Rational software

IBM

Rational Business Driven Development for Compliance

Say what you do, do what you say, and be able to prove it

Manage compliance using Rational tools and processes

Leverage compliance for business advantage

Ueli Wahli
Majid Irani
Matthew Magee
Ana Negrello
Celio Palma
Jason Smith

Redbooks

ibm.com/redbooks

International Technical Support Organization

Rational Business Driven Development for Compliance

November 2006

SG24-7244-00

Note: Before using this information and the product it supports, read the information in "Notices" on page ix.

First Edition (November 2006)

This edition applies to IBM Rational software tools, such as RequisitePro, ClearCase, ClearQuest, ClearQuest Test Manager, Portfolio Manager, BuildForge, Functional Tester, Manual Tester, Performance Tester, Method Composer, Unified Process, ProjectConsole, and SoDA.

Contents

Notices ix
Trademarks x

Preface xi
The team that wrote this IBM Redbook xii
Thanks xiii
Become a published author xiv
Comments welcome xiv

Chapter 1. A discussion about compliance 1
Overview of today’s regulated environment 2
What it means to be compliant 3
Policy creation and management 4
Regulations 5
Sarbanes-Oxley 6
USA Patriot Act 7
Basel II 7
Title 21 CFR 11 8
HIPAA 8
Gramm-Leach-Bliley 9
Sustainable compliance management 9
How auditors inspect 10
Compliance: an opportunity to improve the business. 13
Regulations versus standards 14
Software development oriented standards 16
COSO 16
COBIT 16
ITIL 17
SPICE 17
ISO 900x 18
Six Sigma 18
CMMI 19
RUP 20
Typical compliance challenges and concerns 22
External business factors impacting compliance initiatives 25

Chapter 2. Compliance guidelines 29
Why business compliance often requires software development compliance . 30
Compliance in automated business transactions 30

Governance versus compliance 31
Implementing software applications is a kind of governance 33
Compliance process requirements 35
Satisfying provisioned policies 36
Demonstration of audit data 37
Compliance cost and controls 38
Adoption of best practices 40
Workflow automation for tool-directed behavior 41
Considerations for compliant process design 42
Software development process documentation 42
Separation of duties 42
Approvals 43
Financial approvals 43
Technology approvals 44
Software development audit trails 44
Electronic signatures 45
Authentication and authorization 45
Points of control 46
Documentation and reports 46
Metrics 47
Defining metrics and measures 48
Selection of key metrics 49
General considerations and practical control strategies 50
Leveraging the infrastructure 50
High trust and high security 51
High and low ceremony 51
Establishing your organizational position 52
Practical control strategies 52
General strategy 52
Define a compliant software development process 53
Assign roles, responsibilities, and reporting requirements 54
Establish points of control in the flow 54
The three points of control strategy 54
Control point 1: Deliver change 55
Control point 2: Register derived objects 55
Control point 3: Deploy objects 56
Roles and responsibilities in this strategy 56

Chapter 3. IBM Rational's key capabilities for compliance management 59
The IBM Rational solution for compliance management 60
Business-driven development for compliance 61
Control principles 65
Development governance 66
Discussion 67

Supporting tools 71
Rational Portfolio Manager 71
Supporting process 73
Conclusion 73
Requirements management 73
Discussion 74
Supporting tools 75
Rational RequisitePro 75
WebSphere Business Modeler 78
Supporting process 78
Conclusion 79
Software change management 79
Discussion 80
Supporting tools 81
Rational ClearQuest 81
Rational ClearCase 82
Supporting process 84
Conclusion 84
Build and deployment management 85
Discussion 85
Supporting tools 88
Rational BuildForge 90
Tivoli Provisioning Manager 90
Supporting process 91
Conclusion 91
Reference 92
Software validation 92
Discussion 92
Supporting tools 93
Rational ClearQuest Test Manager 94
Rational Functional Tester 99
Rational Manual Tester 100
Rational Performance Tester 100
Supporting process 100
Conclusion 100
Validating your compliance process 101
Discussion 102
Supporting tools 106
Rational Method Composer 106
Rational Unified Process 106
RUP Plug-in for Compliance Management 107
Supporting process 107
Conclusion 107

Documenting results 108
Discussion 108
Supporting tools 110
Rational ProjectConsole 110
Rational SoDA 111
Supporting process 111
Conclusion 112
Defining a customized solution for an organization 112
Three-phases approach 113
Extensibility 118

Chapter 4. Roadmap to compliance management 119
A solution using Rational tools 120
Compliance roadmap 121
Roles 122
Capability patterns 123
Rational tools 124
Solution details 125
Process modeling and design for compliance 125
Objectives 126
Challenges 126
Discussion 127
Activities 128
Conclusion 128
Manage compliance portfolio 129
Objectives 129
Challenges 130
Discussion 130
Activities 136
Conclusion 136
Manage projects 137
Objectives 137
Challenges 138
Discussion 138
Activities 139
Conclusion 140
Initiate compliance project 141
Objectives 142
Challenges 142
Discussion 142
Activities 143
Conclusion 143
Deliver auditable change 144
Objectives 145

Challenges 146
Discussion 146
Activities 148
Conclusion 149
Securely deploy a release 149
Objectives 150
Challenges 150
Discussion 151
Activities 153
Conclusion 153
Validate automated business controls 154
Objectives 155
Challenges 155
Discussion 155
Activities 157
Conclusion 157
Manage and generate metrics and reports 158
Objectives 158
Challenges 159
Discussion 159
Activities 162
Conclusion 162
Summary 162

Appendix A. Unified Method Architecture 163
Unified Method Architecture defined 164
Fundamental principles of UMA 164
The basic elements of UMA 164
Key concepts of UMA 166
Method content 166
Development process 167
Separation of method content and process 168
Content reuse 168
Process families 168
Multiple life cycles 169
Flexible extensibility and plug-in mechanisms 169
Multiple process views 169
Reusable process patterns 170

Abbreviations and acronyms 171

Related publications 173
IBM Redbooks 173
Online resources 173

Rational 173
Tivoli 174
Compliance regulations. 174
How to get IBM Redbooks 175
Help from IBM 175

Index 177

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

eserver®
eserver®
Redbooks (logo) ™
z/OS®
ClearCase®
ClearQuest®
DB2®
IBM®
MQSeries®
OMEGAMON®
ProjectConsole™
Rational Method Composer®
Rational Portfolio Manager®
Rational Summit®
Rational Unified Process®
Rational®
Redbooks™
RequisitePro®
RUP®
SoDA®
Summit®
Team Unifying Platform™
Tivoli®
WebSphere®
Workplace™

The following terms are trademarks of other companies:

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Visual Basic, Visual Studio, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbook will help you design your processes and solutions using Rational® products to manage compliance. It shows how IBM Rational solutions can be applied by an organization to realize a controlled, auditable software development environment to help alleviate specific challenges imposed by standards, policies, and regulations.

This book provides a usage model and product configuration guidance to help a tools administrator implement and configure some or all of the Rational tools to address compliance challenges. This book will help you and your partners understand the design and deployment of IBM Rational's Business Driven Development for Compliance solution.

This book focuses on problems addressed by Rational products. The emphasis is on auditable change processes, in particular using ClearQuest® and ClearCase®. Additional coverage is on requirements for compliance using RequisitePro® and compliance attestation using ClearQuest Test Manager, IT governance using Rational Portfolio Manager®, Rational Method Composer®, and the Rational Unified Process® (RUP®).

This book presents a solution for compliance management that demonstrates support of all three dimensions of business driven development of software for compliance, *say what you do, do what you say*, and *be able to prove it*.

The team that wrote this IBM Redbook

This IBM Redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Ueli Wahli is a Consultant IT Specialist at the IBM International Technical Support Organization in San Jose, California. Before joining the ITSO over 20 years ago, Ueli worked in technical support at IBM Switzerland. He writes extensively and teaches IBM classes worldwide about WebSphere® Application Server and WebSphere and Rational application development products. In his ITSO career, Ueli has produced more than 30 IBM Redbooks. Ueli holds a degree in Mathematics from the Swiss Federal Institute of Technology.

Majid Irani is a Senior IT Specialist at IBM Software Group, Rational in Cupertino, California. He works with IBM teams, performs on-site assessments, and delivers workshops to ensure successful deployment and adoption of Rational UCM/SCM solutions. He has over 10 years of experience in software configuration management and is a Rational certified engineer. He has been with IBM for over six years. Majid holds a Bachelor’s degree in Computer Science and Engineering from the University of California in Los Angeles.

Ana Negrello is an IBM Certified IT Specialist in the IBM Software Group in Brazil, specializing in process, requirements, and project management. She has worked in software engineering for the last 20 years. She graduated with a degree in Computer Science from the University of Campinas and earned a post-graduate degree in Marketing at the Escola Superior de Propaganda e Marketing in Sao Paulo, Brazil.

Celio Palma is an IT Specialist in Uruguay. He is the leader of the tools group for BCS Uruguay, a team member of the tools and technology core group for SSA, Rational focal point for BCS Uruguay, and SCM responsible for BPS projects. He has seven years of experience in the service delivery center and two years of experience in tools configuration and deployment. His experience includes participation in CMMi level 4 and 5 certifications and IBM Rational tools administration. He is a Computer Analyst, holding a degree from Facultad de Ingeniería at the Universidad de la República, Montevideo, Uruguay.

Jason Smith is a Senior IT Specialist for IBM Rational Sales based in Waltham, Massachusetts. He has 14 years of experience in software configuration management, the last eleven supporting Rational tools. He holds Rational certifications in ClearCase, ClearQuest, and UCM. He has been with IBM for just over a year, and before that worked for a variety of start-up companies in the New England area. Jason holds a degree in Electrical Engineering from Tufts University in Medford, Massachusetts.

Thanks

A special thank you to **Matthew Magee**, the author of a white paper on IBM Rational's solution for compliance that was used as a primary input for this IBM Redbook. Matthew also provided a thorough review and partial rewrite of several chapters.

Thanks to the following people for their contributions to this project:

- **Lynn Mueller**, IBM West Chester, for her overall project guidance, support, and content reviews.
- **Jamie Klein**, IBM Lexington, for her legal guidance and review of the contents of this book.
- **Thierry Paradan**, IBM Cupertino, and **Zoe Eason**, IBM UK, for their contribution regarding the Unified Method Architecture and the RUP plug-in for compliance management.
- **Robyn Gold**, **Bernie Coyne**, **David Lubanko**, and **Paul Tasillo**, IBM Lexington; **Roger Le Blanc**, IBM Minneapolis; **Joseph Meagher**, IBM Dallas; **Dudley Thomas**, IBM Nashville; and **Calvin Powers**, IBM Durham, for providing help, reviews, and material for this book.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

A discussion about compliance

"*If companies view the new laws as opportunities – opportunities to improve internal controls, improve the performance of the board, and improve their public reporting – they will ultimately be better run, more transparent, and therefore more attractive to investors.*"

-William Donaldson, former SEC Chairman

This chapter explores the compliance environment and introduces key regulations and standard frameworks. We also discuss business controls, policy management, audits, and compliance challenges.

Disclaimer: The reader is responsible for ensuring his own compliance with legal requirements. It is the reader's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the reader's business and any actions the reader may have to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the reader is in compliance with any law.

Overview of today’s regulated environment

When asked the question, “Do you operate in a regulated industry?” many businesses will respond with a resounding “No.” The reality is that nearly all businesses operating in the United States and abroad are directly or indirectly influenced by regulations, standards, and policies. For example, in the United States, businesses must comply with Occupational Safety & Health Administration (OSHA) regulations. In Europe, the European Agency for Safety and Health at Work (EASHW) provides similar legislation. Generally, it is fair to say that if you operate in a country with a structured form of government, you are regulated in some way.

Moreover, *regulating* frameworks can take a variety of forms, for example, federal or state laws, governmental regulations or codes, industrial organizations’ standards and guidelines, corporate policies, and ethical practices. All of these sources of *regulation* can create a dizzying atmosphere in which to operate a business.

Recognizing whether you are a regulated business often seems in some way to have more to do with the impact it has on your bottom line than whether the regulations actually exist. As regulations become part of standard operating procedures within an industry, the actual impact of these regulations is easily overlooked. For example, walk through a mall or restaurant near closing hours and you may see a familiar “Wet Floor” warning sign. Although these signs originated in response to OSHA regulations, today they are typically regarded as a standard operating procedure, and as a cost of doing business.

Regulations often share the following common characteristics:

- *Mandatory*—Regulatory authorities may require qualification or adherence to specific standards, and may dictate explicit penalties for non-compliance.
- *Non-directive*—While regulatory authorities typically define *what* needs to be done, rarely do regulatory authorities detail the *how*, that is, the specific steps that companies must take in order to ensure compliance. The interpretation of regulations into policies and operating procedures is left up to organizations and their legal and policy-making staffs.
- *Continuously changing*—Regulations, their interpretation, and their adoption/absorption/reflection in corporate policies can change frequently and at inconvenient times. Commonly accepted standards for meeting regulatory requirements can evolve as regulations are tested and their interpretations refined. Regulatory authorities themselves can be malleable. For example, when the Food and Drug Administration (FDA) developed guidance governing the implementation of the Title 21 Part 11, Electronic

Records and Signatures Rule, guidance was developed and revoked four times before the agency settled on a final direction.

- *Increasing IT impact*—While few apply directly to IT, regulations are increasingly affecting IT systems.
- *Non-certifiable*—Agencies generally do not approve, recommend, or validate a company's compliance solutions. Most agencies will not certify that any solution guarantees compliance.
- *Intrinsically related to harm*—Regulatory authorities tend to set the bar for compliance relative to the harm associated with a given occupation. To take Occupational Safety and Health regulations created by OSHA, it is generally acknowledged that a lower bar of compliance has been set for a small bookstore operation versus a steel framing or construction company where jobs are more dangerous.

  As you move from one regulated domain to another, harm and accountability fluctuate as well. The risks to life and health to the general population are significantly higher in the life sciences, bio sciences, pharmaceutical, and medical device industries. Consequently, the regulations in this area tend to set higher bars for compliance, and have a significantly higher cost of implementation and a stricter standard of accountability. Where risk of non-compliance can be measured in loss of life or health risks, regulations typically carry stiffer penalties as well.

The impact of complying with regulations, standards, and policies runs enterprise-wide, affecting people, processes, information, technology infrastructure, and facilities. Regulatory business controls impact everything from facilities ("Is the front door locked?"), to organization ("Is their an independent board of directors?"), to strategy ("Is our product strategy subject to FDA approval?").

What it means to be compliant

To be in compliance with a regulation, standard, or policy generally means that you can substantiate that you meet both the performance and the procedural requirements.

- Performance requirements demonstrate your ability to deliver functionality or tasks. Examples of performance requirements include producing a specific audit log or a financial report as mandated by a regulation.
- Procedural requirements demonstrate adherence to your documented operational process, for example, "employers will be responsible for appropriately validating the identity of employees."

Because many regulations do not tell companies exactly what they need to do, companies must rely on their own legal staffs and risk officers to interpret regulations and reflect, adopt, or absorb their interpretation into company policy.

Policy creation and management

Regulations and their interpretations are one of many inputs that drive corporate policy. Corporate policy drives business processes. These business processes in turn drive business software requirements. Given the linkage between regulations, policy, and software, it is instructive to review the policy-making process in organizations.

For most large organizations, publishing a *corporate policy* from the office of a C-level executive is a major event. While policy statements are usually short statements of principle and goals, a large amount of thought and legal review goes into them. It is not unusual for development of a policy statement to take over a year.

These policies can be such significant events because they can have wide sweeping effects on how the enterprise does business. A one-page policy statement could require huge sums of money to be spent, and occupy large numbers of employees to change the enterprises' business processes to bring them into compliance with the policy.

For example, people throughout the enterprise will create a wide variety of derivative work products to put into daily practice the principles and goals outlined in the policy statement. Employee training modules may need to be updated to reflect the requirements of the policy. Legal departments may have to update standard text in their agreements. Service level agreements with outsourcing projects may have to be updated. Accounting procedures may need to be updated to reflect new policies. These *policy artifacts* are owned by a wide variety of different people in different roles throughout the organization, but must be consistent with the policy statements they are derived from. In some cases, multiple layers of derivative work products may be needed to guide the business process changes.

Over time, large organizations can have difficulty keeping the downstream policy artifacts consistent with the policies from which they are derived. This is largely because the responsibility is spread across so many different people in different roles and usually across geographies. This is particularly difficult when a policy is changed or one of the derivative work products need to be changed. If an interpretation of how to stay in compliance when a policy is changed, what other downstream policy artifacts must be reviewed and possibly updated?

Given the interdependencies of policy artifacts, it is useful to consider structured policy management techniques that can create a *chain of traceability* among policies and policy-related artifacts in an enterprise. This can allow an enterprise to show a traceable chain from the IT controls implemented in the enterprise all the way back to the corporate level policy statement.

Regulations

Figure 1-1 shows examples of regulations and Figure 1-2 shows certain regulations that are gaining a lot of focus, by industry.

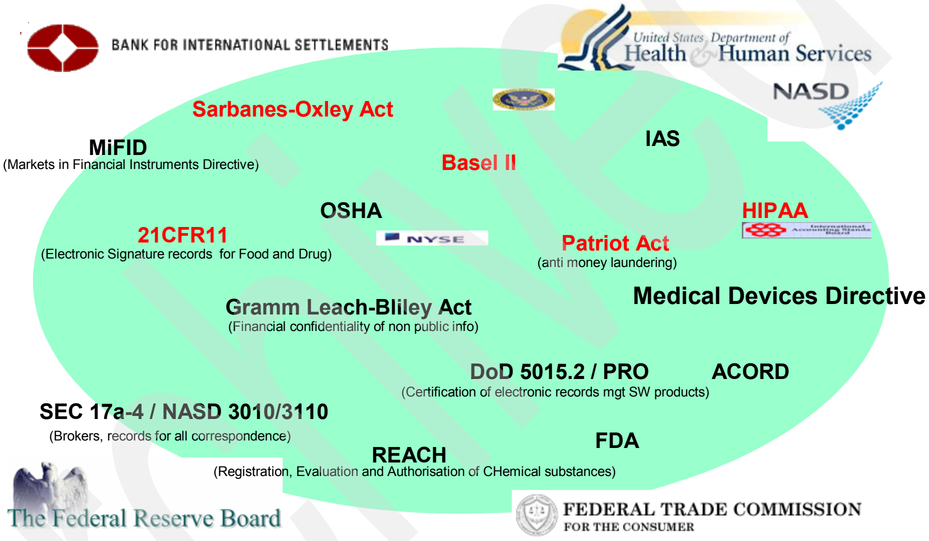
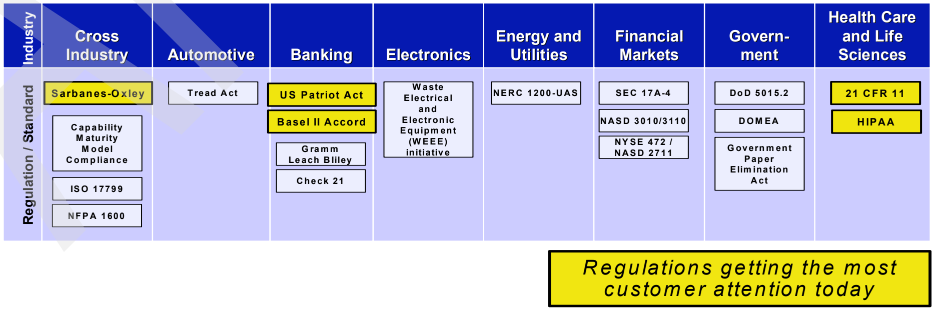

BANK FOR INTERNATIONAL SETTLEMENTS
United States Department of Health & Human Services
NASD
Sarbanes-Oxley Act
MiFID
(Markets in Financial Instruments Directive)
Basel II
IAS
OSHA
HIPAA
21CFR11
(Electronic Signature records for Food and Drug)
NYSE
Patriot Act
(anti money laundering)
International Accounting Standards Board
Gramm Leach-Bliley Act
(Financial confidentiality of non public info)
Medical Devices Directive
DoD 5015.2 / PRO
(Certification of electronic records mgt SW products)
ACORD
SEC 17a-4 / NASD 3010/3110
(Brokers, records for all correspondence)
FDA
REACH
(Registration, Evaluation and Authorisation of CHemical substances)
The Federal Reserve Board
FEDERAL TRADE COMMISSION
FOR THE CONSUMER

Figure 1-1 Examples of regulations



Industry	Cross Industry	Automotive	Banking	Electronics	Energy and Utilities	Financial Markets	Govern-ment	Health Care and Life Sciences
Regulation / Standard	Sarbanes-Oxley Capability Maturity Model Compliance ISO 17799 NFPA 1600	Tread Act	US Patriot Act Basel II Accord Gramm Leach Bliley Check 21	Waste Electrical and Electronic Equipment (WEEE) initiative	NERC 1200-UAS	SEC 17A-4 NASD 3010/3110 NYSE 472 / NASD 2711	DoD 5015.2 DOMEA Government Paper Elimination Act	21 CFR 11 HIPAA

Regulations getting the most customer attention today

Figure 1-2 Regulations getting customer attention today

The regulations that are getting the most attention today include:

- Sarbanes-Oxley
- US Patriot Act
- Basel II Accord
- 21 CFR 11
- HIPAA
- Graham-Leach-Bliley

We briefly introduce these regulations next.

Sarbanes-Oxley

Intended to restore confidence in public financial reporting in the wake of a series of public scandals, the Sarbanes-Oxley Act of 2002 (SOX) represents a fundamental change in how management carries out its fiduciary responsibilities. Not only does this mandate require a magnified look at process, but it has also becomes a catalyst for CFOs to build proactive control into processes like revenue recognition.

Some of the objectives are:

- Requires management to demonstrate knowledge of underlying process of the business
- Describes how transactions are authorized or accepted for input into processing
- Identifies critical data files used during processing
- Defines key reports resulting from processing
- Ongoing process to monitor internal controls while continuously evaluating and improving their effectiveness

Internal controls over financial reporting are those controls that ensure that the relevant financial statement assertions are complete and accurate and in accordance with Generally Accepted Accounting Principles (GAAP).

To learn more about SOX refer to:

http://www.sec.gov/news/studies/principlesbasedstand.htm

http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_bills&docid=f:h3763enr.tst.pdf

USA Patriot Act

The USA Patriot Act is aimed primarily at institutions that are regarded as gatekeepers of our nation's financial systems.

Many of its requirements are designed to verify customers' identities, that is, that they are who they say they are and that they are not on a terrorist list. To assist firms in their compliance the Office of Thrift Supervision, Department of Treasury, developed a *Preparedness Check-Up List*. The check-up list outlines a methodology, called *ADApT*, which can be used as a framework to adapt existing Banks Secrecy Act – Anti-Money Laundering solutions to adhere to the new Patriot Act requirements.

Here are the components of the ADApT methodology:

- *A* is for analyzing and understanding the components of your current program.
- *D* is for developing a comprehensive Bank Secrecy Act – Anti-Money Laundering application, which includes a customer identification program.
- *Ap* is to apply your revised program throughout the affected day-to-day operation.
- *T* is to test the new program through internal audits and testing to ensure that the program functions as intended.

As you read through the check-up list you will find many great control guidelines.

To learn more about the USA Patriot Act refer to:

`http://thomas.loc.gov/cgi-bin/query/z?c107:H.R.3162.ENR:`

Basel II

The premise of the Basel accord was to address the amount of capital that banks were required to hold against loans, which provides protection to depositors and shareholders in the event of default on the loans. The accord required that all banks use the same measure of risk for their loan portfolios. The result is that most banks are required to hold more capital than may be necessary.

To address this discrepancy Basel II was developed. This updated version of the Basel accord allows banks to use their own performance data to determine their risk and thus the amount of capital they will be required to hold. Based on this new framework, institutions must now address the systems they have in place.

Basel came about because of financial market loss (due to poor risk management practices and fraud) since 1992. Implementation of the new capital rules is due in 2007, but requires that banks are using Basel-compliant systems and data before then.

To learn more about Basel II refer to:

http://www.bis.org/publ/bcbs118.htm

Title 21 CFR 11

The Code of Federal Regulations (CFR) is a codification of US federal rules and regulations. Title 21 is allocated to the Food and Drug Administration. Chapter I Part 11 of Title 21 describes the criteria under which the agency considers electronic records, electronic signatures, and handwritten signatures executed to electronic records to be trustworthy, reliable, and generally equivalent to paper records and handwritten signatures.

To learn more about Title 21 CFR 11 refer to:

http://www.access.gpo.gov/cgi-bin/cfrassemble.cgi?title=200521
http://www.access.gpo.gov/nara/cfr/waisidx_05/21cfr11_05.html

HIPAA

The Health Insurance Portability & Accountability Act (HIPAA) adopts standards for the security of electronic protected health information to be implemented by certain health care institutions.

The Center for Medicare and Medicaid Services (CMS) has prepared a HIPAA readiness checklist to assist companies with compliance. The checklist provided is designed to help companies start to think about what they incorporate into their transaction processing to ensure that these security requirements are met.

Health care providers must have a complete understanding of how each of these processes is currently run, and how they need to be updated and documented so that customers and auditors alike can be assured the critical data is not at risk.

