Alexander Ramsey  
UMUC SWEN 603

The Rapid Application Development and Agile Model

Following on the principles of manufacturing, presented in Waterfall, had some implications for the rapid changing industry of software development. Disruptions in technology, language, and mediums of exchange caused traditional development models to become cumbersome and inefficient by the late 1980s. A need for a more streamlined, efficient approach became present and was first coined as Rapid Application Development (RAD) by James Martin in 1991 (Martin, 1993). Martin proposed that RAD would greatly accelerate a project due to a shift in how systems were developed with the client rather than for the client. The end user would interface with the development teams to analyze and provide feedback throughout the process. Outside of this concept, RAD is a broader term which represents the category for other rapid development methodologies to slot into. For example, Agile methodology, which is a current popular methodology would be considered a sect of RAD.

The RAD process typically consists of 4 main phases: 1) Requirements, 2) User Design, 3) Construction, and 4) Rollout. When beginning a project, client requirements are gathered to determine the best outcomes for a particular problem or set of problems that can be broken down into an iterative development process. Development is to occur during the prototyping phase, contributing to the iterative cycle of feedback from the client. In addition, the prototypes generated should lend themselves to being reusable and extensible so that there is limited waste during the iteration of development. This process of communication extends itself throughout development and requires that employees are trained in the right areas and that the clients are willing and able to participate in a significant amount of facilitated discussions and analysis.

Traditional vs. RAD in Diagram:

Traditional:

Deploy

Test

Develop

Design

Analysis

Rapid Application Development:

**Client Meeting**

Analysis

Implementation

Test

**Prototype   
Development**

**Build**

**Refine**

*Joint Application Development Process*

*\* Development Team high caloric intake.*

The benefits of the RAD process include better quality of the end product, a greater degree of control over risk, and a higher probability that projects will be completed on time and within budget. During the Joint Application Development process, a great deal of effort is produced by both the client and the development team, with joint meetings leading to a better communication model for success of the project. These meetings may involve the need for donuts to ensure that the teams are working at their maximum effectiveness. In addition, by having open communication with the client, there is a reduced risk for communication related errors between the end user and the development team. These meetings will decrease uncertainty in the project. In the Waterfall model, these types of errors in communication and development can lead to greater costs and development windfalls later in the project. This can be best related to the pilots 1 in 60 rule, which dictates that for every sixty miles a pilot has travelled, there will be a mile difference per one degree of error in heading. So, in a development project that spans 6 months, you may see down the road many issues pop up that greatly deviate from the original scope. RAD aims to nail this problem and prevent it with solid communication through prototyping.

The disadvantages of RAD present themselves when the project, people, or available resources are not a good fit. The first bad fit may present itself when the approach is unknown to a traditional, Waterfall acquainted developer. The development team must be adept at not only the technical requirements, but also with communicating in a manner that spurs the furthering of the development process with the client. In addition, the development team and client must be willing to contribute a significant amount of time to ensure the project success. An additional disadvantage is when the project must be structured or follow a specific set of development principles. There is a possibility that the developers may not follow such standards due to the rapid and iterative approach. Prototyping as a process can present many opportunities for time and resources to be wasted. For example, when a developer focuses on the User Interface requirements of the client rather than the overall program process, there are certain elements that may be overdeveloped, diverting the precious time with the client away from the end goal: the overall functionality and value adding business/organizational components.

Rapid Application Development is a powerful idea that the process of software development can be a part of the client relationship. Harnessing this power is the most difficult aspect of the methodology and requires excellent communication skills as well as advanced technical ability that suits the needs of the client at the moment of inception. This combination when available, though, can provide significant efficiency in development.

Martin, J. (1993). James Martin: 'You are taking a terrible risk if you don't do fast development.' (Interview). Computerworld, (22). 83.