Rational University Curriculum Paths

Rational University has compiled a description of the role of the system analyst, the designer and the Enterprise architect. The following outline describes their activities and what is needed to succeed. To help you achieve your training goals, various curriculum paths reference available instructor and web-based training courses at Rational University.

System Analyst:

The System Analyst role leads and coordinates requirements and use-case modeling by outlining the system's functionality and delimiting the system. For example, establishing the actors and use cases and how they interact.

- Responsible for eliciting and writing the requirements of the software project.
- At a senior level, responsible for the requirements management plan, use-case modeling guidelines and other requirements guidelines for the whole project.

Activities performed by this role:

- Develop vision.
- Elicit stakeholder requests.
- Manage dependencies between requirements.
- Maintaining the glossary.
- Find Use cases and actors; structure the use case model.
- Detail the use cases and supplementary specifications.

What they need to succeed:

- To effectively elicit, organize and document the software requirements in the system.
- To understand how to use business-modeling artifacts as an input to system definition.
- An in-depth knowledge of requirements management.
- A quick, focused introduction to using and setting up RequisitePro.
- An in-depth knowledge of how to write a clear statement of requirements.
- To effectively capture their requirements using use cases or as declarative statements.

Curriculum Path:

http://www.ibm.com/services/learning/ites.wss/us/en?pageType=page&contentID=a0000460

Designer:

The Designer role defines the responsibilities, operations, attributes, and relationships of one or several classes, and determines how they will be adjusted to the implementation environment. In addition, the Designer role may have responsibility for one or more design packages, or design subsystems, including any classes owned by the packages or subsystems.

Activities they perform:

- Refine requirements of the use case; supplement the use case descriptions with information needed to understand internal system behavior.
- Refine requirements on the operations of design classes.
- Identify the classes that perform a use case's flow of events.
- Identify the responsibilities, attributes, and associates of the classes.
- Distribute the use-case behavior to those classes using use-case realizations.
- Note usage or architectural mechanisms.

What they need to succeed:

A solid working knowledge of:

- Use-case modeling techniques.
- System requirements.
- Software design techniques, including object-oriented analysis and design techniques, and the Unified Modeling Language.
- Technologies with which the system will be implemented.

Curriculum Path:

Rose: http://www.ibm.com/services/learning/ites.wss/us/en?pageType=page&contentID=a0000463
XDE: http://www.ibm.com/services/learning/ites.wss/us/en?pageType=page&contentID=a0000462

Enterprise Architect:

Enterprise Architects define and build the packages, subsystems, and interfaces for enterprise systems. This role "owns" the overall design of the enterprise system. *Could also be known as Sr. Software Engineer, System Architect, or Member of Technical Staff*

What They Do:

- Responsible for translating software requirements into a logical packages, subsystems, and interfaces.
- Work with XDE.
- Work with ClearCase and/or ClearQuest.
- Work with specific designer tools (XDE).

Activities They Perform:

- Define the packages, subsystems, and interfaces for the system.
- Use analysis and design mechanisms to facilitate design of the system.
- Use configuration management tools to manage their assets.

What They Need to Succeed:

- A working knowledge of the capabilities and operation of their design tools.
- A working knowledge of the capabilities and operation of Rational Software tools (including XDE and SCM tools).
- An understanding of the best practices for software development.
- An understanding of Enterprise architecture, architecture patterns, and design patterns.
- An understanding of modeling and testing.

Curriculum Path:

http://www.ibm.com/services/learning/ites.wss/us/en?pageType=page&contentID=a0000468

General Software Development

Code Complete - A Practical Handbook of Software Construction, McConnell, S., Microsoft Press, 1993.

Dynamics of Software Development, McCarthy, J., Microsoft Press, 1995.

The Mythical Man-Month, Brooks, F.P., Jr., Addison-Wesley, 1995.

Object-Oriented Software Development

Best of Booch: Designing Strategies, Booch, G., and Eykholt, E., editor, SIGS Books & Multimedia, 1996.

Designing Object-Oriented Software, Wiener, L., Wilkerson, B., and Wirfs-Brock, R., Prentice Hall, 1990.

Dictionary of Object Technology, Eykholt, E., and Firesmith, D., SIGS Books, 1995.

Object-Oriented Analysis and Design with Applications, Booch, G., Benjamin/ Cummings, 1994.

Object Oriented Design Heuristics, Riel, Arthur J., Addison-Wesley, 1996.

Object-Oriented Modeling and Design, Rumbaugh, J., and others, Prentice Hall, 1991.

Object-Oriented Software Development, Lorenz, M., Prentice Hall, 1993.