In summary, HIPAA provides strong protection of personal health care information.

To learn more about HIPAA refer to:

http://www.cms.hhs.gov/HIPAAGenInfo

Gramm-Leach-Bliley

The Gramm-Leach-Bliley Act requires financial institutions to disclose to consumers and customers their policies and practices for protecting the privacy of non-public personal information.

The bottom line is that each financial institution must develop an information security program that establishes administrative, technical, and physical safeguards.

To learn more about Gramm-Leach-Bliley refer to:

`http://www.ftc.gov/privacy/glbact/glbsub1.htm`

Sustainable compliance management

As organizations mature in their approach to compliance management, they often migrate from *one off* compliance firedrills to a sustainable compliance approach. They recognize common target areas of concern across regulations, standards, and policies—such as information security, risk management, data privacy—and begin to manage these concerns under a structured and unified architecture. This is what we mean by a sustainable approach to regulatory compliance management.

To be *in compliance* generally means that you can achieve both performance and procedural requirements:

- Performance requirements—Your ability to deliver functionality or tasks
- Procedural requirements—Your ability to document adherence to an established operational process

Ultimately, compliance is the ability to demonstrate that you do what you are supposed to. The end goal of compliance is being able to pass the audit now and in the future consistently.

How auditors inspect

Ultimately, an audit is a test. There are a number of ways to initiate and conduct these tests. The test can be carried out as self assessments by teams performing self inspection, or during inspections of business operations, by auditors and inspectors:

- The term *auditor* is generally used to describe someone who is on the inside of the business or has been hired by the business to conduct the business operations inspection.
- The term *inspector* usually describes an authority from a government agency that comes to perform the business operations inspection.

Auditors and inspectors typically seek the same types of information:

- Documented processes for creation of all business artifacts
- Adherence to the defined process by artifact creators
- Linkages between supported tools using tool-directed behavior (TDB)
- Process linkages to automated processes supported by tool mentors
- Linkage between artifacts that align with the business process steps
- Artifact traceability to points of accountability in the delivery chain
- Transparency of reporting
- Accountability in the chain of delivery
- Non-repudiation of any artifact throughout the system

The guiding principle for most auditors and inspectors is to look for process exceptions. In other words, auditors and inspectors will look at your process problems first, and not the consistent data that can be easily managed.

Auditing and inspection is generally carried out by exception. Typically a problem is identified and investigation commences. When an exception occurs, auditors or inspectors will find the loose thread and begin pulling on that thread to discover as many business issues as they can in a given audit window (Figure 1-3).

Figure 1-3 Analogy to an audit: loose thread

Typically, inspectors are looking for one of three things when an exception is raised: Was the exception a result of:

- System design flaw
- Human error
- Malicious behavior

Although each audit can be different, here are some of the main areas the auditors will likely investigate:

- *Evidence of documented process/adherence*—Inspectors often want to know that you have a documented process. In the case of an IT audit, for example, they may ask for your software development life cycle (SDLC) documentation. Upon review the auditors may request interviews with two or three people from the department being audited. They typically interview each person independently to determine whether their practices match both the stated documentation and other team members' practices. Should the results of these interviews not match each other as well as the documented SDLC, it is a sign of a problem that will trigger a deeper exploration.
- *Evidence of process maturity and stability*—Even if the practices match the documentation above and each other, auditors will look for corroborating evidence of process stability.

  For example, auditors may examine application development metrics, which can only be gathered consistently over time where there is process stability (otherwise the metrics become skewed, and invaluable, and may pose another remediation point).

- *Evidence of process compliance*—Documented evidence of a formal change control board (CCB) should also be demonstrated as evidence of control over both the quality process mechanism and the metrics gathering mechanisms, as well as to the applications themselves.
- *Linkages between process and artifacts*—Because the approach to systems inspection tends to be by exception, it is not uncommon for auditor or inspector to examine a problem artifact and request all of the related information for that problem artifact (artifact traceability). This presents a significant challenge to most businesses because the archeology necessary to locate most of these artifacts could be quite lengthy, if these artifacts can be located at all.
- *Linkage between documented test results and requirements*—At the very least an auditor will test a demonstrated ability to show linkages between the feature requirements of a system and the test results that conclusively illustrate compliance with those feature requirements.
- *Documented and formalized hand-offs and sign-offs*—Last but not least, any process should include some formalized authorization procedure to establish accountability for the delivery of the system from development to quality assurance (QA), and from QA to production, and ultimately to the user community at large.

As part of the investigations and to better understand how a business operates, auditors ask questions about your software development practices. Here are some examples of the types of questions that are asked:

- Which compliance requirements were delivered in release x?
- Does the application fulfill compliance requirement x?
- Who signed off on this project at design and deployment?
- Who approved funding for this project?
- Which projects will reduce compliance risk?
- Can you confirm that all of the components are in the release?
- Can you confirm that the software developed was actually deployed?
- Who made changes to the manufacturing application?

Regular inspections should be anticipated and will likely be scheduled to evaluate your level of compliances.

Compliance: an opportunity to improve the business

Of course regulations are designed to control and make companies more accountable, but they are also an excellent *opportunity* to improve the business and transform it into a better run, more transparent, and ultimately more competitive company (Figure 1-4).

> ***"If companies view the new laws as opportunities – opportunities to improve internal controls, improve the performance of the board, and improve their public reporting – they will ultimately be better run, more transparent, and therefore more attractive to investors."[1]***
>
> *William Donaldson*
> *former SEC Chairman*

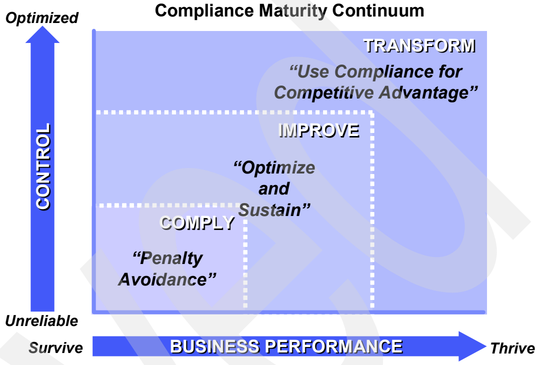


Figure 1-4 Compliance: requirement and opportunity

Often, companies are initially interested in complying with the mandates only, but quickly recognize the opportunity for improving their processes and eventually transforming their enterprises to gain competitive advantage.

Meeting regulatory requirements is more than an obligation—it is also an opportunity to improve an organization's software development transparency, oversight, and results. The implementation of a sustainable compliance architecture typically replaces ad hoc or undocumented development processes with a more structured software development process. They can also capture the project meta-data and metrics that will enable organizations to realistically assess current practices of the software development organization and iteratively improve either the practices or their execution.

In addition, reducing compliance risk is the first step toward establishing a strategic framework for IT governance to improve visibility over IT investment. Over time, the value that good IT governance can deliver continually increases. IT organizations can start out managing risk and monitoring remediation projects.

Then, by automating development workflow, companies can make the best practices operational and finally optimize their execution to enable true business transformation for the enterprise (Figure 1-5).

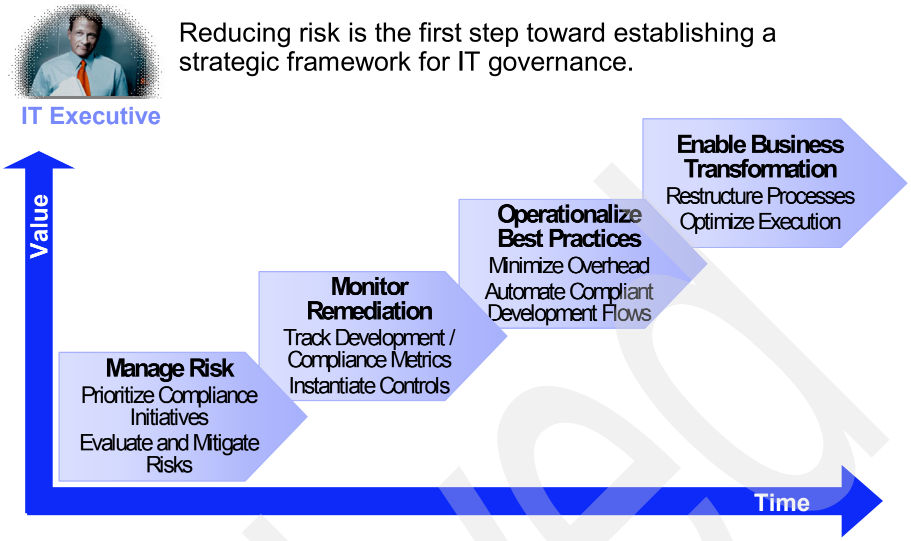

Reducing risk is the first step toward establishing a strategic framework for IT governance.
IT Executive
Value
Manage Risk
Prioritize Compliance Initiatives
Evaluate and Mitigate Risks
Monitor Remediation
Track Development / Compliance Metrics
Instantiate Controls
Operationalize Best Practices
Minimize Overhead
Automate Compliant Development Flows
Enable Business Transformation
Restructure Processes
Optimize Execution
Time

Figure 1-5 Compliance: foundation for good IT governance

Regulations versus standards

Compliance regulations are often non-directive. That means that while most regulations tell you *what* you need to do, they do not tell you *how* to do it. For example, regulations mandate that the financial officers must attest to the integrity of the financial reports for the company, but it does not say what processes they have to put in place or what other action they have to take to comply (Figure 1-6).

Generally, the executives realize that the business controls can be implemented in their IT applications and they seek out standards, such as the Capability Maturity Model Integrated (CMMI) or ISO 900x, which provide process frameworks for the organization. By adhering to such frameworks, businesses can end up implementing the kind of processes, change control, and measurement mechanisms that will institute the type of control that legislations and regulations (such as SOX and HIPAA) are looking for.

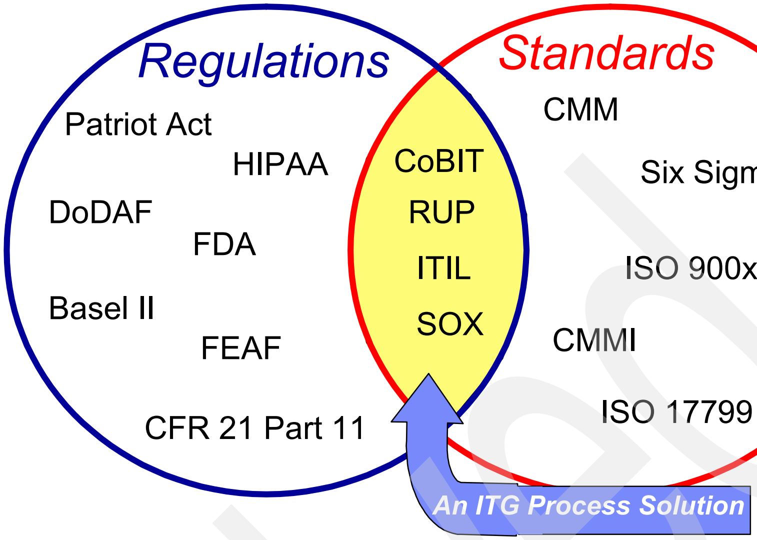


Figure 1-6 Example of regulations and standards

In addition, process standards seek to embody the fundamental philosophy of current good systems practices and quality-by-design as integral parts of business operations. This is essentially what auditors and inspectors are seeking during an inspection. Businesses are looking for a mechanism that will assist them in achieving their compliance objectives while simultaneously improving their overall capacity to produce products and services. Rather than reinvent the wheel, most businesses turn to emerging process standards as a solution. By integrating regulations and standards, organizations can formulate comprehensive IT governance solutions.

Often auditors state that it does not matter so much which framework you choose, as long as you document which one you are using and why. In practice, large organizations implement a mix of these frameworks, drawing from IT Infrastructure Library (ITIL) and Control Objectives for Information and related Technology (COBIT), for example.

There is often a layering of these frameworks, such as organizations who adopt the ITIL to govern their operations environment, COBIT to manage their IT governance efforts, and the Capability Maturity Model (CMM) implementation to govern their software development organization. As an example, SOX does not mandate use of a particular internal control standard or framework, but most companies adopt one or meld them together to come up with framework that fits their needs.

Software development oriented standards

Two of the most widely recognized frameworks for SOX-related applications are Committee of Sponsoring Organizations of the Treadway Commission (COSO) and COBIT. The ITIL is also a framework used for IT operations specific controls. Other well-known standards for software development are introduced next.

COSO

Committee of Sponsoring Organizations of the Treadway Commission (COSO) is a voluntary private sector organization dedicated to improving the quality of financial reporting though business ethics, effective internal controls, and corporate guidance.

Formed in 1985 to sponsor the National Commission on Fraudulent Financial Reporting, COSO is jointly sponsored by the five major financial professional associations in the United States. Commission members (wholly independent of each sponsoring organization) contained representatives from industry, public accounting, investment firms, and the New York Stock Exchange.

COSO promotes consistent *risk and control consciousness* throughout the enterprise and common models for discussing and evaluating risk and internal controls.

To learn more about COSO refer to:

```
http://www.coso.org
```

COBIT

Control Objectives for Information and related Technology (COBIT) helps meet the multiple needs of management by bridging the gaps between business risk control needs and technical issues. COBIT is built in part upon the COSO framework and provides *good systems practices* across domain and process framework and presents activities in a manageable and logical structure. It also serves as an independent yardstick for assessing performance. COBIT's framework principle is to link management's IT expectations with management's IT responsibilities.

To learn more about COBIT refer to:

```
http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/Tagge
dPageDisplay.cfm&TPLID=55&ContentID=7981
```

ITIL

Information Technology Infrastructure Library (ITIL) is a library of books that describe best practices for infrastructure management. ITIL is a registered trademark of the U.K. Government's Office of Government Commerce (OGC).

ITIL is a framework and the models show the goals, inputs and outputs, and general activities of the various processes. Some of the most popular books for ITIL are:

- Service Support—Processes required to support IT services

  Incident management, problem management, configuration management, change management, release management

- Service Delivery—Processes required to deliver IT services

  Service level management, financial management for it, capacity management, service continuity management, availability management

To learn more about ITIL refer to:

```
http://www.itil.org/itil_e/index_e.html
```

SPICE

Software Process Improvement and Capability dEtermination (SPICE) is a major international initiative to support the development of an international standard for software process assessment.

The project has three principal goals:

1. Develop a working draft for a standard for software process assessment.
2. Conduct industry trials of the emerging standard.
3. Promote the technology transfer of software process assessment into the software industry world-wide.

The benefits for the software industry are supposed to be:

- Software suppliers will submit to just one process assessment scheme (presently numerous schemes are used).
- Software development organizations will have a tool to initiate and sustain a continuous process improvement.
- Program managers will have a means to ensure that their software development is aligned with and supports the business needs of the organization.

The hoped-for benefit for the purchasers of software is that purchasers will be able to determine the capability of software suppliers and assess the risk involved in selecting one supplier over another.

To learn more about SPICE refer to:

http://www.isospice.typepad.com/isospice_spice_project

ISO 900x

International Organization for Standardization (ISO) is one of the first process standards to be applied to software. It previously mainly applied to manufacturing processes. IT departments are often required to comply as a result of organization level commitment to ISO 900x.

The ISO 900x:

- Measures the degree to which an organization *says what they do, and does what they say*:
  - Examines the ability to closely follow documented process
  - Does not necessarily imply that those processes are any good
- A total quality management (TQM) approach:
  - Great for organizations that repeat the same thing with little variation
  - Assumes risk and quality can be controlled by repetition

To learn more about ISO 900x refer to:

http://www.iso.ch/iso/en/iso9000-14000/understand/inbrief.html

Six Sigma

Six Sigma is a statistical concept that measures a process in terms of defects. It is also a methodology consisting of phases, tools, and techniques that improves an organization's processes in a way that can measurably impact customers and the bottom line:

- Six Sigma represents the chance that the measured value of a specified item is outside acceptable customer tolerances:
  - Defined as <= 3.4 deviations per million opportunities to measure (DPMO)
  - 99.99966% pass rate (sometimes known as *5-nines* quality)
- Six Sigma is also a business philosophy with a goal of continual process improvement to reduce the costly defects that exist inside an organization. Six Sigma projects follow a life cycle that is designed to ensure that an organization does not jump straight to a solution without first understanding the underlying problems, processes, root causes, and supporting data.

There are three process variants:

- Define, measure, analyze, improve, control (DMAIC) is used to improve performance with existing processes.
- Design for Six Sigma (DFSS) is less widely used and has no universally accepted life cycle.
- Define, measure, analyze, design, verify (DMADV) is the most common variant used for process definition.

Some of the main terminology and concepts in Six Sigma are:

- *Critical to quality* (CTQ) is a subset of features that are critical to the customer's perception of quality.
- *Voice of the customer* (VOC) is the stated or unstated customer needs or requirements.
- *Black belt certified Six Sigma* is a subject matter expert who mentors or manages Six Sigma projects.
- *Tollgate* is a formal checkpoint between phases of the methodology.
- *Quality audit* is a scheduled review of process and project.

To learn more about Six Sigma refer to:

```
http://www.motorola.com/motorolauniversity.jsp
http://en.wikipedia.org/wiki/Six_sigma
```

CMMI

Capability Maturity Model Integrated (CMMI) is a framework that describes the key elements of an effective software process. CMMI provides an evolutionary improvement path from an ad hoc, immature way of doing business to a mature, disciplined, controlled process. CMMI guides software organizations that want to gain control of their activities for developing and maintaining software and to evolve toward a culture of software engineering and management excellence.

The success of the Capability Maturity Model (CMM) spawned a number of other engineering-related process improvement models, such as CMMI. CMMI was developed at Carnegie Mellon University's Software Engineering Institute (SEI). The development of the CMMI was sponsored by the U.S. Department of Defense to provide a means for assessing the capability of software development organizations working on Department of Defense contracts. Since then, the government has reduced its funding to a small fraction because the private industry has recognized the return on investment (ROI) potential.

CMMI provides a model that can be applied consistently across many parts of an organization, lowering cost for training, implementation, and assessment, as well as reducing redundancy and complexity.

CMMI has five levels of maturity:

1. *Initial*—The software process is characterized as ad hoc, and occasionally chaotic. Few processes are defined, documented, or used consistently. Success depends on individual effort and heroics.
2. *Managed*—Basic project management activities are established to plan and track cost, schedule, requirements, and changes. Sufficient process discipline is in place to document and repeat earlier successes on projects developing similar applications. Documented processes are used on all projects, though each project may use different processes. Basic measurements are collected and used to manage projects and process improvement.
3. *Defined*—Processes for both management and engineering activities are documented and integrated into a standard, tailorable software process for the organization. All projects use an approved, tailored version of the organizational software process for developing and maintaining software. More sophisticated measurements are collected and used to manage projects and process improvement.
4. *Quantitatively managed*—Both the software process and work products are quantitatively characterized and controlled. Measurements of software process performance and product quality are collected and used to quantitatively manage projects and improve processes.
5. *Optimizing*—Continuous process improvement is enabled by quantitative feedback from repeated process execution and from piloting innovative ideas and technologies in response to changing business drivers. Processes are statistically controlled and include defect prevention measures.

Studies show that less that 5% of organizations are rated at level 4 or 5.

To learn more about CMMI refer to:

http://www.sei.cmu.edu/cmmi/cmmi.html

RUP

The Rational Unified Process (RUP) is a framework for improving software development effectiveness that helps:

- Enable organizations to achieve better results more predictably.
- Provide a set of proven practices for improving effectiveness.
- Link development activities to delivering stakeholder value.

- Reduce cost of improving effectiveness by leveraging the successful experiences of others.

Figure 1-7 shows the RUP framework.

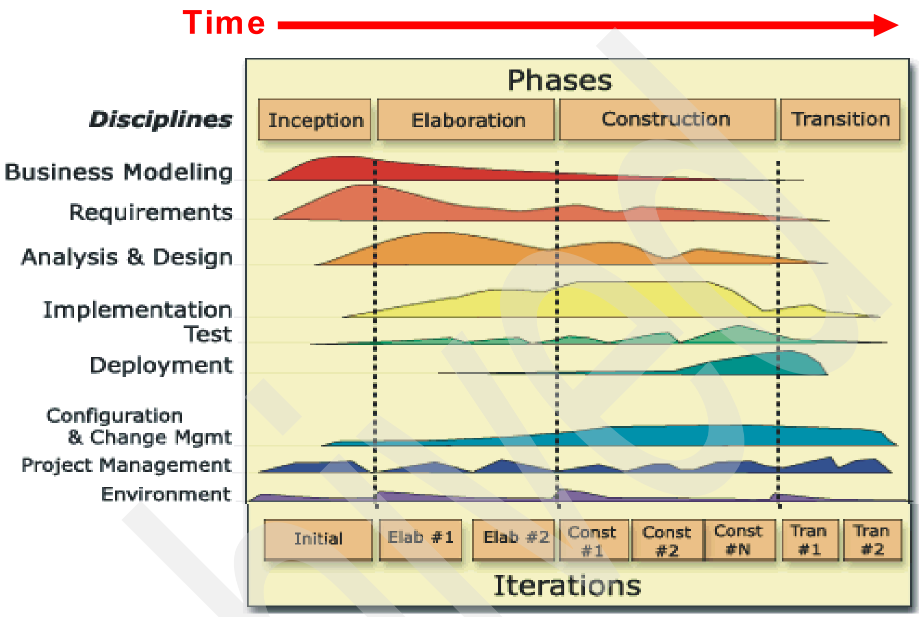


Figure 1-7 RUP framework

The time aspect of the process is enacted through phases (Table 1-1), iterations, and milestones (end of phase objectives). Progressing by meeting milestones helps minimize wasted resources because they are allocated only on a firm basis.

Table 1-1 RUP phases

Phase	Objectives
Inception	Mitigate *business* risks; gain agreement on overall scope. ► Vision, high-level requirements, business case
Elaboration	Mitigate *technical* risks; agreement on solution approach. ► Baseline architecture, most requirements detailed, high-level design
Construction	Mitigate *logistical* risks; apply approach. ► Working product, system test complete

Phase	Objectives
Transition	Mitigate *deployment* risks; roll-out solution into production. ► Stakeholder acceptance

The phases of RUP were chosen such that phase boundaries correspond to significant decision points in the life of a project. For example, at the end of inception, enough work has been done to capture the problem to be solved, and a vision of the system has been developed. An initial set of risks has also been identified and evaluated. Based on this information, a decision is made whether to fund the project. Similar decision points correspond to the end of elaboration, construction, and transition.

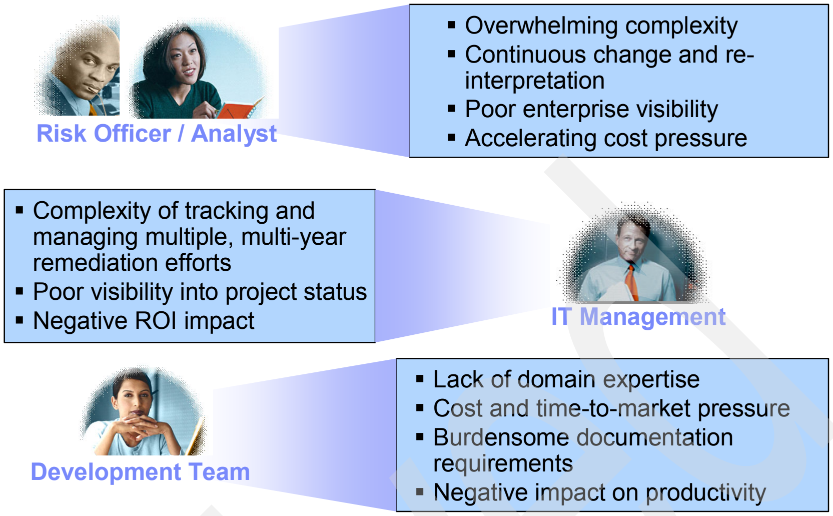
Milestones help you to assess the progress of a project at key points. Management can use these to establish clear criteria from which to decide the course of a project. They provide opportunities to change course. Phases contain iterations that yield executable results.

To learn more about RUP refer to:

http://www-306.ibm.com/software/awdtools/rup/

Typical compliance challenges and concerns

Compliance initiatives often require changes in process and operational activities that impact the day-to-day work of the entire organization. In order to minimize the overhead that these changes can imply, it is important for companies to realize value from compliance. Many companies are overwhelmed by the scope and complexity of compliance (Figure 1-8) and find it necessary to turn to consultants and outside agencies to solve their problems.



Risk Officer / Analyst

- Overwhelming complexity
- Continuous change and re-interpretation
- Poor enterprise visibility
- Accelerating cost pressure

IT Management

- Complexity of tracking and managing multiple, multi-year remediation efforts
- Poor visibility into project status
- Negative ROI impact

Development Team

- Lack of domain expertise
- Cost and time-to-market pressure
- Burdensome documentation requirements
- Negative impact on productivity

Figure 1-8 Challenge for compliance: visibility and cohesion across domains

Compliance, however, is an ongoing process and one that companies must come to terms with by establishing appropriate controls and automation in order to make compliance a productive and value added process. Tools that bridge the gap between the varying roles of the organization and facilitate information sharing and communication are necessary to make compliance something that provides benefit to its participants.

