ARINC Leverages the Rational Unified Process on Flagship Product with Help from Blueprint Technologies

Industry:
Transportation and Aerospace & Defense

Organization:
ARINC, Inc.

Description:
ARINC, Inc. develops and operates communications and information processing systems for the aviation and transportation industries and provides systems engineering and integration solutions to the government and industry. Founded in 1929 to provide reliable and efficient radio communications for the airlines, ARINC is headquartered in Annapolis, Maryland, U.S.A., with over 2,800 employees worldwide.

Business Problem:
After embarking on the development of a very large and complex state-of-the-art digital communication network, project management realized its existing software development process would not scale to meet project cost and schedule demands.

Rational Solution:
The effective application of the Rational Unified Process

Key Benefits:
- Improved communication
- Tangible results, in terms of working software, demonstrated early
- The reduction of key project risks
- Process improvement with each iteration

Technical Architecture:
Development Tools: Rational Rose, Rational SoDA, Rational RequisitePro, MS Visual C++, Tivoli
Platforms: UNIX (PSOS)
Number of Developers: Over 100

Rational®
the e-development company™

Iterative Development Positions Company for the Future

ARINC, Inc. develops and operates communications and information processing systems for the aviation and transportation industries and provides systems engineering and integration solutions to the government and industry. Founded in 1929 to provide reliable and efficient radio communications for the airlines, ARINC is headquartered in Annapolis, Maryland, U.S.A., with over 2,800 employees worldwide.

In 1997, ARINC embarked on a project critical to maintaining its leadership position — the development of a state-of-the-art digital communication network that will allow commercial aircraft around the world to send and receive messages concerning flight information, scheduling and operations thereby providing increased safety and efficiency for passengers. The VHF Digital Link (VDL) system development is expected to take more than five years, and the efforts of over 100 people, to develop. The majority of the development effort is allocated to software.

Waterfall Development Approach Hinders Progress

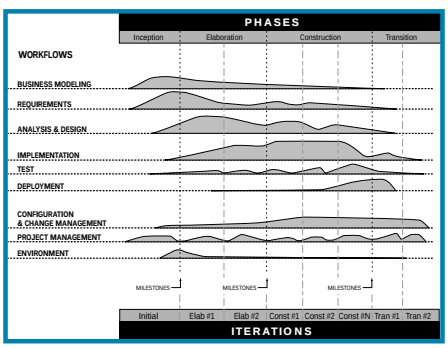
Nearly a year into this critical project, Bob Blakey, ARINC Director Aeronautical Telecommunications Network Engneering, product development director for the VDL system, realized he would need to make some changes. Blakey noticed several issues — largely because the project was using a waterfall approach to development. Blakey stated, "We were spending too much time capturing, refining and documenting our requirements. The VDL system is very large. We needed an iterative development approach to focus on our critical risk areas early in the project."

"The overriding problem was that the team was developing a very large, complex software system but did not have a scalable process to attack the issues," explained Roger Hebden, vice president of Blueprint Technologies, the software architecture firm partnering with ARINC on the program. "Because of the process' inability to scale, the team was not effective at formulating requirements or at developing a design and architecture to satisfy these requirements."

Blakey knew that ARINC needed a way to ensure the success of the VDL program. Adopting an iterative process would allow the team to solve the basic problem of how to attack the issues at hand by breaking them down into manageable pieces that could be handled one at a time. The Rational Unified Process provided that solution. "The Rational Unified Process has emerged as the industry's de facto standard for iterative development. This gave us added confidence" said Blakey. "That's why we chose it."

Introduction of Process Provides Immediate Value

The application of a defined approach, provided by the Rational Unified Process (RUP), had an immediate effect on the team's operations. The team was able to put a viable working process in place — and demonstrate it — in just three months. "Defining a process is a significant task," Hebden explained. "The RUP was instrumental because it is already written. It provided a springboard for the organization to quickly define a working process — which can sometimes take years to write. All we had to do was customize it."



▸ The Rational Unified Process organizes projects in terms of workflows and phases, each consisting of one or more iterations.

"Abstract modeling with the Rational Unified Process forced us to get out of the quagmire of technology and detail and think about the solution from a higher level — allowing us to come up with a more pure, more robust, clearer, simpler, easier solution. Understanding the core solution would have otherwise been impossible."

Because RUP provided team members with a common vocabulary of a well-defined and proven set of terminology, communication problems between individuals and groups were quickly — and greatly — reduced. "As with any large project, the VDL team had a diverse set of engineers and managers, each with their own process vocabulary. Instead of trying to force fit existing or newly fabricated terms into the process, they spent time learning those terms defined by RUP. The results were immediately apparent," noted Hebden.

Use Cases Help Define User Functionality

Use cases, descriptions of dialogues or interactions between system users and the system, are a fundamental element of RUP and one that facilitates the rest of the process. VDL engineers wrote down and then analyzed the essential functionality of the system they were designing. As a result, previously unseen problems became immediately visible. While using cases at first seemed extraneous, the team quickly embraced the approach. "On the first day, the team was amazed at all the deficiencies that were illuminated," Hebden recalled.

Analysts on the team found that they could communicate the project's scope and boundaries more easily with use cases and were able to place more focus on what the system needed to do and less on how it would ultimately be accomplished. Additionally, the team was able to prioritize system functionality, better quantify which functionality was most significant to the system, link requirements to iterations, and improve product quality through a greater awareness of end-user needs.

