St. Xavier’s College

 Maitighar, Kathmandu

**Software Engineering**

Assignment #05

**Submitted By:**

Utkarshika Baidya

BIM 6th Semester

Roll No.: 014BIM061

Section: ‘B’

**Submitted To:**

Mr. Bal Krishna Subedi

Lecturer

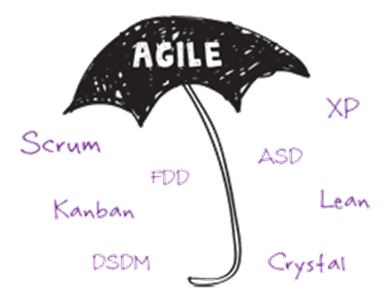
Department of Computer Science

**Agile**

Agile software development methodology is a process for developing software (like other software development methodologies – Waterfall model, V-Model, Iterative model etc.) However, Agile methodology differs significantly from other methodologies. In English, Agile means ‘ability to move quickly and easily’ and responding swiftly to change – this is a key aspect of Agile software development as well.

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. Agile methods or Agile processes generally promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals. [Agile development](https://www.cprime.com/2016/03/what-is-agile-product-development/) refers to any development process that is aligned with the concepts of the Agile Manifesto. The Manifesto was developed by a group fourteen leading figures in the software industry, and reflects their experience of what approaches do and do not work for software development. Read more about the [Agile Manifesto.](http://agilemanifesto.org/)

Agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications. Extreme Programming (XP) is currently one of the most well-known agile development life cycle models.

****

**Advantages of Agile model:**

* Customer satisfaction by rapid, continuous delivery of useful software.
* People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
* Working software is delivered frequently (weeks rather than months).
* Face-to-face conversation is the best form of communication.
* Close daily cooperation between business people and developers.
* Continuous attention to technical excellence and good design.
* Regular adaptation to changing circumstances.
* Even late changes in requirements are welcomed

**Disadvantages of Agile model:**

* In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
* There is lack of emphasis on necessary designing and documentation.
* The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
* Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

**RAD**

The RAD (Rapid Application Development) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.

Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes (components), continuous integration and rapid delivery.

RAD Model Design

RAD model distributes the analysis, design, build and test phases into a series of short, iterative development cycles.

Following are the various phases of the RAD Model −

**Business Modeling**

The business model for the product under development is designed in terms of flow of information and the distribution of information between various business channels. A complete business analysis is performed to find the vital information for business, how it can be obtained, how and when is the information processed and what are the factors driving successful flow of information.

**Data Modeling**

The information gathered in the Business Modeling phase is reviewed and analyzed to form sets of data objects vital for the business. The attributes of all data sets is identified and defined. The relation between these data objects are established and defined in detail in relevance to the business model.

**Process Modeling**

The data object sets defined in the Data Modeling phase are converted to establish the business information flow needed to achieve specific business objectives as per the business model. The process model for any changes or enhancements to the data object sets is defined in this phase. Process descriptions for adding, deleting, retrieving or modifying a data object are given.

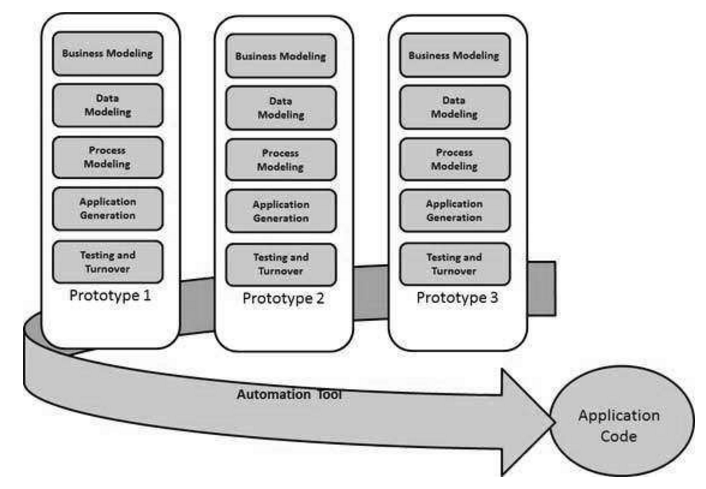
**Application Generation**

The actual system is built and coding is done by using automation tools to convert process and data models into actual prototypes.