Other challenges are in the areas of:

- *Security*—Who can access artifacts? Are the roles valid? Are the employees in the roles?
- *Auditability*—Who made changes to artifacts? Are they authorized? Can they be tracked?
- *Data and application access*—Authorization, execution, acceptance
- *Monitoring of applications in production*—Operation changes/authorization, exception handling/tracking
- *Authorized software licenses*

Some of the problems that businesses seeking compliance have to address are:

- Weak documentation and assessment of internal controls
- Lack of sufficient appropriately qualified/trained resources, both at a corporate level and even more so in business units

- Content/records management and document management strategies not keeping up with heightened demands
- Dependence upon spreadsheets
- Loose *off the ledger* audit trails (insufficient documentation, data consistency, and controls), especially on software development processes and packaged application customization processes

  An example is lack of a process in place to track that functional requirements are implemented in the applications that automate business rules.
- Lagging IT infrastructure
- Unclear accountability structures
- Lack of compliance policy for the retention of key information and process controls when operating the business
- Limited ability to ensure proper destruction
- Inability to prove records were not falsified
- Unable to quickly and easily retrieve records upon request
- Storage media may be inappropriate for retention needs

In addition, here are some examples of the types of compliance concerns that may be expressed. They are shown in no particular order or priority. It is important to note that this list is not exhaustive, as it will vary from problem domain to problem domain:

- How can I manage the multi-year effort required to implement the regulatory changes across all of my systems, controlling the costs, budget, delivery, functionality, and integration of all of my systems?

  This is a complex problem requiring portfolio management expertise, integrated with program and project management.
- How will I demonstrate a linkage between the business processes I develop and the rules that my processes satisfy?

  Auditing specialists are continually looking for something that ties the business back to the language of the legislation, regulations, standards, or policies. This is often accomplished through the use of a document or position paper that describes a plan for achieving compliance. The reality is that many times these documents are drafted and forgotten, particularly on initiatives that run longer than six months.
- How will I document my existing business processes and systems as they are implemented today?

  Certain legislation demands that executives have a full accounting of how the business operates. As a component of any remediation effort for a business, it is now necessary to document how the business operates, and where the key points of control exist.

- Auditors and inspectors can ask you how you did things before you changed them.

  Auditors and inspectors are also keenly interested in demonstrated control over changes to the business. They will want to see that you have “a process for modifying business operations and automation.”

- How will I assess the compliance gap between my current business operations today and where I want to be tomorrow?

  To transform the business, one has to leverage the *As-Is model* of business operations into a *To-Be model* that fully complies. However, one cannot blindly construct a compliance process without having a rudimentary understanding of what the costs of changing the business processes will be.

- How will I transition my automated systems from their existing implementation to a new implementation?

  To achieve compliance, it is often less expensive to replace an existing system than to redesign the old system for something it was never intended to do. Gap analysis has to be performed between these older systems being replaced, and the newer systems you may be considering. Homegrown systems that provide competitive advantage may have to be redesigned. Finally, you will have to reintegrate your newly selected systems to your existing systems or modify the functionality of your homegrown systems to close the loop.

- How can I demonstrate that the business processes I have employed are actually being adhered to as documented?

  Although the executive committee defines how things are done, someone has to monitor the business to ensure adherence to new operating rules. Auditors will be seeking evidence that the newly defined rules have been implemented. Simple implementation of appropriate measurements and metrics can solve this problem easily.

External business factors impacting compliance initiatives

The business environment can have a direct impact on a company’s ability to invest in compliance initiatives. The reasons for this are simple. Businesses have a limited amount of financial and human resource capital to work with.

Here are some examples:

- *Competitive landscape*—Business conditions affect budget. Budget in turn affects a company's material ability to implement changes to business processes and systems. If the profitability margin is small, containment of cost can be imperative.
- *Consistency of operation*—Consolidation is always a challenge. Integrating companies allied through mergers and acquisitions (M&A) poses a set of problems that compound compliance. However, companies often have to demonstrate that the M&A has not impacted their ability to maintain compliance, and are still able to deliver high-quality products and services. Furthermore, companies may be under additional financial and operational pressure to manifest changes quickly. Finally, companies involved in M&A must establish standard operational procedures for all aspects of the business to streamline the business or be compliant with regulations, standards, and policies.
- *Outsourcing of application development*—Another common trend is that of outsourcing application development. On the surface this may appear to be a cost-saving measure. However, if the application is designed and developed using technology foreign to the contracting company, a full-time employee (FTE) from the development company may have to be assigned to maintain the application.

  Some companies view utilization of outsourcing as an approach to defer the accountability for compliance, but there is no escape from this responsibility. Many regulations state that the company producing the product or service consuming the application is still accountable for compliance regardless of who they hire or contract with.

  A simple solution to this problem is to require standard tooling. By requiring contractors to use the contracting company's standardized tooling, the completed application designs and artifact traceability will be consumable by the contracting company when the final application is delivered. Ideally, you would want these tools to be industry standard market leading tools that are either #1 or #2 in their respective markets.
- *Maintenance of COTS customizations*—Management of commercial off the shelf (COTS) applications is just as important from a compliance standpoint as green field systems development.

  In many cases the applications created today are *acts of aggregation* more than acts of creation. Significant work must be performed to understand the gaps between the needs of the business, the capabilities of the products under consideration, and the performance and procedural requirements that must be satisfied.

- *Manifesting cultural change*—Creating ground swell for the use of new technologies to improve the compliance profile of the business is a challenge in itself. In order to create ground swell in an organization, you must offer technologies and solutions viewed as attractive by your technical community.

  You must also invest in proven solutions that have a demonstrated ability to deliver. Ultimately, your business community should feel these solutions simplify their jobs, and not complicate them. Experience has shown that managers that select tooling that enhances the skills and marketability of their technicians vastly improves the success of projects, and improves overall morale. Employees tend to feel that this is an investment in them, as well as in the company.

2

Compliance guidelines

In this chapter we discuss the relationship between compliance management and the software development process, and the importance of having a compliant software development process to achieve a solution for compliance.

We examine the desirable features of a software development processes and we introduce some of the key considerations when examining compliance processes. Finally, we analyze practical control strategies for enforcement of the software development process.

Why business compliance often requires software development compliance

As we stated in Chapter 1, “A discussion about compliance” on page 1, regulations, standards, and policies rarely define *how* these rules must be satisfied. So how are rules policed and what mechanisms exist to ensure that companies remain faithful to the spirit of the rules they must conform to? The primary mechanism used to ensure compliance is internal and external audit.

In order to demonstrate compliance, organizations must always be prepared to respond to compliance audits.

One easy example: In most large-scale regulated companies where risks are high—for example, to life, health, and welfare—significant investment is made in the establishment of standard internal auditing procedures. In many of these companies entire departments may be dedicated to the execution of internal inspection to ensure conformance. Why so much investment? This is because representative enforcement agencies from the respective regulatory authorities are entitled, in many cases, to perform unannounced audits. Therefore, for companies to protect themselves and be adequately prepared to demonstrate regulatory compliance, organizations must always be prepared to respond to a regulatory compliance audit.

For a detailed discussion of compliance in today’s environment please refer to Chapter 1, “A discussion about compliance” on page 1.

Why do regulations, standards, and policies that seem to point only to the organizational level become a matter of software development? To help us answer this question you should consider two important relationships that are relevant to software applications: business operations and the governance of the business.

Compliance in automated business transactions

Today's applications help automate virtually every aspect of the business operation. As a result, organizations may only be as secure, auditable, and transparent as the applications that support them. If you analyze a typical business transaction, you will see that software applications implement, automate, and enforce the business rules and policies of an organization. Most of the core business transactions validate and execute modifications of customer data, product data, generate reports, and so forth, all utilizing the implemented business rules and policies of the organization. Auditors understand the critical link between business transactions and the software applications that execute these transactions. Therefore, to determine an organization’s compliance an auditor must evaluate the integrity of processes used for the implementation of these business rules and policies, and must be concerned with the software applications that support these business transactions.

In this sense, compliance with regulations, standards, and policies quite naturally leads to business application compliance. Compliance audits are thus redirected to software applications and packaged application customization. Businesses may recognize the need to carry an inventory of their systems, to be able to determine which systems may be impacted by change in regulations, standards, and policies. We will explore this idea a little further in a subsequent section on governance of software development.

There is added complexity to this problem to be considered. The systems for which a business has no source code, nor the ability to modify a packaged application, must still comply. Furthermore, consider the need to implement manual business processes that institute checks and balances against business decisions that are recorded in an automated system. These controls must be accounted for as well.

Governance versus compliance

As stated earlier, compliance at the organization level requires that businesses critically evaluate all of the systems potentially impacted. For example, any given financial statement (such as an annual report) may require aggregation of data from multiple financial systems or financial applications to derive and accurately reflect the correct financial condition of the business. To proactively address this type of challenge the organization must implement a process solution that encompasses the organization, generally referred to as *governance*.

We begin this discussion by defining the general concept of governance:

> *Governance is a process that establishes chains of responsibility, authority, and communication, to empower people, as well as the measurement and control mechanisms to enable people to carry out their roles and responsibilities.*[1]

Utilizing this definition and applying it to the realm of software development, we arrive at the following:

> *Software development governance is the alignment of an organization's business strategy and direction to implement appropriate processes, with sufficient chains of responsibility, authority, and communication to empower people, by implementing appropriate measurement and control mechanisms that enable them to carry out their roles and responsibilities.*

[1] *Governing the business process of software and systems development*, by Murray Cantor and Rachael Rusting, IBM Corporation, Liz Barnett, EZ Insight Inc.

The purpose of good software development governance is to address these goals:

- Define and implement a structure within which to execute organization management and administration.
- Provide active direction, periodic review of interim results, and identification and execution of adjustments to ensure achievement of the planned outcome (which contributes to the success of the overall business strategy).

These goals are achieved through a combination of the right personnel, an effective structure for management and oversight, and a set of program roles and responsibilities. It also requires—as our previous general definition of governance outlined—appropriate measurement and control mechanisms to empower people to monitor and improve the process. Roles and responsibilities should be defined and structured with the required outcomes of the organization in mind, and to fit within the management philosophy and enterprise approach. Thus, each person is invested in a role and must have a clear understanding of the decision rights, responsibilities, and objectives assigned to them by management, and the relationship their role has with all the other roles within the organization.

The concept of governance has multiple business and governance components (Figure 2-1):

- *Roles definition*—An important aspect of governance is assigning specific decision-making authority and responsibilities to each role.
- *People empowerment*—It is necessary to have a clear and well-understood assignment of people to specific roles. This is not to say that a given individual cannot function in multiple roles, provided that the responsibility of these roles do not conflict.
- *Structure definition*—Establish clear chains of responsibility and communication to facilitate development coordination.
- *Policy*—It is essential to define, document, and clearly communicate management principles and decision-making criteria.
- *Measurement*—It is essential to ensure that appropriate measures are placed throughout the process at multiple levels, in order to evaluate the effectiveness of the structure, process, product quality, and alignment of the process to organizational goals:
  - Organizational measures should evaluate items such as estimation maturity, value creation, and development maturity.
  - Project or program measures should evaluate items such as results, variance reduction, program or project financials, and product quality.

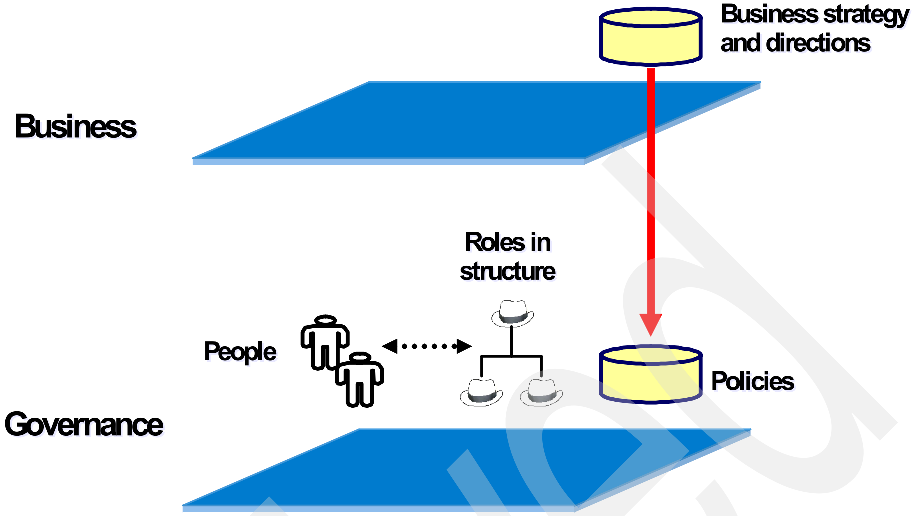


Figure 2-1 Business and governance components

If this is what governance is, then what is compliance? Here is our rather pragmatic definition:

> *Compliance is the set of deliverables, processes, and documentation necessary to satisfy a company's interpretation of a given regulation, standard, or policy.*

Why does our definition contain *the company's interpretation of a given regulation, standard, or policy*? As we have stated earlier, many of these rules are non-directive. In other words, they contain no actionable plan for a company to follow. Consequently, the companies are forced to interpret these rules and devise their own action plans.

These action plans are actually a minimalist form of governance. The reality is, that to implement compliance well, companies should utilize good governance first. Good governance ensures the establishment of consistent process, decision rights, and authority with which to conduct business, as well as the necessary measurements and controls required to track the process and improve its operation and product outputs. After all, what is the first thing an auditor or inspector asks for? Your procedures.

Implementing software applications is a kind of governance

If you are not yet convinced that your company is actually doing governance and compliance, consider this: Many companies currently employ a buy versus build strategy to meet their business goals and objectives. Because the motivational factors for the selection of a packaged applications tend to stem from business needs, implementation of such packages is a form of governance and compliance over the process that you want to automate (or that is automated in the case of package replacement). After all, the criteria typically contained in a request for information or proposal (RFP or RFI) for the selection of such packages is often driven by the need to ensure alignment between the organization's existing process and the behavior of the package. In the case where no package exists to meet these criteria, companies will develop their own package to address the appropriate business need. Incidentally, the RFI/RFP process is in itself a governance of acquisition process.

However, this is not software development governance. This is a form of governance over the business process that the package satisfies. Software applications can be the basis of organizational business activities. Through such an approach the company aims to enforce business policies into organizational activities, to control the execution of these activities, and to review the results. You can see the similarity between this and the previous definition of governance that outlines chains of accountability, responsibility, measurement, and control.

Using information technology components (applications and tools), organizations can automate policy enforcement tasks and facilitate control and review tasks easier, faster, and less expensively than using a manual process.

Compliance is also a business requirement that is often a criteria found in RFPs and RFIs, because it is the intention of the organization to enforce regulations, standards, and policies, and to establish controls associated with them using an automated mechanism that also reduces compliance cost.

Information technology applications allow organizations to:

- Enforce policies in application processes, including those related to regulatory compliance.
- Establish control mechanisms throughout applications processes to guarantee conformance.
- Facilitate information recovery and report generation to help management oversee and gather information needed to demonstrate compliance.
- Share integrated information among all involved roles to consistently distribute data across the organization.
- Ensure information security issues, control access rights and information confidentiality.
- Automate tasks to reduce cost and reduce the likelihood of human error.

Hopefully, you should now see the similarities between packaged application implementation, software development governance, and compliance. If so, then you understand why we advocate for the use of formalized tools and processes for software development governance.

We make this distinction now, because you cannot implement compliance well without some governance. Therefore, when we outline our solution for compliance, you will understand why our process includes some process design components from governance, in addition to documentation, artifact tracking, and accountability as in compliance.

Figure 2-2 shows our area of interest, which is compliance-driven development, and those parts of business controls and IT governance related to software development.

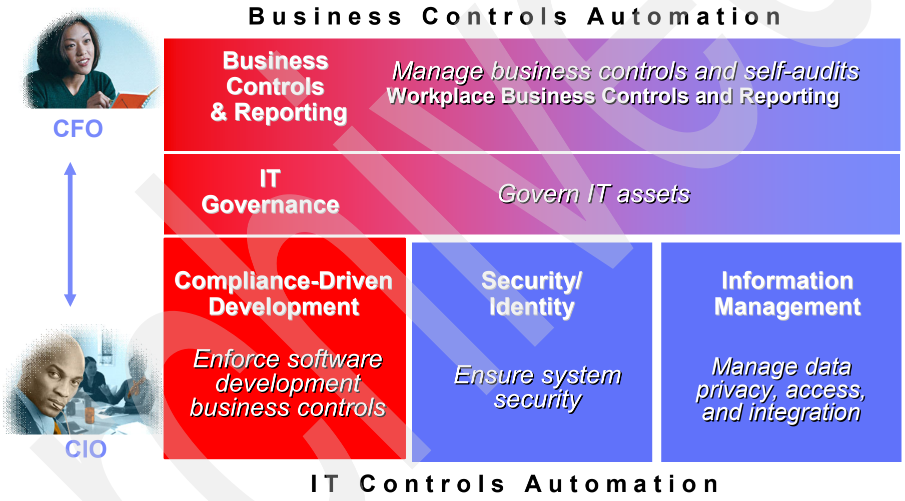

Business Controls Automation
CFO
Business Controls & Reporting
Manage business controls and self-audits
Workplace Business Controls and Reporting
IT Governance
Govern IT assets
Compliance-Driven Development
Enforce software development business controls
Security/ Identity
Ensure system security
Information Management
Manage data privacy, access, and integration
CIO
IT Controls Automation

Figure 2-2 Objectives for software development compliance

Compliance process requirements

What requirements should such a process and automation solution contain? In this section we explore the high-level components, processes, automation, and traceability necessary to manifest a compliance solution.

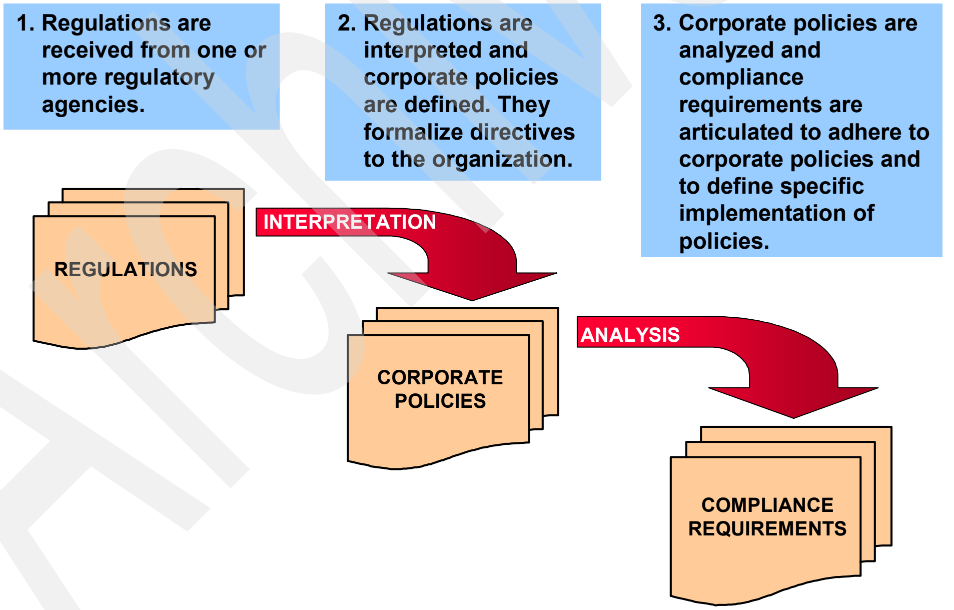
A satisfactory compliant software development process must satisfy several generic goals related to compliance:

- Show adherence and assure implementation of all policies established at the organizational business level, generally referred to as policy provisioning.

- Facilitate information extraction related to the demonstration of compliance.
- Gather sufficient information to respond to internal controls and auditors demands.
- Attend to compliance cost reduction, based on best practices and supported tools.

Satisfying provisioned policies

Here is an example of how a company will provision a policy: Organizations operating in a regulated industry generally formulate a policy that describes how their organization intends to respond to a pertinent regulation or act. These organizational policies can be formal directive specifications to align organizational activities with business needs. They are the result of interpreting the meaning of regulations for this organization. After corporate policies are defined, they are analyzed and compliance requirements are articulated to the various business units for implementation as process. In the event that these processes are automated, these policies must be implemented or modified in the existing target applications (Figure 2-3).



1. Regulations are received from one or more regulatory agencies.

2. Regulations are interpreted and corporate policies are defined. They formalize directives to the organization.

3. Corporate policies are analyzed and compliance requirements are articulated to adhere to corporate policies and to define specific implementation of policies.

Figure 2-3 Evolution from regulation to compliance requirements

During an audit it may be necessary for such a business to substantiate that it has taken appropriate action in the implementation of corporate policy to respond to a given regulation or act. Based upon requests from auditors or inspectors, the organization may have to demonstrate that appropriate controls have been implemented, by following the chain of traceability from corporate policies to structured application requirements (that is, traceability from corporate policies to software artifacts) (Figure 2-4).

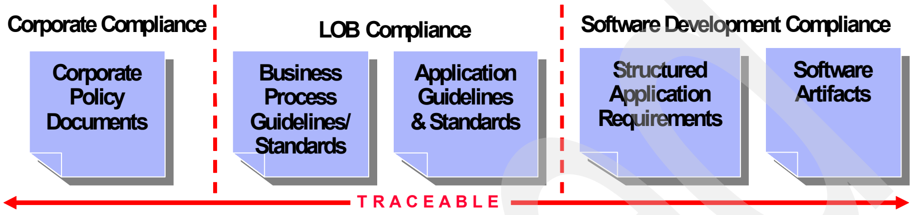

Corporate Compliance
LOB Compliance
Software Development Compliance
Corporate Policy Documents
Business Process Guidelines/ Standards
Application Guidelines & Standards
Structured Application Requirements
Software Artifacts
TRACEABLE

Figure 2-4 Traceability from corporate policies to software artifacts

It is important to note, however, that implementation of such traceability is not sufficient. Auditors and inspectors are also just as concerned about how the artifact chain of traceability was established. Thus, software development governance is designed to ensure that the process is followed and that all actions are recorded to substantiate compliance.

A compliant software development process should institute good governance practices in order to enforce proper authorization points and execution of the critical controls with the process. See "Points of control" on page 46 for a more detailed description.

Demonstration of audit data

The availability of audit data is often a key concern for companies operating in today's regulated environment. Companies that have not grown up in a culture of audit are often at a loss to locate important artifacts, such as system or regulatory requirements, release to manufacture (RTM) records, or design documents. Failure to deliver these items can mean the difference between passing an audit or being sited. For some inspectors, the speed with which important audit data can be returned is a reflection on the compliance capability of the organization or project being audited. Consequently, being able to quickly locate, retrieve, and display or print any artifact related to a system's design puts the projects being audited in a better light.

Any data provided must be consistent with all other artifacts presented, reflecting a consistent strategy for artifact linkage. Some rules go so far as to describe the data requirements for non-repudiation of a decision.

Examples of the type of data to be recorded include items such as:

- Who did what activity and when it was done
- The action that changed the record and its final state
- The name of each changed field, its old value, and its new value
- Deltas for multi-line text edits
- Reason or explanation of change

Compliance cost and controls

Given the requirement to substantiate due diligence, most organizations implement a process to ensure artifact linkage and appropriate controls. In a perfect world, there would be no need for any new processes. Organizations would only have to focus on their core business mission for product or service delivery, and as a consequence of delivering the product or service they would automatically comply with all regulations, standards, and policies as a matter of course. Unfortunately, it does not work that way. Under pressure to implement new processes and controls, inevitably many companies choose to implement these processes manually. This is due to the perception that manual processes are the easiest to implement, as well as the least expensive method for implementation.

In reality, this perception is true but only in the short term and only for the establishment of the process. Taking a more comprehensive view of the needs of control processes reveals a significant cost for implementation. Typically, manual implementers only examine the requirements to document the process, provide training to the practitioners, and periodic checks for adherence to the now documented process.

Often, what is overlooked is the need to periodically review and change the process to document any additional controls determined necessary. Another step that is frequently overlooked in the establishment of manual processes is the need to record and document effective control over all of these manual activities, and to extract the information required to demonstrate compliance.

Using manual processes in the long run is likely to fail due to the requirement for rigid adherence. Consequently, manually implemented processes are likely to break down. This is due to the level of effort required to ensure strict monitoring, policing, and enforcement by management. Often, the feeling is that having the process is enough. However, without an effective method for retrieving the process metrics from a manual process, what good is it?

Finally, there are additional hidden costs in the implementation of manual processes. Examples of these hidden costs are:

- Increased workload for practitioners impacted by the process change

- Loss of productivity due to overhead introduced by the additional process constraints
- Increased backlog of transactions impacted by the process change
- Degradation of morale among those with excessive workloads
- Failure to completely follow the prescribed process as documented
- Bypassing of the documented process, threatening the integrity of the established controls

Naturally, these costs impact all areas of the business, including the software development process itself, and potentially the integrity of the controls implemented.

One should consider the documentation of the process as only a first step. If implemented properly, compliance can actually lower overall operational costs by streamlining business operations, ensuring complete control over the task, establishing accountability for product or service quality, and automatically establishing artifact traceability and metrics gathering.