Object-Oriented Software Engineering, Jacobson, I., and others, Addison-Wesley, 1992.

*"*OMT Insights: Perspectives on Modeling*",* Rumbaugh, J., *Journal of Object-Oriented Programming*, SIGS Books & Multimedia, 1996.

The Rational Unified Process, Philippe Krutchen, Addison Wesley Longman, Inc., 1998.

Using CRC Cards, Wilkinson, N., SIGS Books, 1995.

UML

The Object Primer: Agile Model-Driven Development with UML 2.0, Scott Ambler, Cambridge University Press, 2004.

Real-Time UML: Developing Efficient Objects For Embedded Systems, Douglas, B. Powel, Addison Wesley Longman, Inc., 1998.

UML in a Nutshell: A Desktop Quick Reference, Si Alhir, Sinan, O'Reilly & Associates, Inc., 1998.

The Unified Modeling Language User Guide, Booch, G. and others, Addison-Wesley, 2003.

The Unified Modeling Language Reference Manual – Second Edition, James Rumbaugh, Ivar Jacobson, Grady Booch, Addison-Wesley, 2004.

UML Bible, Pender, T., Wiley Publishing, Inc., 2004.

UML Distilled—Third Edition, Fowler, Martin, Addison Wesley Longman Inc., 1997.

UML Toolkit, Eriksson, H., Penker, M., Lyons, B., and Fado, D., Wiley Publishing, Inc., 2004.

Unified Method for Object-Oriented Development (documentation set, v0.8), Booch, G. and Rumbaugh, J., Rational Software Corp., 1995.

Architecture

Software Architecture in Practice, Bass, Ken, Bass, Len, Clements, Paul, and Kazman, Rick, Addison-Wesley, 1998.

"The 4+1 View Model of Architecture", Kruchten, P, *IEEE Software*, November 1995.

"Foundations for the Study of Software Architecture," Perry, D.E., and Wolf, A.L., *ACM Soft. Eng. Notes*, Oct. 1992.

Systems Architecting: Creating and Building Complex Systems, Rechtin, E., Prentice Hall, 1991.

Software Architecture—Perspectives on an Emerging Discipline, Garlan, D., and Shaw, M., Prentice Hall, 1996.

Computer Architecture: A Quantitative Approach, Second Edition, Goldberg, D., Hennessy, J., and Patterson, D, Morgan Kaufman Publishers, 1996.

Patterns

Analysis Patterns: Reusable Object Models, Fowler, M., Addison-Wesley, 1996.

Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design, Larman, C., Prentice Hall Press, 1997.

Design Patterns: Elements of Reusable Object-Oriented Software, Gamma, E., and others, Addison-Wesley, 1995

Pattern-Oriented Software Architecture - A System of Patterns, Buschmann, F, Meunier, R., Rohnert, H., Sommerlad, P., and Stahl, M., John Wiley and Sons, 1996.

Project Management

Object Solutions, Booch, G., Addison-Wesley, 1996.

Software Project Management: A Unified Framework, Royce, Walker, Addison-Wesley, 1998.

Metrics

Object-Oriented Software Metrics, Kidd, J., and Lorenz, M., Prentice Hall, 1994.

User Interface Design

Designing Object-Oriented User Interfaces, Collins, D., Benjamin/Cummings, 1995.

Object-Oriented GUI Application Development, Lee, G., Prentice-Hall, 1993.

Distribution

The Essential Client/Server Survival Guide, Second Edition, Edwards, J., Harkey, D., and Orfali, R., John Wiley & Sons, 1997.

The Essential Distributed Objects Survival Guide, Edwards, J., Harkey, D., and Orfali, R., John Wiley & Sons, 1995.

Object Database

The Object Database Handbook, Barry, D., John Wiley & Sons, New York, NY, 1996

COM/OLE

Understanding Active X and OLE, Chappel, David, Microsoft Press, 1996.

David Chappel's Web site at **http://www.chappellassoc.com**

Inside COM, Rogerson, D., Microsoft Press, 1997.

COM's home page: **http://www.microsoft.com/cominfo/**

CORBA

Advanced Java Development for Enterprise Applications, Berg, Clifford J., Prentice Hall, 1998.

CORBA Fundamentals and Programming, Siegel, J., John Wiley & Sons, 1996.

Instant CORBA, Edwards, Jeri, Harkey, Dan, and Orfali, Robert, John Wiley & Sons, 1997.

For information on other books, see the Rational Unified Process recommended reading list at:
http://www.rational.com/uml/?SMSESSION=NO