**ITEC 630**

*Information Systems Analysis, Modeling, and Design*

***Lecture Notes***

**Agile Modeling and Prototyping**

Created by Daniel LeRevised by Daniel Le 08/11/2016

**Learning objectives**

1. Use prototyping for gathering information system requirements
2. Recognize the advantages and disadvantages of prototyping applicable to systems developments.
3. Understand the relation between the Agile modeling and the prototyping and the roots of Agile modeling in prototyping.
4. Recognize the Agile modeling as an alternative to the traditional waterfall development
5. Learn the four core values of the Agile modeling
6. Learn the Agile modeling principles and their applications in to a project

**Overview**

This week lecture “Agile Modeling and Prototyping” introduces another information gathering technique called prototyping. Prototyping is useful for supplementing the traditional systems development life cycle and in seeking its user reactions to determine the project’s requirements. Next, it discusses the agile modeling, which is a user-centered software development approach. This modeling has its root in prototyping and it could adjust rapidly to changing users’ requirement to produce a finished application. In an Agile development environment, prototyping sessions are used to elicit and analyze requirements, and to construct and test working functionality.

***Note #1: All links provided in this lecture can be activated with a "Ctrl + Click"; however, you can also activate these links by copy and paste the link content to the Web browser address bar, just in case.***

***Note #2: To access links associated with "http://library.books24x7.com.ezproxy.umuc.edu", you might have to log into UMUC Library and activate the link "Books24x7" first.***

**PROTOTYPING**

Prototyping is a technique to build quickly a simplified version of an application called a prototype that can be used for eliciting, analyzing, demonstrating and validating requirements. Systems analysts can use a prototype to work with system users to identify the project’s requirements in order to help build the final application. The basic principle of prototyping is that system users know what they are looking for when they see it working or when they can use and test it. A prototype can be used to prove the systems concepts, to show users how the final application looks like, or to test users interface, code designs, or documentation tools. As a result, prototyping can be applied for almost every phase of system development.

**Kinds of prototypes**

1. Patched-up prototypes
2. Non-operational prototypes
3. First of Series prototypes
4. Selected Features prototypes
5. Throw-away prototypes

**Prototype development guidelines**

* Work in manageable modules
* Build the prototype rapidly
* Modify the prototype in successive iterations
* Emphasize the user interface—it should be friendly and meeting user requirements.
* ***Prototyping***

***Database Systems*** [**http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/databases/9781484208779/appendix-3-review-of-information-gathering-techniques/sec14\_html\_15?query=((patched-up+prototype))#snippet**](http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/databases/9781484208779/appendix-3-review-of-information-gathering-techniques/sec14_html_15?query=((patched-up+prototype))#snippet)

***Or***

***Software Engineering: A Methodical Approach - (Go to the section 5.6 Prototyping)***

[**http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/9781484208472/chapter-5-information-gathering/sec11\_html\_3?query=((patched-up+prototype))#snippet**](http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/9781484208472/chapter-5-information-gathering/sec11_html_3?query=((patched-up+prototype))#snippet)

**Advantages and Disadvantages of Prototyping**

Prototypes are useful for clarification on any uncertainty areas during the system development; however, the prototyping has some problems that need to be addressed to avoid spinning out of control and user confusions.

* ***Prototyping and its advantages and disadvantages***

[**http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/software-requirements/9781780172774/5dot-investigation-techniques/h57\_html?query=((%22disadvantages+of+prototyping%22))#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODE3ODAxNzI3NzQlMkZoNTdfaHRtbCZxdWVyeT0oKCUyMmRpc2FkdmFudGFnZXMlMjBvZiUyMHByb3RvdHlwaW5nJTIyKSk**](http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/software-requirements/9781780172774/5dot-investigation-techniques/h57_html?query=((%22disadvantages+of+prototyping%22))#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODE3ODAxNzI3NzQlMkZoNTdfaHRtbCZxdWVyeT0oKCUyMmRpc2FkdmFudGFnZXMlMjBvZiUyMHByb3RvdHlwaW5nJTIyKSk)

**AGILE MODELING**

Agile modeling is used to plan quickly, develop and release software quickly, and revise software quickly. Similar to prototyping model, the basic principle of agile development is that the system development lifecycle should be quick and responsive. In order to shorten the development lifecycle, the agile model promotes iterative development processes by producing small software releases quickly and collecting users’ feedbacks for systems requirements revision and improvement.

The Agile methodology is an alternative to the traditional waterfall sequential methodology in the area of software development. The Agile software development refers to a set of methods and methodologies based on iterative and incremental development. It helps software think more effectively, work more efficiently, and make better decisions. One of the core values of the Agile project management is the response to change and based on its iterative and incremental foundation, the Agile methodology could help project teams respond quickly to customers, product users, and the market.

* **What Is Agile?**

[**http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/agile-development/9781449363819/1dot-learning-agile/chapter-idp4397568#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODE0NDkzNjM4MTklMkZzZWN0MS1pZHA0NDE0ODgwJnF1ZXJ5PSgoYWdpbGUlMjBwaGFzZXMpKQ**](http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/agile-development/9781449363819/1dot-learning-agile/chapter-idp4397568#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODE0NDkzNjM4MTklMkZzZWN0MS1pZHA0NDE0ODgwJnF1ZXJ5PSgoYWdpbGUlMjBwaGFzZXMpKQ)**==**

**Understanding Agile Values**

The Agile methodology consists of a set of four core values called the “Manifesto for Agile Software Development”:

* Individuals and interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan
* ***Agile philosophy***

[**http://agilemanifesto.org/**](http://agilemanifesto.org/)

* ***Chapter 2.Understanding Agile Values (Read the entire Chapter 2)***

[**http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/agile-development/9781449363819/1dot-learning-agile/chapter-idp4397568#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODE0NDkzNjM4MTklMkZjaGFwdGVyXzJfX3VuZGVyc3RhbmRpbmdfYWdpbGUmcXVlcnk9KChhZ2lsZSUyMHBoYXNlcykp**](http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/agile-development/9781449363819/1dot-learning-agile/chapter-idp4397568#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODE0NDkzNjM4MTklMkZjaGFwdGVyXzJfX3VuZGVyc3RhbmRpbmdfYWdpbGUmcXVlcnk9KChhZ2lsZSUyMHBoYXNlcykp)

**Principles of Agile Software**

In addition to the four core values, there are 12 principles that Agile developers should keep in mind when working on a software project team in order to keep their team and project on the right track.

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

* ***Principles of Agile Software***

[**http://agilemanifesto.org/principles.html**](http://agilemanifesto.org/principles.html)

* ***The 12 Principles of Agile Software***

**http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/agile-development/9781449363819/3dot-the-agile-principles/chapter\_3\_\_the\_agile\_principle#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODE0NDkzNjM4MTklMkZ0aGVfdHdlbHZlX3ByaW5jaXBsZXNfb2ZfYWdpbGUmcXVlcnk9KChhZ2lsZSUyMHBoYXNlcykp**

**Application of the 12 Principles into a Real Project**

The application into a real project of these 12 principles is arranged into four sections: delivery, communication, execution, and improvement.

* ***Chapter 3. The Agile Principles (Read the entire Chapter 3)***

**http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/agile-development/9781449363819/3dot-the-agile-principles/the\_customer\_is\_always\_right\_r#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODE0NDkzNjM4MTklMkZjaGFwdGVyXzNfX3RoZV9hZ2lsZV9wcmluY2lwbGUmcXVlcnk9KChhZ2lsZSUyMHBoYXNlcykp**

**References**

1. Debra Paul, James Cadle, and Donald Yeates (2014). Business Analysis - Third edition, BCS Learning & Development Limited.

<http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/software-requirements/9781780172774>

1. Elvis C. Foster (2014). Software Engineering: A Methodical Approach, Apress.

<http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/9781484208472>

1. Elvis C. Foster and Shripad V. Godbole (2014). Database Systems, Apress.

<http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/databases/9781484208779>

1. Jennifer Greene and Andrew Stellman (2014). Learning Agile, O'Reilly Media, Inc.

<http://proquestcombo.safaribooksonline.com.ezproxy.umuc.edu/book/software-engineering-and-development/agile-development/9781449363819>

1. http://agilemanifesto.org/
2. <http://agilemanifesto.org/principles.html>

***Disclaimer:*** *Some articles or sites that I refer you to may include materials or opinions associated with particular political or other ideological positions. My pointing to these sites in no way suggests that I am encouraging a particular ideology or position on this or any other related topic; these are simply some of the more interesting and informative sites I have found to cover the topic(s) at hand. As always, you must consider the source when reading any materials that may reflect a particular ideological point of view on an issue.*