In order to reduce these process costs, two main items should be considered (Figure 2-5):

- Adoption of best practices in the target domain
- Use of automated workflow, also known as tool-directed behavior (TDB)

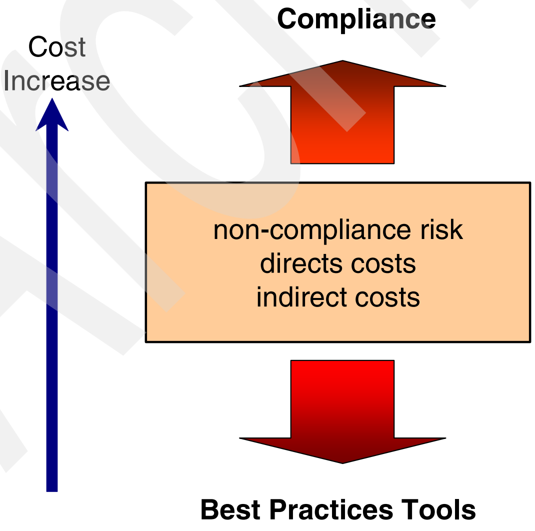


Figure 2-5 Best practices and tools help to reduce compliance costs

Adoption of best practices

The term *best practices* is really a misnomer. In reality there are no *best* practices, just the most current good practices. After all, software development methodology changes and improves over time. Thus, models and frameworks based on industry best practices really serve to document and improve organizational processes as a key element to reduce cost, time, and risk and improve the operational efficiency of the process of software development. In the complex world of information systems, best practices are useful as a strategy to ensure that organizations implement good governance to reduce costs related to the establishment and maintenance of organizational processes.

As stated previously, methods based on best practices are in a state of constant evolution, taking advantage of other demonstrated current good practices and lessons learned as a result of application of the method in other organizations. Organizations that adopt commercial methods are incorporating an important set of reusable assets that were created as a result of experiences shared with other organizations.

From a compliance perspective, use of commercial methods improve the likelihood of surviving an audit. This is because commercial methods provide organizations with additional guidance that regulations, standards, and policies do not. Use of commercial methods can aid organizations in establishing a commonly held operational model that is likely to be found elsewhere within their industry. Consequently, it is probable that auditors moving from company to company are likely to find the same kind of probative elements in similar organizations. In this sense, best practices can act as an accelerator in the process of demonstrating compliance.

Finally, one of the most important aspects of leveraging commercial methods is their extensibility. Vendors of commercial methods recognize the need for extensibility, and as a consequence design their products to give customers and other vendors the ability to modify their existing methods. Rational provides support for this capability through support for process plug-ins. Plug-ins allow process components to be designed as reusable assets, which can inherit linkages to and from related artifacts in the method, as well as other plug-ins. Plug-ins also provide the added advantage of supporting extensibility, by allowing customers to augment the existing process, and still provide upward compatibility when a new version of the method is released form the vendor.

Generally, valuable characteristics of commercial methods are:

- Based on industry-proven best practices
- Practical approach in processes
- Adaptable to organization needs
- Supports extensibility through plug-in style extensions

Workflow automation for tool-directed behavior

Software development typically involves repetitive executions of tasks related to a predefined process. These repetitive tasks involve many different IT staff members, over many hours, days, weeks, and even months of effort. As we had alluded to earlier, implementation of manual processes can exacerbate the costs of performing these tasks. However, through the use of workflow automation, not only can these costs be reduced, but companies can actually begin to manifest cost savings.

Workflow automation improves organizational capability by reducing the overhead of manual tasks and controls. Workflow automation also enforces adherence to the defined process by practitioners, by having these tools direct the behavior of each practitioner, so that the integrity of the process is enforced by the tools. The term we use to describe this function of automation is *tool-directed behavior*.

Because the execution of the defined process is nearly guaranteed, one can implement whatever control points, measures, authorization points, and restrictions deemed necessary to comply with the defined process or commercial method. This nearly eliminates the possibility of malfeasance in the execution of such processes. Finally, cost savings are often realized when automating processes, because the use of a consistent communication channel facilitates increases in overall productivity.

When evaluating the use of tool-directed behavior to ensure process adherence, several considerations must be examined to select the best tools to meet your organization's compliance objective:

- *Tool stack integration*—To reduce the risk of inconsistent artifact linkage across the development domains and to provide common access to all development assets, it is desirable that the tools are designed to interact directly with each other.
- *Configurable*—To enforce software processes and minimize the overhead of manual execution of control tasks, the tools should be configurable to match your commercial method or in-house method design.
- *Flexible*—The stack of tools should not create constraints on the process design due to limitations in process support.
- *Traceability and reporting*—The tools should provide the ability to conveniently expose the chain of traceability or the metrics associated with evaluating program or project quality.

Considerations for compliant process design

As we have established earlier, availing oneself of appropriate software development process documentation would be a good start to being *audit ready*. However, having a well-documented process is not much good if the process is inherently flawed. To this end, we offer the following considerations for compliant software development process design. They are in no particular order:

- Software development process documentation
- Separation of duties
- Workflow and artifact approvals
- Audit trail
- Electronic signature
- Authentication and authorization
- Points of control
- Reports
- Metrics

Let us examine each of these key considerations in more detail.

Software development process documentation

Among the myriad of tasks that an auditor or inspector could choose from, typically the first is acquainting herself with your software development process. Therefore, a prudent inventory of your compliance assets should include documentation of your existing process. Ideally, this documentation will not be in the form of a document, but as a dynamic Web site that hyperlinks all pertinent materials together.

The objective of such a solution allows practitioners the ability to self-enable on an as-needed basis. This frequently occurs when their position demands new skills, tasks, or artifacts to be created. Recall, as we discussed earlier, that this solution should be a commercially available set of extensible methods. This type of structure is also advantageous to an auditor or inspector because their investigatory style is likely to require a need to understand the way in which an individual might work in a given role, and all of the artifacts that role might produce or consume.

Separation of duties

When constructing a robust process, in a set of related activities, separation of duties is a policy that consists of assigning responsibility to different people for each major activity in the flow of execution. That is, each critical activity must have a different person responsible for the creation or approval of a given artifact. Consequently, use of authentication is an unavoidable matter to be implemented in the separation of duty policy.

Common uses of separation of duties policy include:

- Assure that no important activity can be bypassed in a process.
- Mitigate the risk of consecutive errors in process execution.

Approvals

Approvals ensure that specific actions that occur throughout the process are appropriately sanctioned. Approvals are put in place to prevent the introduction of unapproved changes in the target business system or software, and thus ensure accountability. Approvals are also required from a business perspective to allow the execution of IT-related projects at the portfolio level. The term used to describe this accountability is *non-repudiation*. In other words, the elevated accountability through approval eliminates the possibility that someone may repudiate or deny having authorized one or more activities throughout the process. These approvals may be paper-based, or electronic, utilizing handwritten or digitized electronic signatures or perhaps a simple and unique user ID and password combination.

Each organization has to define and tailor the approvals to align with the compliance requirements of the development or portfolio management process. Approvals may be added or removed in accordance with the organization policies. We can differentiate between two kinds of approvals:

- Financial approvals—These approvals are related to funding.
- Technical approvals—These approvals are related to the design or construction of a system.

Financial approvals

Financial approvals in the software development process may seem foreign to some, but may be a necessary component of a compliance process to substantiate due diligence in the eyes of an auditor or inspector. The reasons for this tie back to our earlier discussions regarding demonstration of intent. Funding may indicate the level of executive and management commitment to a compliance effort.

Organizations implementing appropriate governance processes for changes to information technology infrastructure recognize the need for prioritizing requests for these changes. Furthermore, implementation of a comprehensive approach to managing these requests improves the likelihood that the organization will receive a good return on investment related to these expenditures.

Earlier we established that the need to implement new processes and business controls in response to changes—particularly if these controls are automated—necessitates the need for modifying several business systems simultaneously. Because the impact of changing multiple systems concurrently is likely to have a significant impact on the business, a careful accounting of cost, schedule, and effort must be an integral part of the systems management process. Ideally, this process should begin by evaluating the systems inventory. The processes should also include an estimation of time, effort, and cost by constructing project plans based on previous experience in implementing similar changes to the impacted systems, and based on complexity and current good practices utilized by each organization performing the changes.

The combined assessment, along with these estimates, can then be evaluated together to present better information with which to make decisions about which systems to modify and when. Through the implementation of such a process, organizations ensure that they are adequately prepared to demonstrate compliance or, if remediation is needed, intent to take corrective action, or at least demonstrate the reasons for the pace with which adoption is proceeding based on currently available resources, schedule, and effort.

Technology approvals

Technology approvals are typically performed after the approval of funding and the initiation of a systems modification effort. These controls are implemented to ensure that adequate technical oversight was applied to the implementation of automated business controls, and to ensure alignment of these controls with the strategic direction of the application to the those of the business. For example, a technical reviewer may examine whether the solution for a business control is designed to conform the organization's technical standards for service publication, if the component is part of a system design with service-oriented architectural style.

Software development audit trails

An audit trail provides a record of the changes made to a given work product. The purpose of audit trails is two-fold. Audit trails provide a reasonable assurance that all changes made to a given work product are recorded for the posterity of the organization. Furthermore, audit trails also serve to protect application stakeholders by tracking, with full accountability, any changes made to a work product.

An example of the type of content expected in a robust audit trail is described in the United States Code of Federal Regulations (CFR) by the Food and Drug Administration (FDA). Title 21 Part 11 describes what the contents of such an audit trail should contain.

The audit trail description for CFR Title 21 Part 11 includes:

- Who changed it—Identifying information for the person who performed the modification to the electronic record.
- When it was changed—Information describing the date, time, and time zone of the change.
- What action—The action that changed the work product and the process state.
- What fields were changed—Delineating the name of each field modified including its old and new value.
- Purpose—Some people even believe this requires an explanation of the reason for the change to the work product.

When utilizing such audit trails it is presumed that this information is never deleted. Because the nature of creating such audit trails is tied to an appropriate good electronics records management strategy, audited work products can never be directly deleted. Typically, systems implementing solid audit trails implement a strategy where an entry is made against the record indicating that a work product is flagged for deletion, and will subsequently be deleted in accordance with organization policies regarding record retention periods.

Electronic signatures

Electronic signatures, also known as *e-signatures*, provide approval capabilities for automated tooling environments. The purpose of an electronic signature is to implement an approval that will be made by one or more authenticated users with the intention of replacing a handwritten signature.

The three predominant mechanisms acceptable as an e-signature mechanism in use today include a unique user ID/password combination, biometric authentication (such as retinal scan, or thumb print mapping) or digitized signature analysis.

In all three cases, the intent of an electronic signature is to validate the credentials of the signer, and to ensure non-repudiation of any artifacts created, modified, or deleted by the signer. *Non-repudiation* is the term used to indicate that it is not possible for the authenticated user to refute performing the recorded action against the electronic record or work product.

Authentication and authorization

Closely related to the implementation of electronic signature is the need for organizations to implement a mechanism of authentication and authorization.

Typically, customers will look to identity management solutions that provide a single point of authentication for the enterprise. Such identity management systems should provide the ability to grant, manage, and enforce user access permissions, implement a standardized recertification process (to validate each user account is still needed for a valid business purpose), and the ability to quickly produce reports to help speed the preparation for internal audits.

Many of the identity management solutions available today are extensions to the open standard of the Lightweight Directory Access Protocol (LDAP). LDAP is an industry standard protocol used for accessing and managing information directories. Using LDAP, users can search for e-mail addresses and other information in a directory service and also update that information. One or more LDAP servers contain the data making up the LDAP directory tree or backend database.

The goal of identity management solutions is to protect sensitive information and prevent that information from unauthorized access.

Points of control

The purpose of implementing workflow is usually to ensure that those who participate in its execution conform to the various steps to produce output of consistent quality. As an additional mechanism to ensure accountability of product quality approval and gating procedures can be installed in the workflow throughout.

These gating procedures typically require an inspection step, whereupon the reviewer is required to indicate his satisfaction or dissatisfaction with the produced output. These control points are key events in the development workflow, where an action of approval has to be taken to allow the process to continue.

Careful consideration must be given to several items simultaneously when building an automated workflow solution. These considerations must include the establishment of those roles that participate in the workflow, the decision rights each role will be granted, and the points in the workflow where sufficient data is available with which to make a determination of the output quality.

Documentation and reports

One of the greatest challenges in producing documentation for software applications is keeping the documentation up to date. Just when you concluded putting the finishing touch on that document, it is out of date. The challenge is that documents and reports are useful mechanisms for demonstrating process compliance. Through reports, organizations can show information required for auditors in an easy-to-read and portable form. Reports are also essential for management to quickly assess issues and identify appropriate actions under changing business conditions. Once issues are identified, reports are also useful for monitoring the results of decisions and consequential activities taken. As a consequence, by having relevant reports on-hand, organizations can expect that the time required to complete an audit may be shortened.

To ensure that these documents and reports are timely, several actions should be taken from a governance or process perspective.

In most cases, work products created during software development activities can be documented through the use of automation, by leveraging the artifact definitions contained within their design tools and repositories. Automated documentation generation tools, which have the ability to reach into such repositories, can create robust and fully formatted reports containing even greater, painstaking detail than any human might wish to pursue.

Documents or reports regarding the governance process itself, or those related to business transactions, will have to be retrieved and generated from the related transactional systems under scrutiny by the inspector or auditor, in the same manner.

The second action that teams should consider is the institution of scheduled generation of these types of reports, with consistent archiving based upon aging. When implementing such an infrastructure, consistent and timely information can be on-hand, even in the event of a surprise audit. You should also keep in mind that these types of reports and documentation also have to be retired, based upon your company's records retention policy.

Metrics

As Lord Kelvin once said, "*If you cannot measure it, you cannot improve it.*" However, as many experienced scientists know, the moment you start observing or measuring something is the same moment that its behavior begins changing. As humans beings, we are very much influenced by both positive and negative feedback. Consequently, once you establish a set of criteria by which you will begin measuring a team, that is the moment their behavior is likely to change.

Formally metrics are numbers used as a measurement for comparing different items for trend, distribution, and aging analysis. The main objective of metrics is to enable management to take corrective action in the execution of a given process, based upon information about the quality of the product or service, or regarding the execution of the processes that actually produces or manages the given product or service.

Metrics share a common set of desirable characteristics:

- Must be simple, objective, easy to collect, easy to interpret, and hard to misinterpret
- Must be automated and non-intrusive to minimize interfering with other project activities
- Must contribute to take fast corrective actions, when actions can be more effective
- Must be consistent and expressed in some absolute value

Defining metrics and measures

Before we continue this conversation about metrics and measures, we should more carefully define our terms (Figure 2-6).

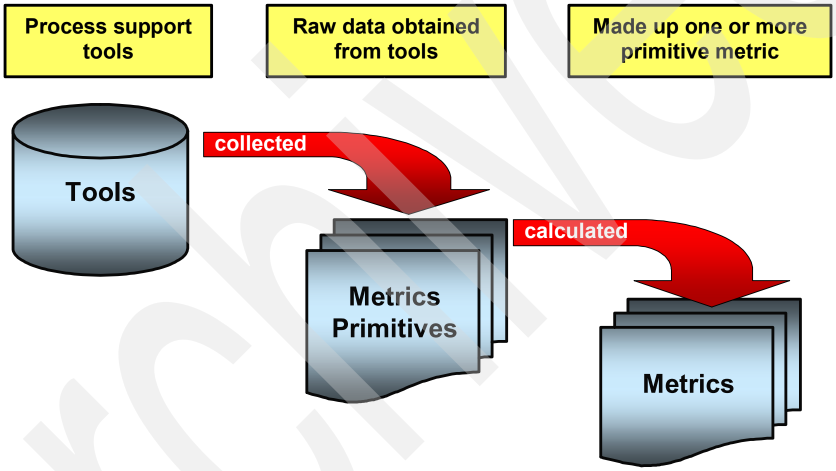


Figure 2-6 Types of metrics and obtaining mechanisms

- The term *measures* is used to describe metrics primitives. In other words, measures are metrics used to calculate other metrics. These measures are raw data and are usually obtained directly from the tools that support processes, or from the tools that are used to generate work products. Primitive metrics cannot be interpreted in isolation.
- The term *metrics* is used for measurable attributes of entities, where each metric is made up of one or more primitive metrics and can consequently be evaluated alone or in conjunction with other metrics.

During the process of metrics selection, each organization decides what goals and objectives will be monitored during the measurement process. Utilizing the agreed upon goals and objectives, appropriate measures can be selected with which the metrics are derived. Be careful not to select too many measures, or to select measures that are unfitting measures to observe behavior from an organizational perspective to be of value. In other words, measures should be placed at the organizational level to monitor the governance process itself. Measures should be implemented at the program or project level to monitor software development process conformance, and at the artifact level to monitor product or service quality levels.

Finally, be careful not to take unnecessary metrics. Collecting a large number of metrics is likely to be a waste of time because of the effort typically required to collect, process, and store the raw measure data. A minimal set of more extensive and simple metrics is preferred. A reasonable set of measures reduces the effort of collecting measurement data while still delivering the same level of benefit. A good rule of thumb allows no more than four to five metrics, with appropriate measurement data to support them.

Selection of key metrics

Organizations unaccustomed to supporting metrics and measurement programs are often challenged by the thought of selecting only a few metrics from which to operate and control a process. One useful approach to identify metrics is to look at the primary group of interested stakeholders and align selected metrics with each group's interests. Quite naturally this will lead to the selection of organizational metrics, project or program metrics, and technical metrics. We define these as follows:

- *Organizational*—Examples of organizational metrics are those related to manage cost and improve organizational performance, reduce risks, and manage business compliance.
- *Project*—Examples of project metrics are those related to being able to manage functional and non-functional capabilities, and technical, budgetary, and scheduling constraints.
- *Technical*—Examples of technical metrics are process-related and product-related metrics that contribute to project-level metrics, but are at a lower level and more useful in the technical analysis or product quality, typically supporting technical personnel.

For more information related to metrics and measurements refer to:

- Rational Unified Process (RUP)

  http://www-306.ibm.com/software/awdtools/rup/
- Practical software and system measurement

  http://www.psmsc.com/psmrmc.asp

General considerations and practical control strategies

Although this book has spent a great deal of time discussing the need to implement appropriate processes, controls, and metrics to ensure process conformance and the ability to demonstrate due diligence, not every project in an organization is in need of formal compliance. This is to say that within a company there are many software development projects and packaged application implementation projects that pose little, if any, risk to either the business or to its stakeholders. However, there are clear advantages to the business should these *uncontrolled* projects adopt some of the standard infrastructure required by those projects that are in need of formal compliance. Among these advantages are:

- Consistency in many of the steps in the workflow
- Consistency in work product templates and best practices for their development
- Standardization on tooling infrastructure and usage
- Economies of scale, provided by the shared infrastructure afforded form the regulated projects

Leveraging the infrastructure

The infrastructure we have described thus far for compliance is extremely robust. You may even be saying to yourself that this looks like far too much. If you have this feeling, it is likely that you might just be right, but do not stop reading yet.

Every organization, even within highly regulated, multinational, geographically distributed operations, can have quite different characteristics. Even those projects within each organization can have very different priorities, requirements, and technologies.

What we have described throughout this book has been the type of infrastructure necessary to support extremely large, high integrity projects, such as projects for the development of medical devices or pharmaceutical products. In these conditions, the risks and the costs of bringing the products to market are extremely high. Consequently, extremely formalized and strict process conformance is required to protect people from problems in product quality.

The reality is that there is a continuum that begins at very small projects that are completely uncontrolled, right through to large projects that we have described in this document.

To illustrate this range of operational tolerance, we look at two common relevant considerations related to compliance: security and process (Figure 2-7).

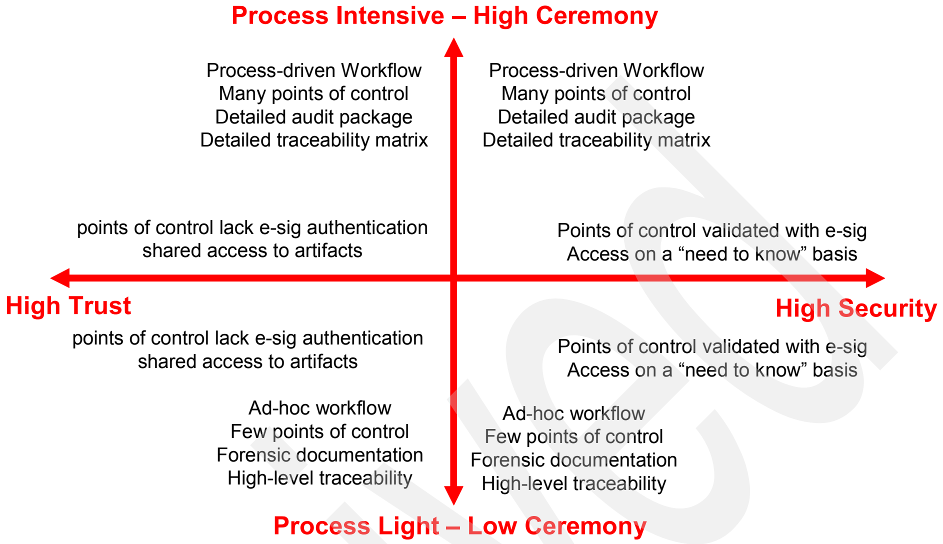

Process Intensive – High Ceremony
Process-driven Workflow
Many points of control
Detailed audit package
Detailed traceability matrix
Process-driven Workflow
Many points of control
Detailed audit package
Detailed traceability matrix
points of control lack e-sig authentication
shared access to artifacts
Points of control validated with e-sig
Access on a "need to know" basis
High Trust
High Security
points of control lack e-sig authentication
shared access to artifacts
Points of control validated with e-sig
Access on a "need to know" basis
Ad-hoc workflow
Few points of control
Forensic documentation
High-level traceability
Ad-hoc workflow
Few points of control
Forensic documentation
High-level traceability
Process Light – Low Ceremony

Figure 2-7 Compliance quadrant: finding the right process for your organization

High trust and high security

In Figure 2-7 movement from left to right in the four-quadrant diagram takes you from those high trust environments (where security constraints are very low) to those environments where security and data integrity is a major concern:

- In high security environments a requirement for the use of e-signature to implement points of control for non-repudiation of data may be an unavoidable requirement.
- Conversely, in less formal environments where there is no need for such non-repudiation requirements, simply having developers check in their code might be enough.

High and low ceremony

Movement along the vertical axis from bottom to top illustrates the need for organizations to formalize the documentation and execution of their software development process:

- Low ceremony organizations are typically those with a small number of members where communication is easily facilitated.
- High ceremony teams are usually large-scale teams, often geographically distributed. In these conditions, implementation of strict adherence to process

ceremony is encouraged and enforced. In such conditions it is often difficult for all members of the team to communicate regularly. Consequently, institution of very formalized and ceremonious processes ensures that product quality is high through the use of formalized process-based checks and balances. In this way, high ceremony processes protect stakeholders, but increase cost considerably.

Establishing your organizational position

When evaluating your organizational position on the chart, consider the following. Most regulated organizations typically seek to implement a set of consistent best practices across the organization, as well as produce the appropriate number of deliverables or work products mandated by a given target standard.

Two considerations facilitate impact on the business. First, the implementation of a consistent process ensures that several fundamental measures and metrics can be kept for projects in need and not in need of formal compliance. This permits the organization to compare organizational metrics for effectiveness, productivity, alignment with business goals, and so forth. The second function this provides is greater flexibility regarding the use of resources within the organization. Because the process is consistent, the company can transfer development resources more easily from project to project with minimal disruption. This adds to the overall productivity of the organization.

Practical control strategies

Once a company has determined in which quadrant its capabilities currently lie, it is necessary to determine the strategy to be followed to address a software development process in need of change.

General strategy

One possible approach to implement a compliant software development process is based on the five-step flow of activities (Figure 2-8).

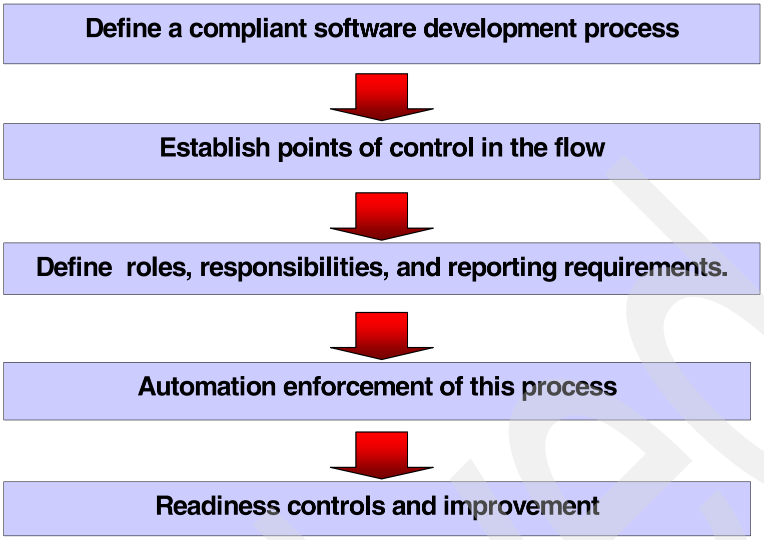

Define a compliant software development process
Establish points of control in the flow
Define roles, responsibilities, and reporting requirements.
Automation enforcement of this process
Readiness controls and improvement

Figure 2-8 Practical control strategy steps

Define a compliant software development process

Current best practices for the orchestration of software development activities should include accounting for the management of releases of software products. This requirement is closely followed by the requirement for the management of change requests, which in turn is linked to releases for scheduling, development coordination, and deployment.