**Testing and Turnover**

The overall testing time is reduced in the RAD model as the prototypes are independently tested during every iteration. However, the data flow and the interfaces between all the components need to be thoroughly tested with complete test coverage. Since most of the programming components have already been tested, it reduces the risk of any major issues.

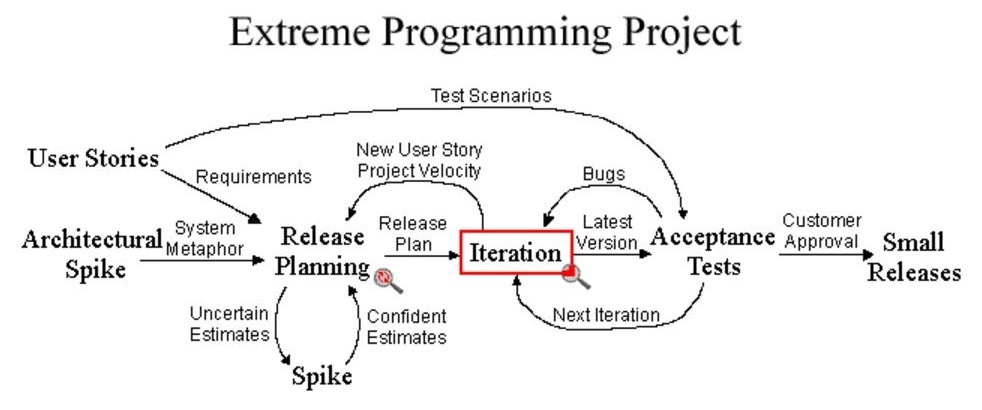
The following illustration describes the RAD Model in detail.



**Extreme Programming (XP)**

XP, originally described by Kent Beck, has emerged as one of the most popular and controversial agile methodologies. XP is a disciplined approach to delivering high-quality software quickly and continuously. It promotes high customer involvement, rapid feedback loops, continuous testing, continuous planning, and close teamwork to deliver working software at very frequent intervals, typically every 1-3 weeks.  
The original XP recipe is based on four simple values  simplicity, communication, feedback, and courage  and twelve supporting practices:

* Planning Game
* Small Releases
* Customer Acceptance Tests
* Simple Design
* Pair Programming
* Test-Driven Development
* Refactoring
* Continuous Integration
* Collective Code Ownership
* Coding Standards
* Metaphor
* Sustainable Pace



In XP, the “Customer” works very closely with the development team to define and prioritize granular units of functionality referred to as “User Stories”. The development team estimates, plans, and delivers the highest priority user stories in the form of working, tested software on an iteration-by-iteration basis. In order to maximize productivity, the practices provide a supportive, lightweight framework to guide a team and ensure high-quality software.

**Scrum**

Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one.

* A “process framework” is a particular set of practices that must be followed in order for a process to be consistent with the framework. (For example, the Scrum process framework requires the use of development cycles called Sprints, the XP framework requires pair programming, and so forth.)
* “Lightweight” means that the overhead of the process is kept as small as possible, to maximize the amount of productive time available for getting useful work done.

A [Scrum process](https://www.cprime.com/2015/03/5-tips-to-manage-scrum-processes-in-the-real-world/) is distinguished from other agile processes by specific concepts and practices, divided into the three categories of Roles, Artifacts, and Time Boxes. These and other terms used in Scrum are defined below. Scrum is most often used to manage complex software and product development, using iterative and incremental practices. Scrum significantly increases productivity and reduces time to benefits relative to classic “[waterfall](https://www.cprime.com/2011/01/integrating-waterfall-and-agile-development-hybrid-model/)” processes. Scrum processes enable organizations to adjust smoothly to rapidly-changing requirements, and produce a product that meets evolving business goals. An agile Scrum process benefits the organization by helping it to

* Increase the quality of the deliverables
* Cope better with change (and expect the changes)
* Provide better estimates while spending less time creating them
* Be more in control of the project schedule and state

**Reference:**

[1] <https://www.versionone.com/agile-101/agile-methodologies/>

[2] <https://www.versionone.com/agile-101/>

[3] <http://agilemethodology.org/>

[4] <https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm>