially envision* the system.
- As a design artifact that enables us to explore the solution space of your system.
- A vehicle for you to communicate the possible UI design(s) of your system.
- A potential foundation from which to continue developing the system.
# Advantages & Disadvantages of Prototyping

<table>
<thead>
<tr>
<th><strong>Advantages</strong></th>
<th><strong>Disadvantages</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Users can try the system and provide constructive feedback during development</td>
<td>Each iteration builds on the previous iteration and further refines the solution. This makes it difficult to reject the initial solution as inappropriate and start over.</td>
</tr>
<tr>
<td>An operational prototype can be produced in weeks</td>
<td>Formal end-of-phase reviews do not occur. Thus, it is very difficult to contain the scope of the prototype.</td>
</tr>
<tr>
<td>Users become more positive about implementing the system as they see a solution emerging that will meet their needs</td>
<td>System documentation is often absent or incomplete, since the primary focus is on development of the prototype.</td>
</tr>
<tr>
<td>Prototyping enables early detection of errors</td>
<td>System backup and recovery, performance, and security issues can be overlooked.</td>
</tr>
</tbody>
</table>

Reference: [http://facpub.stjohns.edu/~wolfem](http://facpub.stjohns.edu/~wolfem)
Journey of the Prototyping process

Goals → Functionality → Evaluate → Develop
Types of prototyping

- Throw-away Prototyping
- Evolutionary Prototyping
- Low Fidelity Prototyping
- High Fidelity Prototyping
Throw Away Prototype

- Throw Away Prototype is developed from the initial requirements but is not used for the final project.

- Written specifications of the requirements

- Some developers believe that this type is a waste of time because you don’t use it.

- Regardless if prototype is discarded or kept for production, you must use an easy to use language.
## Advantages & Disadvantages

<table>
<thead>
<tr>
<th><strong>Advantages</strong></th>
<th><strong>Disadvantages</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Significantly reduce project risk</td>
<td>The prototype actually does nothing, it's just presentational.</td>
</tr>
<tr>
<td>Has a short project timeline</td>
<td>Only for a limited purpose</td>
</tr>
<tr>
<td></td>
<td>Starting become a thing of the past. Not getting used as much now.</td>
</tr>
</tbody>
</table>
Evolutionary Prototype

- Evolutionary prototyping is considered the most fundamental form of prototyping.

- Evolutionary prototyping main concept is to build a robust prototype and constantly improve it.

- Objective to deliver a working system to the end user.

- According to Steve McConnell, "evolutionary delivery is a lifecycle model that straddles the ground between evolutionary prototyping and staged delivery."
Evolutionary Delivery

Rapid Development, Taming Wild Software Schedules, by Steven McConnell, Press 1996
Evolutionary Prototyping phases

1. Define the basic requirements
2. Create a working prototype
3. Verification of the working prototype
4. Evaluate and deliberate the requirements

The process cycle flows from Define the basic requirements to Create a working prototype, then to Verification of the working prototype, and finally to Evaluate and deliberate the requirements.
Advantages

- You are always looking for new ways to improve the system.
- This model increases the chance of having the client satisfied with the working system.
- The model can be used even when the requirements are not defined.
- Quicker delivery of the system
Disadvantages

- This method can be used to avoid documenting the requirements of the system.
- Management is required
- Long term maintenance can be expensive
- Uncertain design idea’s
- Information can be lost through so many improvement changes
Low-fidelity Prototyping

- Low-fidelity prototyping is generally limited function, limited interaction prototyping effort.

- They are constructed to depict concepts, design alternatives and screen layouts. They are intended to demonstrate general look and feel of the interface.

- They are created to educate, communicate and inform, but not to train, test or serve as a basis for which to code.

- Low fidelity prototyping is used early in the design cycle to show general conceptual approaches without much investment in development.

High-Fidelity Prototyping

- High-fidelity prototypes represent the core functionality of the products user interface.

- High fidelity prototypes are fully interactive systems. Users can enter data in entry fields, respond to messages, select icon to open windows and interact with user interface as if it were a real system.

- They trade-off speed for accuracy.

- Building high fidelity prototypes consume resources and have high cost.

# Comparison of two prototyping efforts

<table>
<thead>
<tr>
<th>Type</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| **Low-fidelity prototypes** |  • Lower Development cost  
  • Evaluate multiple design concept  
  • Useful communication device  
  • Address screen layout issues  
  • Address screen layout issues  
  • Useful for identifying requirements |  • Limited error checking  
  • Facilitator driven  
  • Limited utility after requirements established  
  • Navigational and flow limitations  
  • Poor detailed specifications to code |
| **High-fidelity prototypes** |  • Complete functionality  
  • Fully interactive  
  • User Driven  
  • Clear definition of navigation  
  • Look and feel of final product  
  • Use for explorations and tests |  • More expensive to develop  
  • Time-consuming to create  
  • Not effective for requirements gathering |
Prototyping-based Methodology

Prototyping as methodology for systems development

❖ A preliminary design of the system is built using throw-away prototyping.

❖ Demonstrations to end users, as well as investigations on this prototype, allows for the design of more precise requirements as well as the evaluation of techniques to be used in the final system.

❖ Refinements on throw-away prototype concerning the requirements are done.

❖ Evolutionary prototyping used to build a model prototype (an accurate and complete description of the system). These prototypes can be studied under various simulated conditions.

❖ Refinement on model prototype, concerning the actual system, is done and final system is rolled out after testing.
User Interface Prototyping

Development of highly interactive software system with graphical user interfaces (GUI) has become increasingly common and acceptance of such a system depends highly on the quality of GUI.

Prototyping is an excellent means of generating ideas about how the GUI can be designed and it helps to evaluate quality of solution at an early stage.
Classification of user interface prototypes

- Presentation Prototypes
- Functional Prototypes
- Breadboards
- Pilot Systems
Risks in Prototyping

- Client may believe that system is real.
  - Unrealistic expectations of the progress

- Implementers make poor choice
  - Justified in prototype but not in real system
  - Tempting to build real system same way

- Prototype is not identical to the real system
  - Users may interact differently due to different response characteristics
  - Must interpret prototype experience with care
Do we need prototyping??

Two “points of interest” for companies to adopt prototyping based methodologies are:

- **Point 1:** They allow us to reduce the cost and time-to-market of a system.

- **Point 2:** For companies building critical systems, prototyping would help them perform formal verification when required. These methodologies provide high level of reliability in the system design and implementation.
Key Learning

- A prototype can be used to give end-users a concrete impression of the system’s capabilities
- Prototyping is becoming increasingly used for system development where rapid development is essential
- Throw-away prototyping is used to understand the system requirements
- In evolutionary prototyping, the system is developed by evolving an initial version to the final version
Key Points

- Rapid development of prototypes is essential. This may require leaving out functionality or relaxing non-functional constraints
- Prototyping techniques include the use of very high-level languages, database programming and prototype construction from reusable components
- Prototyping is essential for parts of the system such as the user interface which cannot be effectively pre-specified. Users must be involved in prototype evaluation