Some useful questions to define a compliant software development process are listed here:

- Key software project questions
  - How do projects get approved and funded?
  - Who is authorized to access project assets at each stage?
  - What authorizations are required, by whom, and when?
  - What are project audit trail requirements?
  - What are the project documentation (reporting) requirements?
  - Where are the project packages stored, and for how long?
- Key software change questions
  - How do software changes get approved?
  - Who is authorized to access project assets?
  - What approvals are required, where, and by whom?
  - What are audit package requirements?
  - Where is the change package stored, and for how long?

Assign roles, responsibilities, and reporting requirements

One of the most natural mechanisms for the selection of roles and responsibilities is to lift the existing procedures and automated them. Quite naturally, organizational boundary crossing dictates the roles for which implicit authority is granted. However, recognition of the need to bypass these conditions should be accounted for, and built within the workflow. After all, there will be a robust audit trail should the actions executed by the backup be deemed inadequate or inappropriate. Use of existing documented processes reduces the need for additional training, and increases the likelihood of success by accelerating process adoption due to reduced learning curve.

Establish points of control in the flow

Once the compliance software development process is defined, it is time to determine where constraints on workflow are required to introduce points of control. Typically, this is implemented through the use of electronic sign-off or reauthentication in order to impose needed business and technical controls.

Typically, these control points are related to the satisfaction of quality conditions on the work products being produced throughout the process. However, these control points may also be established in response to the need for accountability when crossing organizational boundaries or technical domains. Examples of such conditions may include:

- Movement of executables from the development environment to a quality assurance environment
- Sanctioning of a given software architectural construct by the architectural team

The three points of control strategy

One practical control strategy that incorporates much of the infrastructure we have discussed throughout this paper is the *three points of control strategy* by IBM Rational. This strategy—although primarily focused on the configuration and change management realms—provides an excellent reference for customers looking to instrument appropriate checks and balances over the management of software development artifacts and work products.

The strategy employs four types of records, each implementing their own workflow:

- *Release*—The release record is used for the management of target build activity intended for release to production at some future date, ultimately transitioning to production once all change requests have been completed.

- *Change requests*—Change requests are child records to releases and serve to track related activities that must be completed together to support some delivered application functionality. For example, a particular application may have both a Motif and Windows™ client. However, the change request will not be considered complete until the functionality has been implemented for both interface types.
- *Activity*—The activity record gives team leaders and release managers the ability to assign related, but dissimilar tasks to different developers, while tracking the delivery of functionality through the change request record. Ideally, these activities should be used through an integration with the configuration management repository, thus linking all code modifications for each activity to all versions of code modified to implement a feature or application fix.
- *Deploy*—The deployment record type is used to manage the movement of application release builds for each operational environment to the next. One example of such a movement would be the approval of movement of code from the development environment to the quality assurance build area on a certified build machine. Naturally, your organizational procedures may be different, perhaps permitting a copy operation of built bits from the development environment into the quality assurance area. Refer to Figure 2-9 for the details regarding the actual steps.

The three control points are explained in the sections that follow.

Control point 1: Deliver change

This control point manages the incorporation of individual changes associated with an activity into the integration environment. During this control point the implementer posts activities for approval by the technical lead.

The technical lead can approve the activity and allow the incorporation or disapprove the activity, sending the changes back to the implementer for rework. Once the activity has been approved, no additional changes can be made to the change set associated with the activity.

Control point 2: Register derived objects

This control point manages the addition of deliverable objects to source control. This control point occurs once development-level integration testing is complete and captures the build results targeted for quality assurance (QA) testing. Only the integrator has the authority to register build objects for QA.

Control point 3: Deploy objects

This control point manages the deployment of objects to QA and to the production environments. Once a registered release is ready for QA or production, this control point verifies that the work included in the release is complete and the build was performed in a controlled environment. Only the release manager has the authority to approve a deployment to QA or production.

Figure 2-9 shows the three points of control strategy.

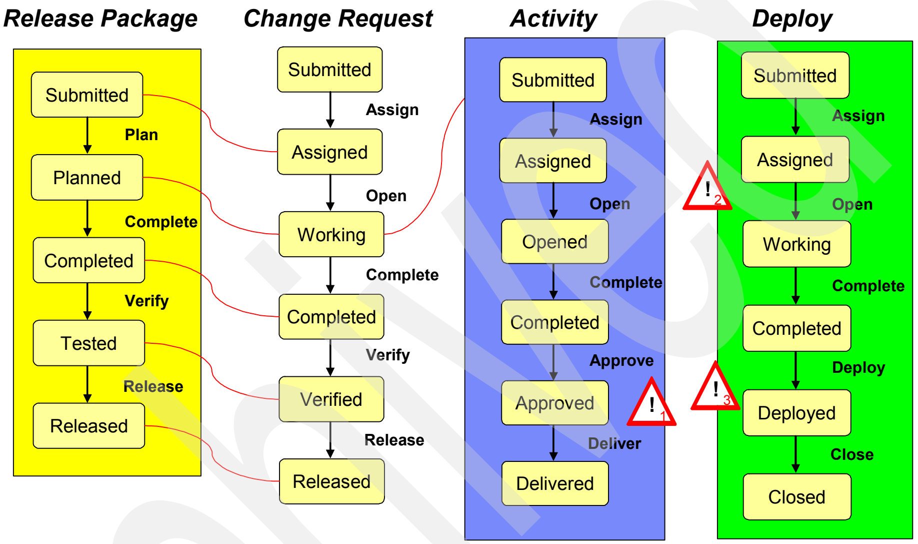


Figure 2-9 Representation of the three points of control strategy

Roles and responsibilities in this strategy

Table 2-1 shows the roles and their responsibilities in the three points of control strategy.

Table 2-1 Roles and responsibilities in three points of control strategy

Role	Description	Responsibility
Project manager	Plans and manages software releases from request for change through the development and distribution process. Sometimes known as the release manager.	Owns the release package. Creates and assigns change requests to the appropriate application teams. This role also approves releases to production environments.

Role	Description	Responsibility
Technical lead	Guides and oversees development of application software components.	Ensuring that all configuration items are included for the application to function in a test or production environment. Primary source change approver. Creates and assigns UCM activities.
Integrator	Creates and maintains all scripts necessary to build an application for use. Performs controlled builds for one or more applications.	Creates deployable objects and the appropriate deployable baselines.
Release manager	Deploys configuration items from controlled sources to the various environments used to test an application or into production itself.	Creating and executing the deployment instructions for an application.
Tester	Representative from the quality organization responsible for verification of one or more applications.	Controls the approval or rejection of a release package in the appropriate quality states. Approval is required before any release package can be moved into a production environment.
Implementer	Participates in design activities and constructs, tests, and documents application configuration items.	A primary user of the solution, this role is responsible for making changes to code and assuring the level of quality of those changes. Relies on the solution for version control and workflow enforcement.

3

IBM Rational's key capabilities for compliance management

In this chapter we describe the process, tools, and services components of the IBM Business Driven Development for Compliance solution. We also illustrate how the IBM solution can be customized to fit an organization’s specific needs. Finally, we refer to a number of IBM Rational tools that are used within the solution. In the interest of brevity, we only provide brief descriptions of each of the tools, using sufficient detail to provide the reader with an understanding of what functionality each tool provides. Significantly more detailed descriptions of each of the tools is provided in Chapter 4, “Roadmap to compliance management” on page 119.

The IBM Rational solution for compliance management

There are essentially three things that any business should be prepared to do in order to respond adequately to an audit. These three things are to be prepared to *“Say what you do, do what you say, and be able to prove it.”*

The objective, as simple as it may seem, is actually quite complex. This is because if a company is to succeed in substantiating this level of control, the company must be adequately prepared at all times to demonstrate sufficient evidence. Often these three simple items can be thwarted by the very processes that are designed to ensure their integrity. They can also be undermined by the needs of the business to innovate, or respond to, market demands.

Thus a demonstration of these capabilities often requires evidence of appropriate organizational behavior, as well as appropriate linkage of artifacts related to an organization's compliance efforts. In other words, you must be prepared to “Say what you do, do what you say, and be able to prove it” at all times.

As we established earlier, the goal of any company is to demonstrate that the performance and procedural requirements of their policy can be implemented effectively to demonstrate compliance. However, implementation of a great deal of process can have an adverse effect on productivity. Furthermore, it is possible to create a significant amount of new process, thereby improving process compliance, with little or no productivity gain.

After all, in this highly competitive world you must not only comply, but you should increase productivity, and make a profit as well. IBM understands that automation is a key factor to transforming a compliance solution into a more cost-effective operation. In a perfect world, customers would like to focus on the delivery of their core business product or service, and as a consequence they would automatically comply. Although there is no solution on the market that can accomplish this today, this is the kind of vision IBM is working toward.

Figure 3-1 shows a stack of IBM Rational product offerings designed to provide automation functions for various areas in the compliance domain.

The entire solution provides a complete infrastructure that can allow for sustainable compliance, providing evidence that is captured in several modules. Each of these modules applies specific control principles that can be automated through the use of IBM products. Automation and integration are key characteristics of this solution. The business and technology controls required within the software development life cycle are implemented in a non-intrusive manner and serve as the basis for our governance solution.

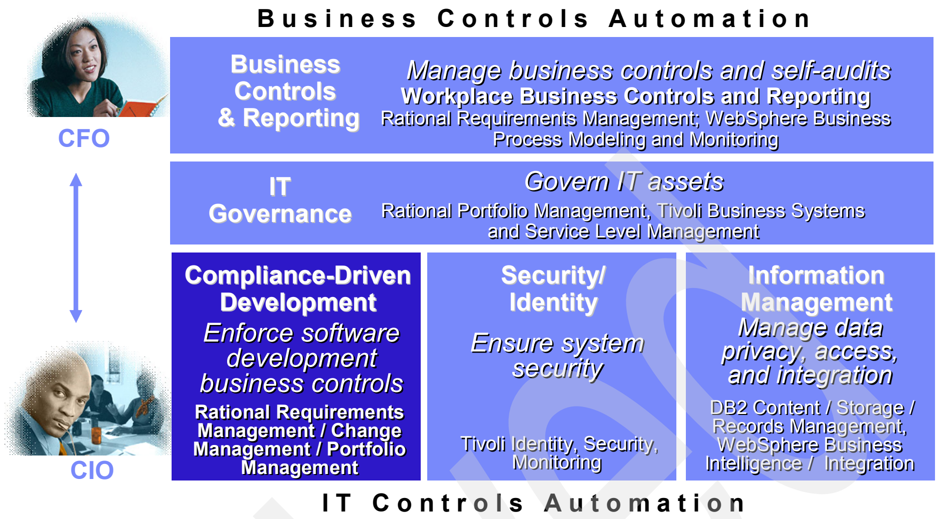

Business Controls Automation
Business Controls & Reporting
Manage business controls and self-audits
Workplace Business Controls and Reporting
Rational Requirements Management; WebSphere Business Process Modeling and Monitoring
CFO
IT Governance
Govern IT assets
Rational Portfolio Management, Tivoli Business Systems and Service Level Management
Compliance-Driven Development
Enforce software development business controls
Rational Requirements Management / Change Management / Portfolio Management
Security/ Identity
Ensure system security
Tivoli Identity, Security, Monitoring
Information Management
Manage data privacy, access, and integration
DB2 Content / Storage / Records Management, WebSphere Business Intelligence / Integration
CIO
IT Controls Automation

Figure 3-1 IBM Rational: a modular solution for reducing cost of compliance

Business-driven development for compliance

Although many organizations are cited during audits for inadequate controls throughout the business, it is those organizations within the development governance context that we will focus on. If you are asking yourself, “what does compliance have to do with software?” you will be interested in what we have to say.

An organization’s policy reflects the organization's interpretation of a regulation. Because policy is typically implemented as both manual procedures and as code in automated systems, any change to a given policy would require a rewrite of the manual procedures, or a rewrite of the code in an automated system. Thus, we have our linkage between policy and automated and manual systems.

Organizations seeking to implement a solution, such as Rational's, should begin by executing an analysis of existing policies and procedures that are targeted for compliance. This is an important first step in the selection and assembly of the components of the IBM BDD for Compliance solution.

Since we have embarked on a discussion regarding policy, standards, and regulations, it is important that IBM state clearly that IBM customers are solely responsible for ensuring their own compliance with their legal requirements. It is also the customer's sole responsibility to obtain competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements (for example, in the translation of regulations into business policies). This is a very important step, as it is likely that changes in company policy will affect the customer's business and any actions the customer may have to take to comply with the law.

After identifying policies and procedures designed to address compliance concerns, a company may begin appropriate projects to implement the policies defined. These projects might be related to manual control implementations, sales force training, systems development, and changing or implementing new requirements on existing systems. Once a company has established its policies, IBM Rational can assist the customer with the automation of those policies through several automated mechanisms contained within the IBM Rational Business Driven Development for Compliance solution.

Consequently, the scope of this book is limited to business the driven development for compliance capability offered by the IBM Rational brand with a light treatment regarding those components of the solution outside of the brand offerings.

So let us review what we have established thus far. Compliance requires both manual and automated business controls over the application portfolio, as well as the software development life cycle to address three main capabilities:

- *Say what you do*—Establish a compliant development process using:
  - Business controls and workflows
  - Technical controls and workflows
  - Approvals, authorizations, quality gates, separation of duties
- *Do what you say*—Automate enforcement of your process trough:
  - Process guidance
  - Automated workflow
  - Tool-directed behavior
- *Be able to prove it*—Automate generation of audit documentation, such as:
  - Audit reports
  - Audit trails for the software development organization

Establishing, enforcing, and monitoring such an integrated process affects the way a company develops systems, and how these systems are managed. Compliance becomes both a requirement and an opportunity to drive fundamental improvements in software development maturity. Compliance implies audits, audits imply traceability, and traceability can be a headache, which we discuss later in this document.

The specific components of the general solution that support audit, traceability, and workflow enforcement are shown in Figure 3-3. Notice that the key components used to manage business controls and reporting include Workplace™ for Business Controls and Reporting (WBCR), and those for IT governance include Rational Portfolio Manager (RPM).

These components facilitate the management of the artifacts used for tracking regulations, standards, policies, and other business risks, demand management, resource management, project proposal tracking and workflow, balanced scorecard design and evaluation, project tracking, and cross organizational coordination.

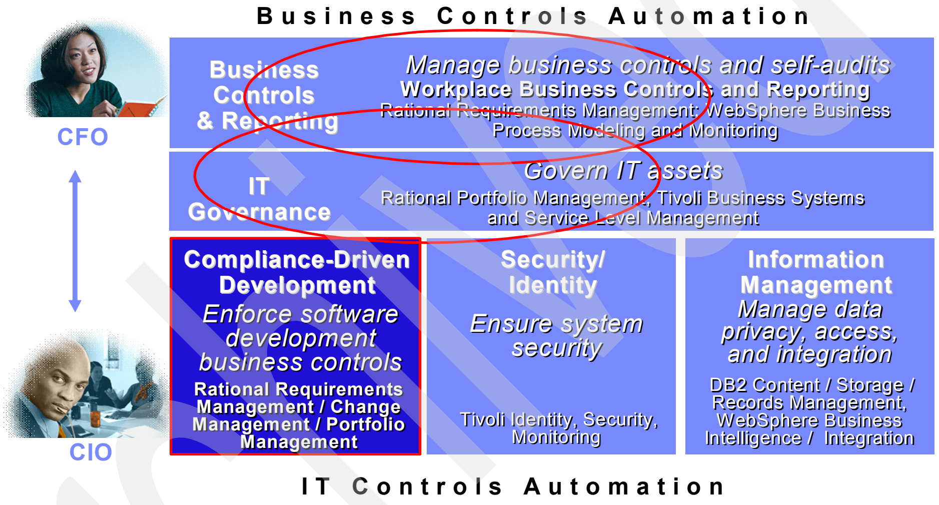
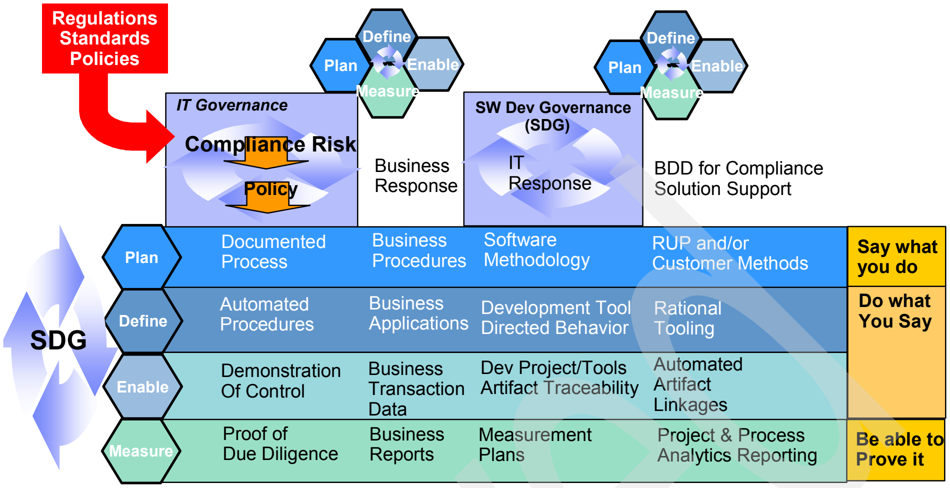

Business Controls Automation
CFO
CIO
Business Controls & Reporting
Manage business controls and self-audits
Workplace Business Controls and Reporting
Rational Requirements Management, WebSphere Business Process Modeling and Monitoring
IT Governance
Govern IT assets
Rational Portfolio Management, Tivoli Business Systems and Service Level Management
Compliance-Driven Development
Enforce software development business controls
Rational Requirements Management / Change Management / Portfolio Management
Security/ Identity
Ensure system security
Tivoli Identity, Security, Monitoring
Information Management
Manage data privacy, access, and integration
DB2 Content / Storage / Records Management, WebSphere Business Intelligence / Integration
IT Controls Automation

Figure 3-2 IBM BDD for Compliance solution

The components in the upper two layers of the IBM BDD for Compliance solution benefit customers with the following:

- Control principles that ensure appropriate capture of evidence of due diligence
- Tools to automate these principles and provide traceability across development artifacts
- Services to customize the BDD process such that it applies these principles to conform to the organization's existing environment

These components are combined to address compliance issues and needs through IBM BDD features (Figure 3-3).



Plan	Documented Process	Business Procedures	Software Methodology	RUP and/or Customer Methods	Say what you do
Define	Automated Procedures	Business Applications	Development Tool Directed Behavior	Rational Tooling	Do what You Say
Enable	Demonstration Of Control	Business Transaction Data	Dev Project/Tools Artifact Traceability	Automated Artifact Linkages	
Measure	Proof of Due Diligence	Business Reports	Measurement Plans	Project & Process Analytics Reporting	Be able to Prove it

Figure 3-3 Compliance problem and solution spaces

Some of the most commonly desired information for an audit is listed here:

- Documented software development life cycle (SDLC) processes for creation of all business artifacts
- Adherence to the defined process by artifact creators/SDLC compliance
- Linkages between supported tools using tool-directed behavior
- Process linkages to automated processes supported by tool mentors
- Linkage between artifacts that align with the business process steps
- Artifact traceability to points of accountability in the delivery chain
- Transparency of reporting
- Accountability in the chain of delivery
- Non-repudiation of any artifact throughout the system
- Evidence of SDLC process maturity and stability
- Documented test results requirements linkage
- Documented and formalized handoffs and signoffs

The following are the key capabilities of the IBM BDD solution that make it possible to gather the proof necessary to substantiate compliance. They are delivered through the application of both the principles and the tools:

- Life cycle requirements traceability to help auditors verify that compliance requirements were accurately captured and implemented in key applications

- Auditable workflow management capabilities that help ensure and document that all software changes were made by authorized personnel for valid business reasons
- Flexible metrics and reporting, electronic signature, and audit trail capabilities that can be tailored to the exact processes and IT controls that govern your development environment
- Verifiable software builds to help ensure and document that software developed was actually deployed
- Automated deployment that is fully integrated into the development process
- Continuous validation of compliance mandates through integrated test management
- Tool-directed behavior (TDB) with appropriate product and process quality metrics management
- A fully integrated process and product audit console solution for the process and development artifacts

Control principles

Any solution for compliance should assist a company in the demonstration of due diligence for compliance with that regulation, standard, or policy. This is accomplished by accumulating documented evidence of organizational behavior, responsibility for artifact management, and due diligence regarding the implementation of policy within these efforts. Remember, the goal is to “Say what you do, do what you say, and be able to prove it.”

The control principles defined within the IBM BDD for Compliance solution are a set of automated current good practices for application development. These control principles have been pre-tested as the success factors for many other software development projects over the course of the past ten years. In the context of compliance, these control principles play a much more crucial role. The automated control principles provided with the tools, in conjunction with the documented best practices, aid organizations in the delivery of compliance-related information required by auditors. The data that must be provided to auditors is accumulated through the automated traceability features provided within the tools. We refer to the features that establish the traceability, along with the process enforcement provided, as *tool-directed behavior*.

Look back at the list of information most commonly requested by IT auditors. Notice that the key to demonstrating control over your process is traceability. Demonstration of control, from a traceability perspective, requires linkages from the regulation to the business policies, to application requirements, to the project plans, to business process changes, to application changes, and all of it must be under change control.

These principles are the foundation of a repeatable process that together with tool-directed behavior provides the traceability required. Due to the transparency of this approach, managers and executives are now empowered to make more effective decisions based upon near real-time project data.

The IBM BDD for Compliance solution is a compliance-driven development infrastructure that consists of a set of well-established principles, tied to a documented and easily customized operational process standard, that directs practitioners in their day-to-day conduct through the use of specific tooling. Finally, the process enforcement provided through the tooling ensures process conformance by practitioners, ensuring measurement data is accurate across these projects as well. This is an important aspect of our solution, which can be easily overlooked.

Companies that fail to implement a consistent software development process will fail in more ways than one. Failure to implement a consistent process means that any measures or metrics from one project to another will be inconsistent. Therefore comparative results will be skewed. This has a detrimental impact on decision making.

Development governance

Say what you do

In order to accurately assemble an appropriate solution for compliance, it is necessary that we describe some of the overlapping functions of IT governance, which by definition fall outside of the domain of software development. You must understand that IT governance is a comprehensive process comprised of many other smaller processes. Although IT governance is a goal to strive for, it may encapsulate a process that is far too heavy for most organizations.

The purpose of including some of the unrelated software development processes of IT governance with our development governance solution is to incorporate the necessary related processes to help achieve compliance. Examples of such processes include application release management and application deployment. IT governance also contains other peer processes to IT operations, such as those used for the software development process itself. The reason these processes must coexist together is because of the implied linkage that exists among these interrelated processes.

An example of when a peer process is described within the solution is when a software patch has to be deployed into production to fix an application problem. It also occurs, for example, during the process of coding and testing the fix. Finally, you must recognize that the formalized best practices for IT operations and software development have evolved into standalone codified process standards, such as the Information Technology Infrastructure Library (ITIL) and the Rational Unified Process (RUP). Each of these process standards can be considered a kind of governance for IT operations and software development, respectively.

Consequently, a failure to describe some of the high-level processes for IT governance and the interrelated processes of IT operations would be a failure to describe a complete development governance solution. Thus we begin our description of the IBM BDD for Compliance solution.

Discussion

Meeting compliance mandates is both a challenge and a business need for organizations in most markets. The potential impact that a given policy can have on a business is wide spread and enterprise encompassing. Consequently, businesses responding to regulations, standards, and policies must realize that implementation of appropriate processes and business controls must be implemented throughout the business operation as well as throughout the systems inventory.

However, companies cannot attack this problem without first assessing the impact that new policies will have on the business operations, as well as the systems impacted by the resultant policy. This dictates that companies evaluate the impact a given policy will have on the business as well as its systems inventory.

Examining this problem from a business process perspective, the organization can leverage tools such as Workplace for Business Controls and Reporting (WBCR) to capture policy requirements and trace them to the manual business controls (processes) that ensure their implementation.

However, from a systems perspective, the ability to accomplish this task can be significantly more daunting. The challenge is that most companies do not utilize technology that links the systems inventory to all of the related project management plans, the resources executing those plans, in conjunction with the project proposal and demand management process. Further complicating this problem is the absence of any functional knowledge at the executive level regarding which systems, or systems components, implement policy. Finally, without a complete understanding of the business issues each application is designed to mitigate for the business, how can any company hope to perform an informed analysis on the systems inventory?

This has a direct impact on the quality of business decisions regarding IT investment, because an adequate level of information to make an informed decision is typically unavailable. This absence of information to determine which projects or proposals an organization should invest in typically leads to the hiring of a firm to gather this information, or a large-scale information-gathering initiative. Although this approach is pragmatic, it will surely elongate the time for change, which is likely to be operating on a fixed deadline to begin with. Once all of this information is aggregated, a careful balancing of business issues and business investment can then be acted upon. Compliance initiatives may or may not be related to software development, but from a business perspective, all of these processes are part of an operational portfolio that must be governed.