Abstract Modeling Simplifies Conceptualization of Architecture

The VDL program is an embedded real-time application with a variety of complex project requirements and technologies. Because of this complexity, the team was getting bogged down in details and unable to get a clear view of the problem it had to solve. In addition, most team members had not previously performed abstract modeling. Using RUP, however, in conjunction with Rational Rose, they were able to create a domain (business) model that captured key abstractions and patterns from the real world and could then be used as the basis for the design of the system. "Abstract modeling with RUP forced us to get out of the quagmire of technology and detail and think about the solution from a higher level — allowing us to come up with a more pure, more robust, clearer, simpler, easier solution. Understanding the core solution would have otherwise been impossible," said Hebden.

Abstract modeling provided a method for the team to rapidly achieve a robust software architecture through the use of the Unified Modeling Language (UML). It also introduced additional benefits including improved team communications as team members were able to clearly understand the structure of the architecture and could more easily communicate the design to others.

Another key benefit was the ability to identify and define reusable assets, which significantly reduced the duplication of effort and provided the team with a much more streamlined architecture. "Because the architecture came into focus, software reuse became much more tangible to the team," according to Hebden. "Through RUP, team members realized that not everything in the system needed to be built from scratch — that there was an opportunity to reuse components throughout the architecture, which reduced costs and time."

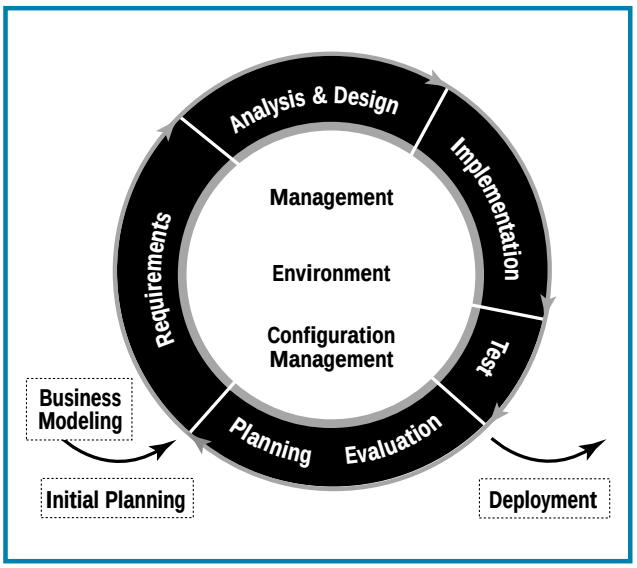
Building an Executable Architecture Accelerates Mature Design

Initially the team had concerns about executable architecture — a working baseline used to validate the architectural design of the software application. "Team members were apprehensive about taking a design and coding it in software. They thought it was too early in the project lifecycle to start coding and that the effort wouldn't provide any value. They quickly realized there was huge value in it," Hebden said. "By taking the paper design and translating it into an executable software program, like translating a blueprint of a house into a scale mockup, they learned about bad design decisions, risks that had not been identified, and flaws in the requirements — much more quickly than they had ever imagined. Our design improved dramatically as a result."

Transition to Iterative Development Moves Project Along

Iterations provided the means for the group to make steady, measurable progress. Furthermore, with each iteration, the architecture came into clearer focus and the system became more and more complete. The team started with a three-week iteration to teach a small group of key project members about the activities involved in RUP. Next came a seven-week iteration addressing a relatively simple set of use cases. In combination, the two iterations provided a successful foundation upon which RUP became a working reality.

"Iterations under RUP, helped us establish an assembly-line rhythm. After the second iteration, the team had attacked two of the most important risks of the project and had dramatically reduced the risk profile over what it was two to three months before," explained Bob Blakey. "Iterations also provided an excellent means for delimiting, and then measuring, the project's progress and quality. The result was more immediate feedback, and a more efficient development team."



▸ **Each project iterations' cycle begins with a plan for what will be accomplished and concludes with an evaluation of whether objectives have been met.**

The Rational Unified Process Addresses Immediate and Long-Term Development Needs

By modifying RUP, the ARINC team started the migration from a waterfall approach to iterative development and produced significant, measurable achievements. Team communications improved, working software was easily demonstrated to executives, risk areas were mitigated early, and the software quality improved by iteration. (Initial Operating Capability will be in September of 2000.) The project remains on track for full deployment in 2002.

On the human side, because RUP enabled the project to move forward and allowed the team to demonstrate measurable successes, frustration and miscommunications were minimized. Engineers became excited about the technology and excited about making progress.

"The RUP provided ARINC with a solution to a very immediate — and very critical — need. But it also provided an avenue for the company to adopt a state-of-the-art software development process, laying the groundwork for the organization to grow and mature its overall capability," said Blakey.

"Team members were apprehensive about taking a design and coding it in software. They thought it was too early in the project lifecycle to start coding and that the effort wouldn't provide any value. They quickly realized there was a huge value in it."

About Rational Software Corporation:

Rational Software Corporation (NASDAQ: RATL), the e-development company, helps organizations develop and deploy software for e-business, e-infrastructure, and e-devices through a combination of tools, services, and software engineering best practices. Rational's e-development solution helps organizations overcome the e-software paradox by accelerating time to market while improving quality. Rational's integrated solution simplifies the process of acquiring, deploying, and supporting a comprehensive e-software development platform, reducing total cost of ownership. Founded in 1981, Rational, one of the world's largest Internet software companies, had revenues of $572 million for its fiscal year that ended in March 2000, and employs more than 2,600 people around the world.

Rational®
the e-development company™

Corporate Headquarters:
Rational Software
18880 Homestead Rd.
Cupertino, CA 95014

Toll-free: (800) 728-1212
Tel: (408) 863-9900
Fax: (408) 863-4120
E-mail: info@rational.com
Web: www.rational.com

International Locations:
www.rational.com/worldwide

CS-175. 5/00. Subject to change without notice.