In Figure 3-4 you see an example where external regulations, standards, and policies (in red) are the starting point. In response to these regulations, companies typically formulate an interpretation of these regulations as their internal policy. After performing impact analysis on the business and business systems, candidate projects are identified for modification. Once approved, these projects are then instantiated in their respective business units, and finally to the supporting application systems or as business process implementation work.

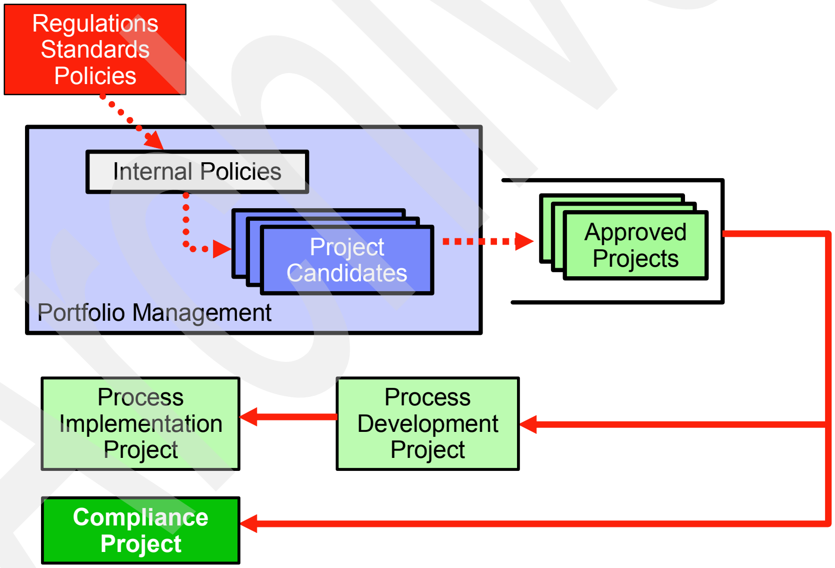


Figure 3-4 Governance for compliance initiatives

The practice of governance for compliance must start at the portfolio level to ensure its effectiveness. The portfolio management process is shown in Figure 3-4 as the blue box surrounding the internal policies and project candidates objects. Unless the organization has at its disposal all of the necessary information with which to demonstrate the intention to make the right business decisions, it can be difficult to show that the organization intends to protect the interests of its business stakeholders. A portfolio-based approach to compliance allows executives to have the necessary information on-hand, in order to prioritize compliance and non-compliance initiatives, and evaluate and mitigate related risks to make appropriate decisions.

However, the benefits of this portfolio-based approach to managing compliance initiatives extend beyond the compliance domain. Other benefits of leveraging such an approach include:

- Improving the effectiveness of project teams and existing technologies to deliver better business results within the constraints of current investments and skill sets
- Improving the management of business and technical risk to lower the costs of the software innovation and change necessary for business resilience and growth

Because the solution also incorporates workflow for the management of the portfolio, these additional challenges are addressed by the governance practices and mechanisms as well:

- Establishment and enforcement of chains of responsibility, authority, and communication to empower people to make the correct decisions
- Establishment of measurement, policy, and control mechanisms that enable people to carry out their roles and responsibilities

In general, you see that our compliance solution is really focused on governance by supporting:

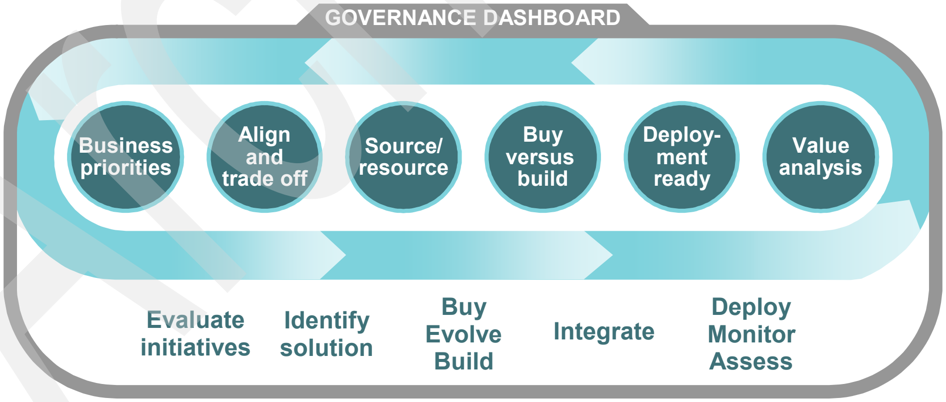
- Chains of authority and responsibility and clear channels of communication to facilitate the objective setting and informed decision making that keeps IT aligned with business
- Establishment and execution of measurements and controls to enable people to carry out their roles and responsibilities in ways that maximize development flexibility, minimize risk, and facilitate effective change management

As we defined earlier, governance is the empowerment of others through appropriate authority, responsibility, communications, and measurement. What is different from the generic definition of governance in the context of software development governance is that we are applying it to the business process of software and systems development.

Consider the following hypothetical scenario: A major automotive manufacturer in early 2006 required its IT organization to take corrective action to comply with deficiencies found during a Sarbanes-Oxley audit. This organization had approximately 10,000 applications in its systems portfolio. Due to the absence of a systems inventory, or any consolidated knowledge of the application inventory, it took five months to determine that 2,000 application were potentially impacted by a new policy. Finally, after interviewing each of the application owners, it was determined two months later that approximately 200 of those applications directly impacted financial reporting. Had the customer been utilizing a portfolio-based approach for managing the systems inventory, a request for self disclosure using an online questionnaire might have saved the company seven months of potential implementation time.

The overall goal of the business-driven development solution from a portfolio perspective is to shift the focus from a purely process discipline of service-level issues to those of corporate risk mitigation. A shift to this perspective adds new value to the IT portfolio, as well as to the IT organization as a whole. Properly balancing risk and return on investments is not simply a set of technical decisions. When you govern the business process of software development and decision making, you move from managing an activity once treated simply as a cost center to managing the decision process itself.

Figure 3-5 illustrates some of the business criteria that impact the portfolio management process, as well as the progression of decisions (circles) that result from those criteria.



Informed decisions drive development activities

Figure 3-5 Business-driven development life cycle

Supporting tools

IBM *Rational Method Composer* (RMC), the industry's leading process platform, is now integrated with Rational Portfolio Manager (RPM). IBM Rational Method Composer is designed to allow customers to either tailor the Rational Unified Process or to capture their own processes in a standard format recognized by the Object Management Group (OMG). The tooling is a based upon the Unified Method Architecture (UMA), which uses UML to internally store the documented processes.

Leveraging this technology allows customers to easily extend documented processes through the application RMC plug-ins. Since each plug-in is designed with UMA, and is UML based, it inherits all of the object-oriented advantages of UML. Thus, any existing processes documented using this technology can be easily extended by simply inheriting the characteristics of other processes or plug-ins. For example, customers wishing to extend their existing process could download and apply a plug-in for SOA, and have their process inherit all of the activities, work products, roles, and processes defined by such a plug-in. Finally, unlike other process documentation solutions, RMC allows documentation of not only process guidance, but work breakdown structure, as well as process dependencies. This allows a complete articulation of process guidance that is fully linked to an executable process.

Through the RMC-to-RPM integration, users can automatically apply industry-leading best practices to project plans by exporting the documented methodology. These project plans can then serve as templates for the instantiation, costing and estimating of potential development projects within the demand management function of RPM. This saves time and money by replacing time-consuming, error-prone manual approaches with proven and consistent processes and tooling.

Rational Portfolio Manager

Rational Portfolio Manager (RPM) is an IBM strategic tool for enablement of enterprise project management, providing a Web-enabled, common repository that supports key project management activities for all stakeholders. RPM can be used to break down the project tasks and track schedule dates, planned, and actual hours. The project status information provided through RPM helps organizations control projects and optimize resources through more timely and accurate information, while also providing portfolio management views for middle and executive management.

RPM helps business executives and IT leadership align IT investments with business goals. It helps project teams efficiently plan and execute projects. It provides a collaborative workflow environment for project and asset management, insight into resource capacity planning, and oversight of project financial.

RPM's marriage of top-down portfolio analysis with bottom-up project management best practices provides a comprehensive on demand capability—from initiative identification and investment prioritization through project execution and closure. It is about achieving the desired return for your IT investment, while balancing fiscal expenditures.

RPM makes it possible to:

- Monitor current projects by periodically reviewing:
  - Investment maps
  - Organizational scorecard for compliance
  - Scorecard mapping
  - Online analytical processing (OLAP) pivots (for more details)
  - Other reports as appropriate
- Monitor and control risks:
  - Perform risk trend analysis.
  - When a risk materializes, incorporate its risk response package into the work breakdown structure (WBS).
  - Execute the risk tasks in the WBS.

Using Rational Portfolio Manager, project information is entered into a repository and it is made available to others as their needs dictate. RPM functions are designed to serve the main stakeholders of a project, namely:

- For project managers to plan, track, and control projects
- For project team members to access relevant standards, procedures, and other documents
- For project team members to receive their assignments and to report their progress
- For project executives to maintain awareness of the status of all projects in their area of responsibility/portfolio
- For resource managers to effectively manage the resources within their area of responsibility

To learn more about Rational Portfolio Manager refer to:

http://www-306.ibm.com/software/awdtools/portfolio/

Supporting process

The supporting process is defined in the *Project Management* discipline in IBM RUP.

Conclusion

Table 3-1 shows the control principles for IT governance.

Table 3-1 Control principles for IT governance

Control principle	IT governance
Benefits	► Align IT projects and investments with business priorities. ► Plan and manage portfolios of projects to meet enterprise objectives.
Pattern	► Manage compliance requirements as risks at the portfolio level. ► Focus on resolving risks. ► Qualify initiatives with business attributes.
Antipattern[a]	► Not using attributes to qualify initiatives. ► Projects being completed late, or not at all, and usually over budget.
Evidence	► Evidence of SDLC process maturity and stability. ► Transparency of reports.

[a] An example of an inferior design solution that is commonly made by developers. They are used to reinforce better planning during the development process as well as provide a problem-solving reference point.

Requirements management

Say what you do

Although requirements are routinely thought of as features for software development, requirements can be captured for nearly any project or initiative. Let us consider for a moment your own criteria for the neighborhood where you selected your home. Did you have requirements for good schools, good resale value, a community you could relate to with similar values, or proximity to other family members? All of these are requirements that could be considered requirements for selection of a neighborhood in which to live.

Shifting the perspective to a compliance example, you might consider other criteria as input for a compliance-related initiative. Regulations can be considered criteria for risk requirements and subsequent risk mitigation, and policies can be considered your interpretation of a regulation. Using this perspective on regulations, policy, and process, you should be able to see the importance of organizing regulations and policies in a structured way so they can be traced to systems and integrated with other life-cycle tools and artifacts.

Of course, it is always essential to ensure that requests for new enhancements or changes to systems are implemented appropriately. By extension, the same regulations and policies described earlier may have an impact on those systems that automate policy as an automated process.

Discussion

As the *say what you do to be in compliance* component of the IBM BDD for Compliance solution, you should be able to demonstrate that all compliance mandates are accurately captured and implemented in your key applications.

The consequence of such an imperative is that compliance mandates become a significant source of the requirements for future releases of those systems otherwise overlooked.

The consequence of such imperative is that compliance mandates become the main requirements source for business and supporting systems.

IBM RUP defines the requirements management discipline as “a systematic approach to finding, documenting, organizing, and tracking a system's changing requirements.”

A requirement is defined as “a condition or capability to which the system must conform.”

RUP also formally defines requirements management as a systematic approach to both:

- Eliciting, organizing, and documenting the requirements of the system
- Establishing and maintaining agreement between the customer and the project team on the system's changing requirements

Every organization must obtain advice of competent legal counsel for the identification and interpretation of any relevant laws and regulatory requirements that might affect its business operation. From this internal interpretation of external regulations, risk officers and analysts create a set of policies for the company, which will guide the company’s behavior. Because systems are used to automate business behavior, these policies might create new systems requirements, change existing ones, or demand a new system development.

In the IBM BDD for Compliance solution policies, standards, and regulations are the highest level requirements source for the systems within an organization. This is because they are usually linked not only to one system, but to many different systems.

The key to effective requirements management is maintaining a clear statement of the requirements, along with appropriate attributes and traceability to other requirements and other project artifacts.

It is also important that these requirements have a high level of maturity, which means that every feature—whether a regulatory requirement or not—be written, organized, structured, and traced to requirements. They must also be integrated with other life-cycle tools and artifacts. This is the only way to implement the traceability necessary to demonstrate compliance to auditors.

At the onset, achieving this level of maturity might require a cultural change for many organizations. However, this is an important step towards the implementation of the IBM BDD for Compliance solution.

Managing requirements helps you build the right thing and many other principles rely on good requirements: change management, validation, documentation, and deployment.

Supporting tools

Requirements management is supported by a number of IBM products.

Rational RequisitePro

RequisitePro is the central repository for all organization regulations, policies, and systems requirements.

Figure 3-6 shows how requirements are used as input to many others activities during the software development life cycle of an application.

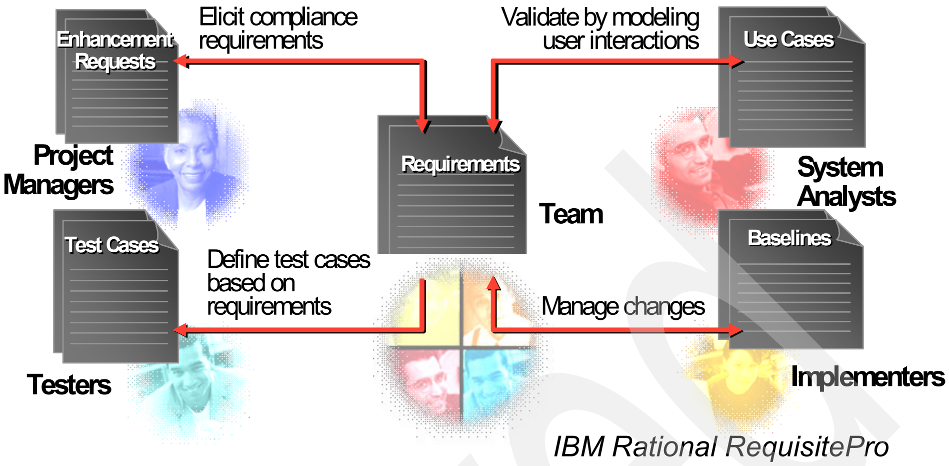

Enhancement Requests
Elicit compliance requirements
Validate by modeling user interactions
Use Cases
Project Managers
Requirements
Team
System Analysts
Test Cases
Define test cases based on requirements
Baselines
Manage changes
Testers
Implementers
IBM Rational RequisitePro

Figure 3-6 Life-cycle requirements traceability: key to documenting compliance

Using Rational RequisitePro, compliance requirements are captured in a central repository and validated by analysts through detailed use cases. These details are used by the development team to design and build the required controls into the appropriate applications. Based on the documented requirements, testers develop application test cases that thoroughly test that the application meets strict requirements. These can be manually described tests or automated tests that are captured once and repeated every time the application is changed in some way.

There is another challenge regarding compliance requirements: they change, mostly due to external changes in regulations, standards, and policies. For example, companies subject to a regulation are typically bound to comply with changes to these regulations within a specified period of times. Being prepared to perform impact analysis when such a change occurs is an important component of the solution.

What makes changing requirements complex to manage is not only that a changed requirement means that time has to be spent on implementing a particular new feature, but also that a change to one requirement may have an impact on other systems. Managing change includes activities such as establishing a baseline, determining which dependencies are important to trace, establishing traceability between related items, and implementing change control.

Figure 3-7 shows the example mentioned above. You can see the detailed relationships between the regulation, the company policy, the process features for the new model, the application features needed in the supporting systems, and the assignment of work to these projects by the change control board or project management office.

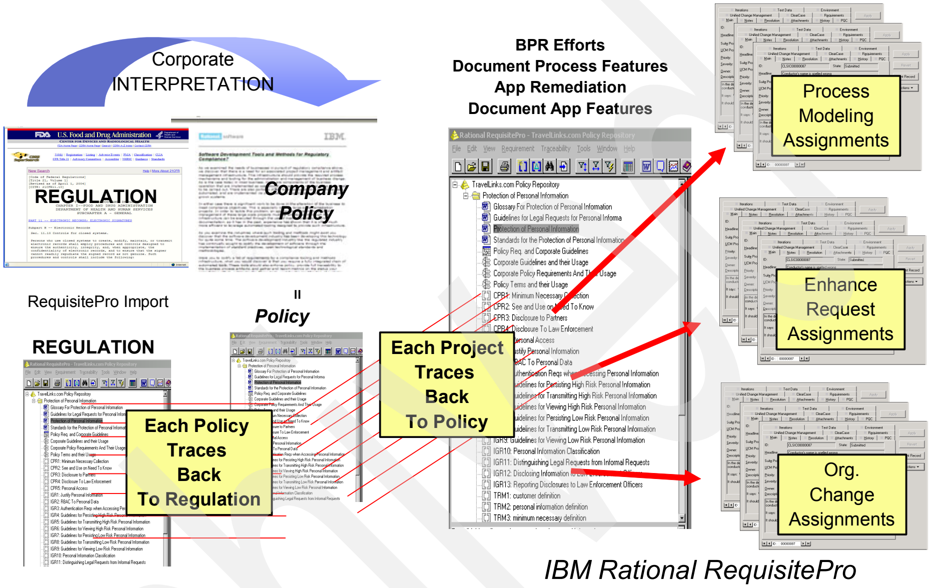


Figure 3-7 Requirements taxonomy for policy

Starting with the external regulations, the organization creates internal policies. Both are stored in IBM Rational RequisitePro and traced into features of supporting use cases. A change on any of these requirements can be easily tracked by RequisitePro and linked to every change request when using IBM Rational ClearQuest.

IBM Rational RequisitePro is a requirements and use case management tool for project teams that want to improve the communication of project goals, enhance collaborative development, reduce project risk, and increase the quality of applications before deployment. Some of the features are:

- Uses advanced integration with Microsoft® Word to provide a familiar environment for activities such as requirements definition and organization
- Incorporates a powerful database infrastructure with real-time Word document synchronization to facilitate requirements organization, integration, and analysis

- Enables detailed attribute customization and filtering to maximize informative value of each requirement
- Provides detailed traceability views that display parent/child relationships and show requirements that may be affected by upstream or downstream change
- Performs project-to-project comparisons using exportable XML-based project baselines
- Integrates with multiple tools in the IBM Software Development Platform to improve accessibility and communication of requirements

To learn more about Rational RequisitePro refer to:

http://www-306.ibm.com/software/awdtools/reqpro/

WebSphere Business Modeler

Often organizations need a broader business view of their operations to make informed decisions. Business transformation requires both knowledge of existing business processes and the ability to visualize alternatives. WebSphere Business Modeler enables you to capture the current business activities and workflows. You can use the tool to perform a simulation of the workflow to estimate cost and duration of a given path. In the same way you can simulate alternative scenarios to uncover business opportunities or to compare the cost of process changes to a baseline simulation. Once an opportunity has been identified, IBM solutions help you analyze business and technology requirements, perform impact analysis on existing systems, and scope projects appropriately.

By performing this step of business analysis, the supporting systems are usually more accurately defined and linked to business priorities. Many requirements and even systems architectural requirements are elicited from this step.

For IBM this link between business needs and support system requirements is realized through the integration between these tools.

WebSphere Business Modeler can export a business process model into development tools, such as WebSphere Integration Developer (for execution in WebSphere Process Server) and WebSphere MQ Workflow. This link from modeling by a business analyst to the IT implementation staff creates synergy between the business domain and the IT domain.

Supporting process

IBM RUP defines the task *Defining Automation Requirements* in the business modeling discipline, which specifies how to derive system requirements from the business modeling work products.

For more details about requirement management, see *Requirement Management and Business Modeling Disciplines* in IBM RUP.

Conclusion

Table 3-2 shows the control principles for requirements management.

Table 3-2 Control principles for requirements management

Control principle	Requirement management
Benefits	► Align applications with business needs and regulatory needs. ► Easily access the impact of implementing or changing a policy into supporting systems. ► Control over regulations and policies.
Pattern	► Create a central repository for regulations, standards, and polices. ► Trace regulations, standards, and policies to supporting systems. ► Establish control over changes in regulations, standards, and polices. ► Define, understand, and prioritize implementation.
Antipattern	► Manipulate regulations, standards, and policies as documents. ► Start their implementation without previous assessment of risks and benefits to business.
Evidence	Linkage between artifacts that align with the business process steps.

Software change management

Do what you say

Software change management is all about the implementation of controls and safeguards to ensure that changes made to business applications are done only by authorized individuals in a way so that these changes are auditable, traceable to related artifacts defining the change, and verifiable. In other words, change management ensures accountability of *who did what, where in the system, when it was done, and why.*

Discussion

Beyond the traditional change requests sources, such as end users and stakeholders, businesses must keep up with current and emerging laws, regulations, and standards. Consequently, organizations must realize that compliance is not a one-time project or activity. Compliancy is an on-going process that must be integrated as a first class business process.

Different changes on regulations, standards, and policies might demand changes in many different supporting systems, which, in turn, might be developed by different teams.

Compliance management requirements highly affect the productivity of software development teams. This scenario becomes more challenging if organizations have geographically distributed development teams.

Auditors want to see the appropriate evidence of IT internal control, showing that changes are developed through a secure, controlled, and auditable process with the following characteristics:

- *Software and asset data control*—Full inventory and safeguard storage of all software assets.
- *Software change control*—Comprehensive change histories detailing why each software change was made, what changed, who changed it, and when it was changed.
- *Separation of duties*—Important feature for preventing fraud. Some roles in the chain of development must be played by different people. The person who changes the code is not the same who approves the change, and is not the same who approves the code to be released to the production environment.

Due to the temporal nature of change, it can be very difficult to manage. Ensuring that many moving and changing parts are delivered simultaneously to construct a cohesive whole requires a great deal of rigor. Change management should be presented using a natural and intuitive interface that reduces the perception of complexity. If change is managed at a higher level of abstraction than simply *working on files*, then the task of managing change appears simpler. Consequently, the change management solution should be design to allow people to work on activities, and as a result of working on those activities all of the necessary assets are tracked, traced, modified, and audited.

This approach reduces the complexity of change management by allowing personnel to focus on what has to be changed, as opposed to the task of tracking all of the individual elements requiring the change.

Supporting tools

Companies operating in highly regulated environments require the implementation of a robust configuration management, which includes change request management, version control, automated work space management, and build management. To fulfill these requirements, the IBM BDD for Compliance solution is based on the use of both IBM Rational ClearCase and IBM Rational ClearQuest tools.

The activity-level abstraction capability is provided by the Rational Unified Change Management (UCM) process framework, which is contained within the ClearCase tooling. Using UCM, practitioners can focus on activities, such as *implement changes for policy 275*. UCM manages the association between this activity and software development artifacts that are created or modified as result.

These features are part of the foundation of the IBM BDD for Compliance solution and play a special role for organizations with geographically distributed teams for development. In such cases, no matter which role the organization is playing, vendor or contractor, having such an infrastructure in place is a must.

Software configuration management (SCM) is a key capability in modern software development practice. It allows teams to carefully manage all artifacts created in the software development life cycle, during which numerous changes occur, including changes to requirements, models, code, and so forth.

Rational ClearQuest

ClearQuest is an auditable workflow change request management solution, featuring electronic signature, audit trail, and user authentication. This tooling helps to ensure that all software changes are performed for a valid business reason by an authorized person. Use of this tooling ensures adequate documentation of actions taken against your artifact base in accordance with an organization's software development business controls.

ClearQuest also enables better insight, predictability, and control of the software development process. Through its flexible workflow management, defect and change tracking capabilities, ClearQuest helps to automate and enforce development processes, manage issues throughout the project life cycle, and facilitate communication between all stakeholders across the enterprise.

Here is a list of some of the features and benefits of the ClearQuest tooling:

- *Audit trail*—Capture and document evidence of who made a change, what they changed, and when they made a change.
- *Electronic signature*—Verify and document the identity of users to ensure that only authorized users may approve the transition of a change request from one state to another.

- *User authentication*—Ensure that ClearQuest passwords provide a reasonable level of security against various kinds of security threats.
- *User authorization and data control*—Ensure that only authorized users may make changes to controlled data.
- *Workflow management*—Define and enforce consistent, repeatable processes with flexible process definition and customizing.
- *Life cycle management*—Automate the capture of key information from requirements to testing related to controlled changes via product integrations.
- *Project management*—Project status, workload, and issues including defect submissions and enhancement requests can be easily monitored and prioritized through queries, charts, and reporting capabilities.
- *Regulation support*—Electronic signatures and audit trails help you to meet regulatory and audit compliance requirements.
- *Interfaces*—Local, remote, and Web interfaces enable access virtually anytime and anywhere.
- *SCM integration*—Integrated with Rational ClearCase and IBM SCLM Advanced Edition for z/OS®, ClearQuest provides a single point of control for enterprise change management.

To learn more about Rational ClearQuest refer to:

http://www.ibm.com/software/awdtools/clearquest/

Rational ClearCase

ClearCase is the central repository designed to provide life-cycle management and control of all software development assets. Unlike most other solutions, ClearCase is a file system that implements integrated version control, automated workspace management, parallel development support, baseline management, and build and release management. Because ClearCase is a file system, and not just a version control tool, it not only versions files, but actually versions the entire file system structure. In other words, ClearCase versions the entire directory tree structure in addition to all of the files supporting a system. Rational ClearCase provides all of the capability required to create, update, build, deliver, reuse, and maintain business-critical assets.

ClearCase can help increase productivity through parallel development, reduced build/release cycle times, and increased software reuse. Because ClearCase provides native integration with most of the leading IDEs including Rational Application Developer, Rational Software Architect, WebSphere Integration Developer, Microsoft Visual Studio® .NET, and the open source Eclipse framework, it further streamlines the development process by allowing practitioners to work with the tool they are most comfortable with.

ClearCase provides these features that benefit the solution in the following way:

- *Version control*—Provides the repository for securely storing and versioning all of the artifacts of the development life cycle.
- *Workspace management*—Have developers work in isolated code spaces yet coordinate their work with others transparently, in a native operating system directory that provides transparent access to versioned objects.
- *Parallel development support*—Version selection for building software, and record keeping to accurately reproduce a build.
- *Build management*—Version selection for building software, and record keeping to accurately reproduce a build.
- *Interfaces*—Local, remote, and Web interfaces enable access virtually anytime and anywhere.
- *Operating systems*—Linux®, Windows, UNIX, and mainframe (z/OS) development support enable enterprise-wide application and build development.
- *ClearQuest integration*—ClearCase is seamlessly integrated with Rational ClearQuest for a complete software configuration management solution.
- *UCM support*—The Unified Change Management (UCM) provides the ability to manage artifacts at a higher level by abstracting individual files and directories to higher level objects that actually represent your software architecture.

To learn more about Rational ClearCase refer to:

http://www-306.ibm.com/software/awdtools/clearcase/

Change request and configuration management integration

One of the greatest challenges many organizations face in the domain of change request management and configuration management is the need for integration of these two domains. Often organizations have implemented one half of the solution or the other half. However, to do either change request management and configuration management well, institution of both is really required. IBM Rational's solution in this domain is one of our strongest capabilities.

When used together, Rational ClearQuest and ClearCase provide an automated and auditable change management solution that helps ensure that software deliverables include only authorized files built using auditable workflow processes. Audited builds in Rational ClearCase help you prove to auditors exactly what is in the generated system version, how it was built and when, and who was responsible.

The combination and integration of ClearCase and ClearQuest unifies activities and artifacts. They provide activity-based UCM.

ClearQuest provides an easy way to assign project work to specific team members and rank the priority of the work.

For example, you are assigned to work on the special promotion Web page. An individualized to-do list appears on your desktop through ClearQuest. The entire team knows their responsibilities on the project and can track the status of the work in real time. From ClearQuest, you can automatically access ClearCase and review all the artifacts associated with the special promotion activity from the ClearCase database.

ClearCase conveniently and automatically gathers all the changes that you make to project files and other artifacts while working on the special promotion Web page and adds them to the activity's change set. Now that all of the artifact changes are associated with a particular activity, ClearCase can manage that change set as a single unit.

By associating activities and artifacts, users save time and are confident that they have all of the information they need to complete a task. This association also assures higher software quality. Practitioners cannot unintentionally *forget to check in a file* and affect the success of a build or a project. Through Unified Change Management, ClearCase takes what was once a manual, error-prone process, and automates the collection of change set data so that you can be sure that all changes related to an activity are accounted for.

Supporting process

The supporting process is defined in the *Change Management Discipline* in IBM RUP.

Conclusion

Table 3-3 shows the control principles for software change management.

Table 3-3 Control principles for software change management

Control principle	Software change management
Benefits	► Provide an auditable workflow management process, featuring electronic signature, audit trail, and user authentication support, to document compliance with your organization's software development business controls. ► Automate asset and change management. When used with Rational ClearQuest, provides auditable asset management to help ensure that software deliverables include only authorized files built using auditable workflow processes.

Control principle	Software change management
Pattern	► Team performs activities, following automated process for change and configuration process. ► Implementing authorization workflow, quality gates, separation of duties.
Antipattern	► Not organizing change request, not following an approval workflow, not prioritizing them, not assessing their impact on supporting systems.
Evidence	► Accountability on the chain of delivery. ► Documented and formalized handoffs and signoffs.

Build and deployment management

Do what you say and be able to prove it

If an organization is highly successful in auditing, tracking, tracing, and constructing an application, but fails to deploy the application into production, it is of no value to the business. However, in the compliance context merely deploying an application into production is insufficient as well. This is due to the need for full accountability of the construction of the application, as well as accountability of production bits during the deployment process itself. In this section we describe this requirement in more detail.

Discussion

From a compliance perspective, every application running in production must be traceable back to the original work products and artifacts used to assemble a given application. The reason for this degree of rigor goes back to our discussion in previous chapters regarding the need to show which requirements and regulatory mandates are satisfied by a given feature deployed on a target system.

This traceability from requirements to application and back to requirements is crucial for organizations to give answers to most common questions posed by an auditor. Absence of this kind of information makes it difficult to prove to an auditor that the system is exactly the same system that was developed to satisfy some specific policy mandate.

In Figure 3-8 we illustrate this bidirectional traceability in detail. Through this traceability we can see that all the activities and artifacts are linked together. The point is that an audit can pose the question starting from any artifact within the software development life cycle, and you still arrive at the correct answer.

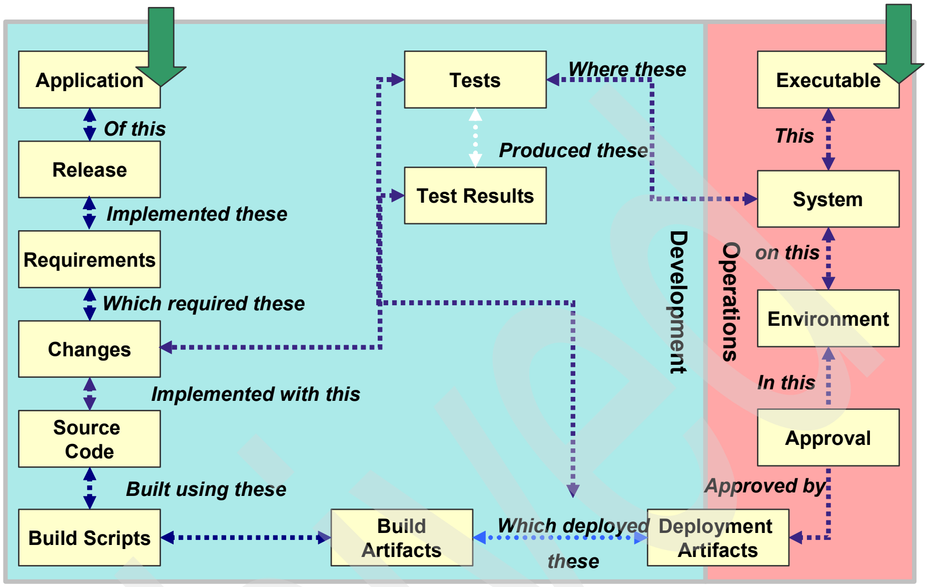


Figure 3-8 Which charges the executable in production contains

Auditors will look for much more information to know exactly *what* was delivered to the customer. Here are other types of questions, easily answered using this chain of traceability:

- Who approves the deployment of applications into test and production environments? Where is the audit trail?
- What version of build artifacts are deployed in the test environment? What about the production environment?
- Can you accurately reproduce all artifacts that you deliver to your customer?
- What build artifacts are associated with this release of the application?
- What version of the source is associated with this build artifact?
- Who deployed the code on this server? How is that controlled?

To ensure the integrity of this data it is important to implement proper security and the ability to electronically sign off at critical steps within the process.

The great challenge faced by many organizations is how to connect the development and operations areas all together, to give auditors the right answers.

This is overcome by establishing a build management process that establishes a repeatable workflow, enforced through tooling to implement traceability of builds and releases throughout the SDLC. The build management solution also gives updated information regarding the status for each build or release. This ensures timely and accurate tracking of related builds as well.

Organizations employing manual or home-grown processes for build management often view build management as a minor concern. This is due to the perception that testing activities are optimally carried out upon completion of a total systems build. Often organizations using this approach fail to realize that testing teams remain idle waiting for a release to be ready for validation. Because project deadlines tend to be fixed, validation or testing duration tends to be hit the hardest, as it is found at the end of a traditional development cycle. This can lead to an incompletely tested release of a system that can shut down the business operation for many hours or days, with very high cost to organizations.

In Figure 3-9 we illustrate a sample build workflow that an application follows from development into the production environment. Testing and validating of the system against its requirement and regulatory mandates is an important step in the approval flow for an application to proceed into the production environment.

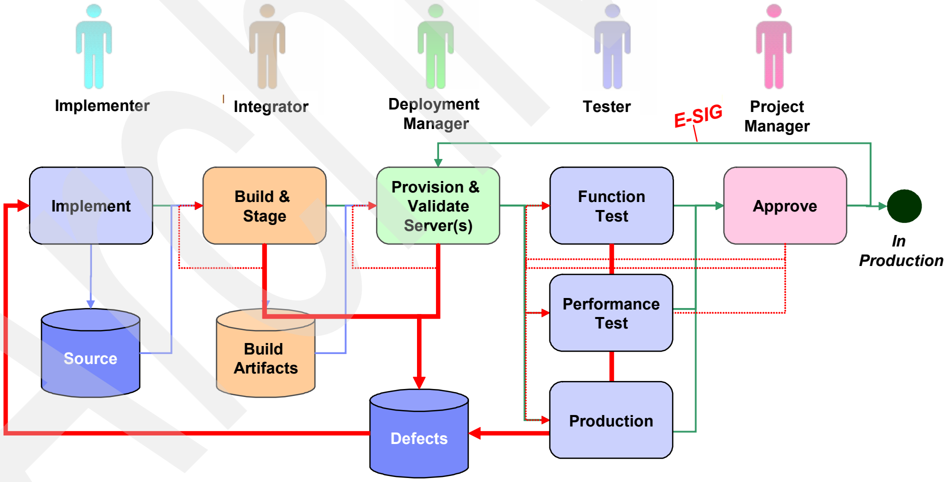


Figure 3-9   How changes go into production

The development team manipulates many project work products that are physically stored in the project repository, to create an operational version of an application, which we called *build* (refer to the definition for build in the *Configuration and Change Management* discipline in IBM RUP).

This operational version must be validated against regulatory requirements into functional and testing environments. Therefore, the team must create a subset of the end product that is the object of evaluation for these major milestones, and we call this a *release*. A release is a stable, executable version of the product, together with any artifacts necessary to use this release, such as release notes or installation instructions (refer to the definition for release in the *Configuration and Change Management* discipline in IBM RUP).

Once this release is validated and approved, it is ready to be deployed into production by the operations team. The operations team must have a complete deployment model to make sure that software and hardware requirements are satisfied for each workstation and server running the application.

To perform deployment well, a great deal of information must be captured, recorded, and shared during these tasks and among the different teams inside the organization. Participants should come from both the development and operation areas and include the roles of developers, testers, build managers, project managers, and deployers. Compliance demands a high level of exactitude of information. This level of exigency and precision is not achieved by a manual log of operations, and certainly cannot be achieved in a reasonable amount of time or expense through a manual process.

Supporting tools

Figure 3-10 shows IBM tools support for automating the build and deployment management process.

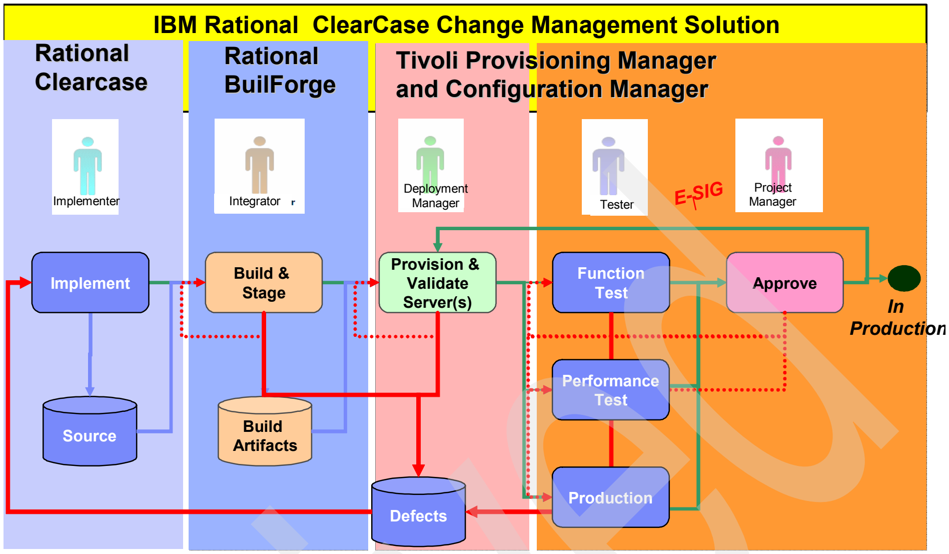


Figure 3-10 IBM solution for build and deployment management

The tools are:

- IBM Rational ClearCase manages source code, artifacts, and deployment units.
- IBM Rational Unified Change Management provides an out-of-the-box development process that integrates ClearCase artifacts with ClearQuest activities.
- This solution allows you to track builds and deployments through the test environment:
  - Track which build is to be used for testing.
  - Define and sequence test environments specific to your organization.
  - Establish approval gates before deploying to an environment.
- Pass regulatory audits:
  - Associate builds with deployments.
  - Capture electronic signatures when needed.
  - Maintain build artifacts under version control in UCM.
- Add build automation:
  - Integrate with BuildForge build automation packages.
  - Automatically create and update build records.

- Add deployment automation:
  - Link to Tivoli® Provisioning Manager to automate the provisioning of servers with the latest build.
  - Deploy approved build files directly from the source control.
  - IBM Tivoli Configuration Manager deploys software for distributed devices.

Rational BuildForge

BuildForge products provide complete build and release process management. They provide a framework that helps development teams to standardize and automate tasks and share information. Build acceleration capabilities reduce the process execution time, resulting in faster time-to-market. Enterprise reporting improves visibility into the build and release process. Process control and audit trails help meet compliance mandates.

BuildForge technology, when combined with other Rational software, can help organizations to automate, track, audit, and analyze their application development life cycle, often using the tools they have in place today. As a result, they can increase product quality, improve team efficiency, and move toward achieving sustainable compliance management.

IBM markets the following BuildForge products:

- *BuildForge FullControl*—Manages and controls builds and releases across the development life cycle
- *BuildForge FullThrottle*—Build accelerator that executes processes concurrently across server pools
- *BuildForge Prism*—Desktop IDE for developer self-service build activities
- *BuildForge Adaptors*—Integrates BuildForge products with third-party change and configuration management tools

To learn more about IBM Rational BuildForge refer to:

http://www-306.ibm.com/software/rational/buildmanagement/

Tivoli Provisioning Manager

Provisioning Manager allows you to create, customize, and quickly utilize best-practice automation packages. Pre-built automation packages provide control and configuration of major vendors' products, while customized automation packages can implement your company's datacenter best practices and procedures. These procedures can then be automated and executed in a consistent error-free manner. In fact, using these automation packages, Tivoli Provisioning Manager has the ability to provision and deploy a server with the single push of a button.

Some of the features of Tivoli Provisioning Manager are:

- Graphical user interface designed to simplify change execution tasks for the datacenter operator, hardware, software, and network resource discovery and drift detection to help ensure that desired configurations are maintained.
- Integration with Tivoli Configuration Manager for enterprise-wide software distribution, and image and script management to help leverage existing company standards and procedures in a consistent and controlled way.
- Provisioning Manager also incorporates solution install, a self-managing autonomic technology enabling the deployment of complex applications to multiple real and virtual servers.

To learn more about IBM Tivoli Provisioning Manager refer to:

http://www-306.ibm.com/software/tivoli/products/prov-mgr/

Supporting process

The supporting process is defined in the *Deployment, Configuration, and Change Management* disciplines in IBM RUP. A build produces an operational version of a system or part of a system that demonstrates a subset of the capabilities to be provided in the final product.

A build comprises one or more implementation elements (often executable), each constructed from other elements, usually by a process of compilation and linking of source code. A release is the delivery of a functional system meeting predefined objectives.

Conclusion

Table 3-4 shows the control principles for build and deployment management.

Table 3-4 Control principles for build and deployment management

Control principle	Build and deployment management
Benefits	► Consistent, reliable, high performance process for build and deployment management ► Eliminates multiples sources for manual and error prone in all SDLC cycle ► Increases team productivity and improves communication among different areas in organization
Pattern	► Processes and automates
Antipattern	► Manual control
Evidence	► Accountability in the chain of delivery

Reference

See IDC White Paper *Establishing Build Management for IT Efficiency and Business Adaptability,* January 2006:

`http://www.buildforge.com/index.php?option=com_wp&Itemid=122&paper=idc`

Software validation

Be able to prove it

Nobody likes to discover that they have application bugs in their software, particularly people who depend on it. This section discusses the importance of using requirements as a basis for software testing and validation and also discusses the importance of traceability to all of the system artifacts, such as testing results. The principle of software validation is not only a product quality indicator, but a development process quality indicator as well. For example, in most highly regulated environments you must be able to demonstrate that all compliance mandates have been accurately captured and implemented in your key applications.

Discussion

To demonstrate due diligence, both the functional and non-functional requirements that were spawned from the company policy must be implemented and the system must be validated against them. If there is one cardinal rule of systems audit, it is the demonstration of traceability between system requirement and system test results. One can then conclude by extension that system requirements spawned from policy requirements must align with system test results that substantiate that a system is fit for its intended use.

Creating these proof points is far more than validating the requirements. You must register and track each of the tests, as well as the corresponding test results. Upon completion you should have documented when and how these tests were conducted, and on which platforms the systems were validated.

Non-functional requirements validation is even more complicated. Creating and configuring test environments is sometimes impossible to achieve without supporting tools. Consider the challenge of how to validate a performance requirement of a simulation of 1,000 simultaneous users accessing a system?

Validating software is a key piece of the overall solution. Because the success of an implemented system is a reflection of the quality of the processes and the team that designed it, the system quality is perceived as a quality indicator for the whole development process that a company has in place. For example, not meeting a regulatory requirement might indicate that policies were not implemented properly. However, this is clearly a misnomer, as it is not always the code that is the problem.

Systems are more complex today, and validating them against all of these combinations of environments and user profiles requires a lot of resource allocation, not only human resources, but in the form of servers and workstations. Validation resources tend to be limited and usually shared among teams that perform the validation of many systems. Depending on the nature of the test, it is quite common for the creation of a test environment to be more costly than the implementation of the system requirement itself.

Once a system is validated, how are the test results communicated? How does the development team gain an awareness that there is a defect to fix? How will you quantify that a given system has achieved the minimum quality level specified by the operations team to permit a release into the production area?

Software validation is a key piece of the IBM BDD for Compliance solution, and tooling support is a key piece necessary to create the traceability needed among work products or artifacts.

The link between software validation and requirements management is stronger in the context of compliance and gives visible value if they are used as input for validation plans. Auditors do care about which specific requirements are being satisfied and how they were validated.

From the traceability perspective, it is important to show to auditors the link between regulatory mandates, test planning, and test results.

Such a level of complexity and required traceability among work products for auditors demands tool support. Manually testing might create bottlenecks, which can result in a negative impact on business operations, and potentially negatively impact the reputation of the business.

Supporting tools

Before executing any tests to validate a system, the organization must establish the scope of validation tests necessary to acquire the data to make the critical decisions about deploying an application. Included among the considerations are:

- Which regulatory requirements must be validated?
- Which functional and non-functional requirements are targeted to be tested?
- When?

- Which environment and platform will be used?

Everything begins with the construction of an appropriate test plan, typically assembled from the original design requirements for the system.

Rational ClearQuest Test Manager

IBM Rational ClearQuest Test Manager (CQTM) is Rational's newest test management solution. CQTM runs as a schema on top of the base ClearQuest product. The advantage of this architecture is the consolidation of the storage for test management artifacts along with the associated test results.

CQTM provides features to allow teams to create test plans and manage the testing process from the initial test case planning, through test development, execution of the tests, and analysis of the test results.

Some of the features of CQTM are:

- CQTM integrates with the following Rational products:
  - ClearCase
  - RequisitePro
  - Rational Manual Tester
  - Rational Functional Tester
  - Rational Performance Tester
- Adapters can be written to support third-party or custom test tools.
- CQTM supports the Eclipse Test & Performance Tools Platform Project (TPTP), a framework that contains testing editors, deployment and execution of tests, execution environments, and associated execution history analysis and reporting.
- Test scripts associated with CQTM test cases may be executed from within the CQTM client, or from within the test tool. When execution is performed from CQTM, results are automatically published to ClearQuest. When execution is performed from the test tool, results can be pulled into the ClearQuest database.
- CQTM features built-in queries, charts, and reports for coverage/suspicion, status, and trends over time. Users can also create custom queries, charts, and reports.
- Support for distributed teams is enabled via replication or through a Web interface.

To learn more about Rational test tools and testing solutions refer to:

http://www-306.ibm.com/software/awdtools/clearquest
http://www-306.ibm.com/software/awdtools/tester/clearquest/functionaltest/index.html

Rational RequisitePro Integration

Setting the requirements that will be the basis of testing: If you use IBM Rational RequisitePro for requirements management, CQTM allows you to use those requirements a basis for developing tests (Figure 3-11).

- The benefit of linking your tests to requirements is that you are able to verify that every compliance requirement is tested and you can report on the progress of testing against your list of requirements. Although you may find that you create additional test cases beyond those that test the application requirements, using the original requirements to guide your test plan is highly recommend when implementing a BDD for compliance process.
- Another important aspect addressed with this integration is about team communication, especially when a policy or compliance requirement changes.

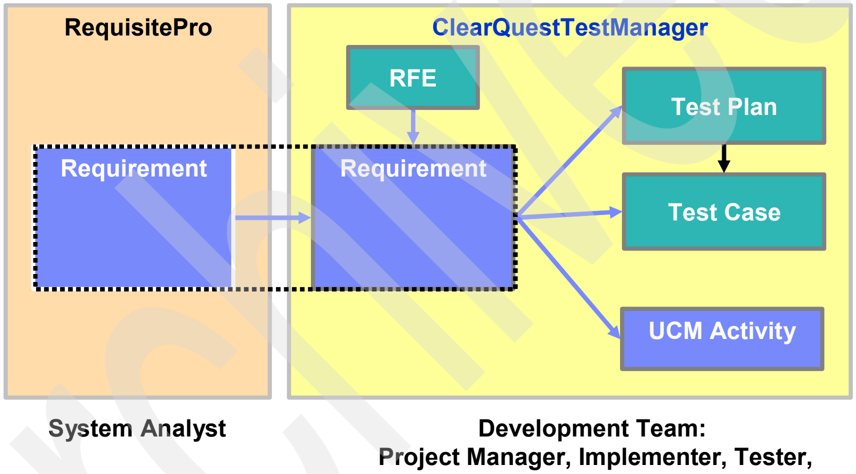


Figure 3-11 CQTM and RequisitePro integration

Usually requirements are managed by system analysts who work with customers to elicit and organize them. However, all members of the team require access to the requirements for the system including developers, testers, and all of the other roles on the team. Often these team members are geographically distributed, so communication can become even more of a challenge.

When the development team uses CQTM they can easily inquire about the last revision of the underlying requirements for a test, and assess the impact that changes to these requirements may have on these tests. Because test procedures must also be updated with any requirements that change, additional information about a requirement can be located by simply selecting the *View* option for the ClearQuest requirement. The point is that transparent access to any artifact can be gained from nearly any tool by leveraging the native integration provided by the solution.

Creating a test plan

A test plan defines the goals and objectives of testing, the items being targeted, the approach to be taken, the resources required, and the deliverables to be produced.

The test plan can be organized in a hierarchical group or testing framework for managing the numerous test cases you may have for an application or new feature. A test case contains the basic set of steps required to perform a test, including validation of those steps.

Test cases can be included in a test suite for sequential execution of a group of test cases. When a configuration is associated with a test case, it then becomes a configured test case allowed to be executed, creating results that are associated with the test case.

The relationship among these test artifacts is illustrated in Figure 3-12.

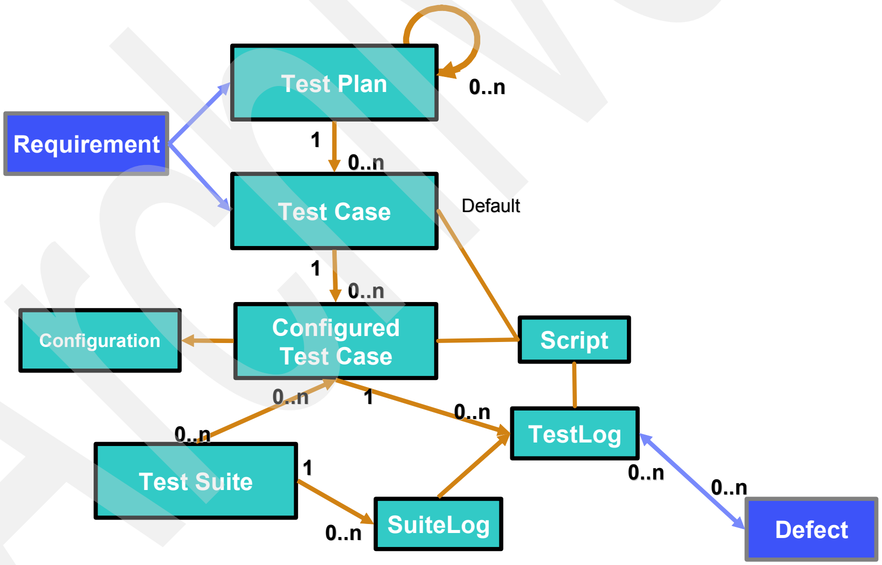


Figure 3-12 Simplified object model for test artifacts

Figure 3-13 shows an example of a test plan.

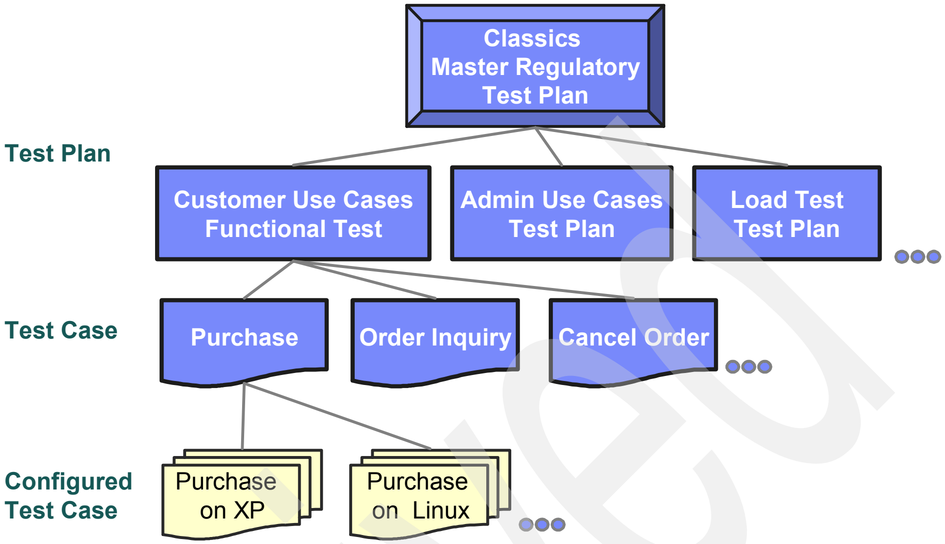


Figure 3-13 Test planning example

The technology allows test cases to be associated with a test configuration, thereby assembling a configured test case record (a child of the parent test case). This provides the ability to execute and report on test cases on a platform-specific basis. For example, the solution under test may be ready on Linux before it is ready on Windows. Similarly, a test case implementation may be different across platforms. Configured test cases provide a mechanism to manage these conditions so that they do not become a bottleneck in the testing sequence.

Test cases may be executed multiple times to establish trend data that can be useful during audit situations. CQTM accomplishes this by storing the result data from every run so that this data is available for comparative analysis and queries.

Developing manual and automated tests

Test automation tools, such as the Rational Functional Tester, allow development teams to create automated test scripts that you can attach to test cases (see also “Rational Functional Tester” on page 99).

If no automated test tool is being used to create test scripts, you can use the Rational Manual Tester to create manual test scripts (see also “Rational Manual Tester” on page 100). Manual test scripts contain a descriptive list of the steps to perform within the test and the expected and action application behavior anticipated to verify the performance of the test case.

Manual tests can also contain notes, attached files, and other information to make test execution, directions, and results reporting more precise. You do not have to use all manual or all automated tests. Mixing both types, manual and automated test script, in a single test plan is quite common. Quite frankly, if a business control contains both an automated and manual component to the control, both halves of the control must be exercised to substantiate that the control is fit for its intended use. Such may be the case where an application is dependent upon the judgement of an authorized user to proceed with a transaction.

Executing tests and analyzing results

An efficient method of test case execution is to create a suite that contains the test cases that the teams want to be executed, for a regulatory mandate to be validated, for example. A suite can be saved and repeated at any point in the future. Suites also provide additional capabilities, such as distributing tests to remote workstations for execution and allowing one to use functions such as randomization, synchronization, delays, groups, and so forth. Suites are also used to execute both functional and performance tests (see "Rational Performance Tester" on page 100).

When a suite is executed, you must specify the application build that you are testing, the log folder for storing results, and log names so that you can track the results of your tests against different builds of the application. A test log with the results of the test is created. If a test script contains multiple verification points, all of them must pass in order to have a pass for the entire test case. One verification point failure can result in a failure for the entire test case. This is true regardless of whether you are using manual or automated test scripts in your test cases.

In the Event log window you can scroll through the details and you should be able to see the actual results that you entered. In the Details tab you can see the cause of a failure. If this failure was the result of an automated test execution, automatic comparators are displayed for each verification point, highlighting the differences between the expected and the actual behavior of the application.

Monitoring test progress

As you create and execute tests against each successive build of your application, you want to keep track of your progress. The reports provided in Rational CQTM allows you to track the progress of defining test cases for your requirements, creating and associating test scripts with your test cases, and executing your test cases.

Integration with Tivoli

CQTM is integrated with Tivoli tools to:

- Ensure functional requirements—IBM Tivoli Monitoring and the OMEGAMON® monitoring suite can provide application design decisions and assist IT in planning the infrastructure needs to deploy the application.
- Ensure system performance—Business areas expect applications to meet or exceed service level agreements (SLAs). This can be monitored by integration between tools:
  - Rational Application Developer for WebSphere Software with Problem Resolution Toolkit

    http://www-306.ibm.com/software/awdtools/developer/application/
    http://www-128.ibm.com/developerworks/rational/library/05/523_prob/
  - Rational Performance Tester with Performance Optimization Toolkit

    http://www-306.ibm.com/software/awdtools/tester/performance/
    http://www-306.ibm.com/software/rational/toolkit/ipo_toolkit.html
  - Rational ClearQuest

    http://www.ibm.com/software/awdtools/clearquest/
  - Tivoli Monitoring for Transaction Performance (TMTP)

    http://www-306.ibm.com/software/tivoli/products/monitor-transaction/
  - WebSphere Studio Application Monitor

    http://www-306.ibm.com/software/awdtools/studioapplicationmonitor/

The integration between Rational and Tivoli tools provides an enhanced solution set combining analysis tools and data from both development and production environments. Key data is shared across organizational boundaries to enable traceability from concept through production deployment.

This capability brings important benefits to customers, such as increased reliability and reduced system downtime, deployment of more robust and secure applications, monitoring of expected service level agreements at minimum cost, and faster analysis of problems and faster introduction of updates.

Rational Functional Tester

IBM Rational Functional Tester is an advanced, automated functional and regression testing tool for testers and GUI developers who need superior control for testing Java™, Microsoft Visual Studio .NET, and Web-based applications.

- Provides novice testers with automated capabilities for activities such as data-driven testing.

- Offers advanced testers a choice of scripting language and industrial-strength editor—Java in Eclipse or Microsoft Visual Basic® .NET in Visual Studio .NET—for test authoring and customization.

To learn more about IBM Rational Functional Tester refer to:

`http://www-306.ibm.com/software/awdtools/tester/functional/`

Rational Manual Tester

IBM Rational Manual Tester is a manual test authoring and execution tool for testers and business analysts. Available standalone, it is also delivered with Rational Functional Tester.

To learn more about IBM Rational Functional Tester refer to:

`http://www-306.ibm.com/software/awdtools/tester/manual/`

Rational Performance Tester

IBM Rational Performance Tester is a multi-user performance testing tool for any team needing to validate Web application scalability before deployment.

To learn more about IBM Rational Performance Tester refer to:

`http://www-306.ibm.com/software/awdtools/tester/performance/`

Supporting process

The supporting process is defined in the *Test* discipline in IBM RUP.

Conclusion

Table 3-5 shows the control principles for software validation.

Table 3-5 Control principles for software validation

Control principle	Software validation
Benefits	► Regulatory mandates are captured and validated. Validation evidences are automatically generated through tools integration.
Pattern	► Using regulatory mandates and other requirements as input to test plan. ► Using tools to create traceability among requirements, test planning, and test results. ► Using tools to configure test environments.

Control principle	Software validation
Antipattern	► Not using traceability among artifacts. ► Not formally assessing validation results before deployment to production. ► Not implementing separation of duties.
Evidence	► Documented test results to requirements linkage. ► SDLC process maturity and stability.

Validating your compliance process

Leverage compliance for business advantage

We begin our next segment by talking about astronauts. Why astronauts? Astronauts are required to perform specific procedural checks to ensure that all of the members of the team understand their roles and responsibilities during a crisis. Although this process is commonplace for astronauts, most view these procedural checks as an extreme measure (unless, of course, you were the one going up into space).

In the same way, validation of your compliance process helps to ensure the integrity of the process itself, and can aid in exposing those critical points of failure during high-pressure execution. However, there are substantively more pragmatic reasons that companies should consider this approach: capability maturity and protecting the interests of stakeholder.

According to the Software Engineering Institute at Carnegie Mellon University, there exists a continuum of software development performance and maturity that can be measured, entitled the Capability Maturity Model Integrated (CMMI). Companies that attain this highest level of maturity, Level 5, are said to be *optimizing* their software development practice by critically examining the measures of their performance. Organizations at CMMI Level 5 are among the highest performing software development shops in our industry.

The other reason for validating your compliance process is to protect the interest of your stakeholders, some of whom are your customers. Consider the impact to your organization's reputation every time a bug is discovered in a release of your application to production. In many other highly regulated industries and areas of business, there is zero-tolerance for mistakes.

Discussion

Let us begin with a definition of *validation*. Validation can be defined as "establishing documented evidence that provides a high degree of assurance that a specific process will consistently produce a product meeting its predetermined specifications and attributes."

The key phrases are:

- Predetermined specifications
- Documented evidence
- Consistent product
- High degree of assurance

The main requirement that a compliance framework must satisfy is helping organizations to prove that they do what they say they do to be in accordance with their policy and procedures.

Therefore, there are three aspects to be validated:

- Have you fully documented your existing software development process?
- Do practitioners actually adhere to the documented process throughout the organization's teams on all projects?
- Does the process yield the expected results of quality and performance, as demonstrated by the product quality and organizational effectiveness?

If you consider the development process itself a product of governing *the process of software development*, you realize that it requires validation just like the software product it produces. This means that the software development process must be validated against the original business needs that motivated executives to invest in establishing such a process in the first place. Using a requirements analysis tool, the business requirements can be mapped to the components of the development process to illustrate compliance with documented policies, methods, or goals it was design to achieve. To utilize this approach, it is important to have established the success criteria for the process, prior to process design.

However, this validation will not always be a manual step. To ensure that practitioners actually adhere to the documented process, certain components of the software development cycle can be automated. This is one of the strengths of the IBM Rational BDD for Compliance solution. Through the use of our automated tooling, practitioner behavior can be guided, thus ensuring that the documented procedure is actually adhered to. For example, management of change requests can be automated through the use of ClearQuest, as described earlier. Because the workflow is configurable, adjusting the workflow to match the documented workflow ensures that practitioners adhere to the process. Because it is impossible to advance artifacts through the software development life cycle, without satisfying the criteria defined in the configured ClearQuest record life cycle, practitioners have no choice but to adhere to the documented process.

If you take the additional step of validating this configuration by using automated testing or manual testing tools, you ensure the quality of this configuration by comparing it with the configuration design goals outlined in the software development methods. Thus, expected results of quality, performance, and organizational effectiveness are apt to improve significantly.

There are other powerful side effects of implementing such a solution that bear mentioning. These side effects include the establishment of a shared understanding of the way in which work should be performed, a common language with which to describe tasks, as well as consistent behavior across the organization. The remaining benefits of this approach trickle out in hidden capabilities of the organization to be more effective. Such an approach enables the business to easily move staff from one development organization or project to another. This is due to the common understanding of the development process held by practitioners. This allows for more effective cross training from a business domain perspective, because no cycle time is wasted acquainting staff with the differences in software development procedures.

Ability to establish a compliant development process

A compliant development process includes:

- Approval checkpoints (both project and technical)
- Quality gates
- Separation of duties

We can use the Rational Method Composer (RMC) to create these definitions and to create a customized process for organization. As stated earlier, IBM Rational Method Composer is designed to allow customers to either tailor the Rational Unified Process or to capture their own processes in a standard format. The tooling is a based upon the Unified Method Architecture (UMA) as recognized by the Object Management Group (OMG), which uses UML to internally store documented processes.

Leveraging this technology allows customers to easily extend documented processes through the application RMC plug-ins. Since each plug-in is designed with UMA, which is UML based, it inherits all of the object-oriented advantages of UML. Thus, any existing processes documented using this technology can be easily extended by simply inheriting the characteristics of other processes or plug-ins. For example, customers wishing to extend their existing process could download and apply a plug-in for service-oriented architecture (SOA), and have their process inherit all of the activities, work products, roles, and processes defined by such a plug-in.

In terms of IBM Rational, this is easily accomplished by using the RUP Plug-in for Compliance Management with Rational RMC to instantiate a BDD for compliance process, using the required policies, process standards, and frameworks or other criteria as input to the process design.

The goal is to assemble a process that satisfies our design criteria by incorporating appropriate approval checkpoints. When automating the development process approval checkpoints, you can enforce the approval behavior through the electronic signature capabilities found in the IBM Rational ClearQuest product. This ensures that the technical approval is in place for your process. The project level approvals for funding, release of budget, and other business aspects of the process can be enforced through workflow found in the Rational Portfolio Manager solution, when proposals are promoted to projects.

Finally, technical quality gates can be enforced by adding steps for the validation and sign-off during the software development life cycle using CQTM and electronic signatures. This ensure appropriate product quality evaluation, for example. Because access to all of these artifacts is constrained by user login, appropriate separation of duties is a natural outcome.

After documenting your software methods Web site using RMC and automating the workflow with tool-directed behavior, you can be assured that practitioners will adhere to the documented process. The generated Web site allows auditors, practitioners, and inspectors to navigate all process definitions, either designed or inherited. Refer to the IBM RMC and RUP Plug-in for Compliance Management for further information.

Ability to enforce a compliant development process

Enforcing a compliant development process includes:

- Automated workflow
- Authentication
- Traceability
- Electronic signature
- Separation of duties with respect to user rights administration
- Build management

Typical metrics are:

- How many projects are applying the process?
- How many people have been trained so far?
- How many projects are in the inception, elaborate, construction, and transition phases?
- How many people are using which tools?
- Ability to monitor and report software development controls status within broader IT-wide and corporate-wide controls management systems.

Ability to provide the expected information as required

Because this product and its process can be used as a framework to build other process systems, these resulting systems might be used to validate the quality of process: Are these systems implementing regulatory mandates as required?

Typical questions include:

- How many projects are using the process and are targeted for compliance audits?
- How many projects have been audited?
- How many projects passed?
- How many projects failed? What is the main reason for failure?

Understanding the causes for failure helps the process improvement. They are the main targets for the next version.

Such a process is not implemented in a single step. Realize that your software development life cycle must be maintained as a living document. The process must be continually enhanced, assessed, and validated against business needs, desired documentation of due diligence, and user feedback to implement new refinements.

Enforcement of the development process allows organizations to more effectively deal with the cultural aspects of process change. Tools that offer help, such as a fully hyperlinked Web site of the process for every role in the organization, or automated tool direct behavior, can ease process adoption by empowering individuals to investigate and learn for themselves. This does not diminish the need for training on how to use the process and tools. Defining a implementation plan, which includes training and mentoring sessions, is recommended.

For more information about services to implement a compliance solution refer to:

http://www.ibm.com/software/rational/services/professional/index.html

Supporting tools

The IBM Rational supporting tools are the IBM Rational Portfolio Manager, the IBM Rational Method Composer, the Rational Unified Process (RUP), and the RUP Plug-in for Compliance Management.

The Rational Portfolio Manager is described in "Rational Portfolio Manager" on page 71.

Rational Method Composer

Rational Method Composer is a flexible process platform containing processes and tools for use throughout IT life-cycle management (ITLM) that will help you deliver customized yet consistent process guidance to your project teams and IT organization:

- It is the next major release of the IBM Rational Unified Process and represents a quantum leap forward in providing a process solution that goes beyond software development.
- Rational Method Composer is easy to use and enables you to configure and present process content in a way that works for your team. It also consists of an Eclipse-based method authoring and publishing tool.
- Provides extensive process content libraries, including all the existing RUP content, as well as Rational-developed RUP plug-ins and also content in other areas such as portfolio management.

To learn more about Rational Method Composer refer to:

http://www-306.ibm.com/software/awdtools/rmc/

Rational Unified Process

IBM Rational Unified Process (RUP) is process guidance content included in the Rational Method Composer framework that delivers proven best practices in a configurable architecture.

The RUP process framework with IBM Rational Method Composer includes:

- A process content library based on the best practices adopted in thousands of projects worldwide. To address all of the process needs of organizations you can reuse what works for other organizations through the content library, rather than inventing everything from scratch.
- Capability patterns that allow project managers to rapidly add or remove reusable chunks of processes addressing common problems. Since no two projects are alike, project managers have to rapidly modify the process to address the specific project needs. This can be done through capability patterns, plug-ins, and process components, allowing content around various

domains, such as database modeling or advanced requirements management, to be added or removed.

- Out-of-the-box delivery processes to provide the project manager with a quick starting point for planning and initiating a project. A delivery process will provide an initial project template, identify what type of milestones to have in the project, what work products are delivered by each milestone, and what resources are needed for each phase.

To learn more about RUP refer to:

http://www-306.ibm.com/software/awdtools/rup/

RUP Plug-in for Compliance Management

The RUP Plug-in for Compliance Management provides guidance in solving the complex issue of complying and implementing organizational policies related to software development procedures and industry process standards. The plug-in assists customers in defining and implementing a compliant software development process that addresses the separation of duties, control points, audit reports, and authorization issues found in this process design space. The goal of this plug-in is to aid customers in implementing solutions that address each of the key factors used in the institution of a compliant process.

To learn more about the RUP Plug-in for Compliance Management refer to:

http://www.ibm.com/developerworks/rational/downloads/06/rmc_comp_mgt/

Supporting process

The process component of the IBM BDD for Compliance solution is itself an instantiation of the RUP process and might be configured and tailored following the guidance in the *Environment* discipline of IBM RUP.

Conclusion

Table 3-6 shows the control principles for compliance validation.

Table 3-6 Control principles for compliance validation

Control principle	Validating your compliance process
Benefits	► Leverage compliance for business advantage.
Pattern	► Identify business needs that must be addressed by the process. ► Prioritize them. ► Define and apply principles for each need. ► Have an owner for the process.

Control principle	Validating your compliance process
Antipattern	► Try to do everything at once. ► Do not automate it. ► Overwhelm team with manual documentation.
Evidence	► Documented SDLC process for creating all business artifacts. ► Process linkages to automated process supported by tool mentors.

Documenting results

Be able to prove it

Very few people derive a great deal of joy out of surprise in the business world, least of all in a regulated environment. Consequently, we design a significant amount of procedure to contend with the most likely eventualities. So why do most organizations operate as though there is no such thing as a surprise audit or inspection? Generally, shops that do not use our type of automated traceability find themselves doing an archeological dig for the artifacts needed for audit every time one occurs.

This section articulates the importance of the automating the documentation in opposition to annual creation. Whenever a system changes, the documentation must de updated to reflect the change.

Many regulations, standards, and policies require a demonstrated ability to show linkages between the feature requirements of a system and the test results that conclusively illustrate compliance with those feature requirements. This need is particularly true when the automated system provides, houses, or stores regulated data. Failure to provide such information will result in an inspection failure. This is because without this data, there is no way to prove that the system is fit for its intended use.

Discussion

As we established earlier in this book, any solution for compliance must help companies to prove they are in compliance with a regulation, standard, or policy based upon the documented evidence for organizational behavior and due diligence in compliance enhancement efforts. To substantiate such a position, a company must be prepared to respond to requests for this information in a reasonable time frame.

However, experience has proven that those that can provide this information in an instant appear to have better control over their processes. This need might lead organizations to implement a *documented driven development for compliance solution* instead of implementing a *business driven development* solution.

Manually generating documentation for compliance purposes is a huge and time-consuming task. Ideally, required documentation should be generated as the result of tool usage, as practitioners perform each of their respective tasks and activities.

Traditional manual software documentation methods provide a high degree of flexibility. However, this method frequently does not offer the same level of timeliness, accuracy, and document linkage that is needed for compliance that an automated solution can provide. This is because manual documentation is extremely labor intensive, is very difficult to maintain, and due to human error can threaten the consistency required across document boundaries. The challenge is that manual documentation efforts typically cannot keep up with the rate of on-going changes that need to occur to keep documentation up-to-date.

As you begin to document your business using automated tools, instead of hand-crafted documents or pictures, you discover that achieving the level of traceability required for compliance is relatively easy. This is due to tool features that support the IBM BDD for Compliance solution documented in the Rational Unified Process plus the RUP Plug-in for Compliance Management.

This process is something we have discussed earlier, tool-directed behavior. Tool-directed behavior is the implementation of process requirements in each tool, such that the process cannot be circumvented. If you think about it, this is something that businesses have been doing for quite some time. This is to say that businesses have been automating their business processes as tool-directed behavior in the application packages they select, or by custom programming systems to carry out transactions as the business dictates.

Because the relationships between all of the artifacts are already there, all you must do to create appropriate documentation is to generate a suitable report that walks the chain of traceability, dumping all of the required information. Generally, the reason for handcrafting such documentation is for formatting purposes. The sophistication of the automated documentation tools available today provide for detailed formatting constraints. These tools even provide for the generation of this documentation into multiple formats simultaneously, such as HTML, XML, and of course Microsoft Word.

Supporting tools

The solution components that support automated documentation include both Rational ProjectConsole™ and the Rational Software Documentation Automation tool (SoDA®). These tools do exactly as described above. Using each of the respective product APIs, each reporting product can access all of the artifact repositories for exposure of the artifact details. The solutions can be used to assemble charts or documents to formulate nearly any style report to serve the needs of the stakeholders.

Rational ProjectConsole

The ProjectConsole enables the development team to automatically quantify the current project status and assess development trends. Measurements are automatically collected from the Rational development environment and selected third-party tools, and then stored in a data warehouse.

The Rational ProjectConsole hyperlinks all artifacts together and displays the analysis in charts, indicators, and tables as your team Web site is automatically created and updated.

The ProjectConsole automates reporting on project status, dynamically creating a project Web site with a graphical dashboard based on data you collect. This reduces the time to build, update, and maintain a team Web site, plus the time and effort of manually gathering status updates.

The ProjectConsole collects both standard and custom metrics from your Rational Team Unifying Platform™ (TUP) and third-party products, presenting the results graphically so that you can easily assess project progress and quality. This allows you to better predict which areas will require special attention and where to focus scarce resources to stay on schedule. More importantly, ProjectConsole enables you to make decisions based on quantitative analysis, rather than subjective status reports.

To learn more about Rational ProjectConsole and Team Unifying Platform refer to:

```
http://www-306.ibm.com/software/awdtools/team/client/
http://www-306.ibm.com/software/awdtools/team/
```

Objective project status

By viewing the graphical dashboard, team members can quickly discern the true status of the progress and quality of their project. The Rational ProjectConsole provides all members with the ability to analyze individual development discipline activities (for example, modeling, coding, testing) and drill down to low-level details.

You can also visualize the planned-versus-actual measures, trace historical data trends, and view cross-discipline measures to get a better view across the entire project. These capabilities enable your software development team to take prompt corrective actions and realize the cause for late deliverable.

Rational SoDA

Rational SoDA is a project-wide documentation automation tool, and is part of the supporting foundation—the IBM Rational Team Unifying Platform (TUP). Its interface leverages well-known, powerful publishing tools. SoDA generates documents by extracting the requested data directly from the tool's data repositories.

Some of the features of SoDA are:

- It automatically generates documents and reports in HTML format.
- Its templates encourage the standardization of document types within a project or throughout a company. It is customizable to comply with individual project's standards.
- It regenerates precise, up-to-date documents easily. It preserves additional data entered directly into the document.

To learn more about Rational SoDA refer to:

http://www-306.ibm.com/software/awdtools/soda/

Supporting process

Documentation is provided as reports and documents:

- *Report*—An automatically generated description, describing one or several artifacts. A report is not an artifact in itself. A report is in most cases a transitory product of the development process, and a vehicle to communicate certain aspects of the evolving system. It is a snapshot description of artifacts that are not documents themselves.
- *Document*—A document is a collection of information intended to be represented on paper or in a medium using a paper metaphor. The paper metaphor includes the concept of pages, and it has either an implicit or explicit sequence of contents. The information is in text or two-dimensional pictures. Examples of paper metaphors are word processor documents, spreadsheets, schedules, Gantt charts, Web pages, and overhead slide presentations.

Conclusion

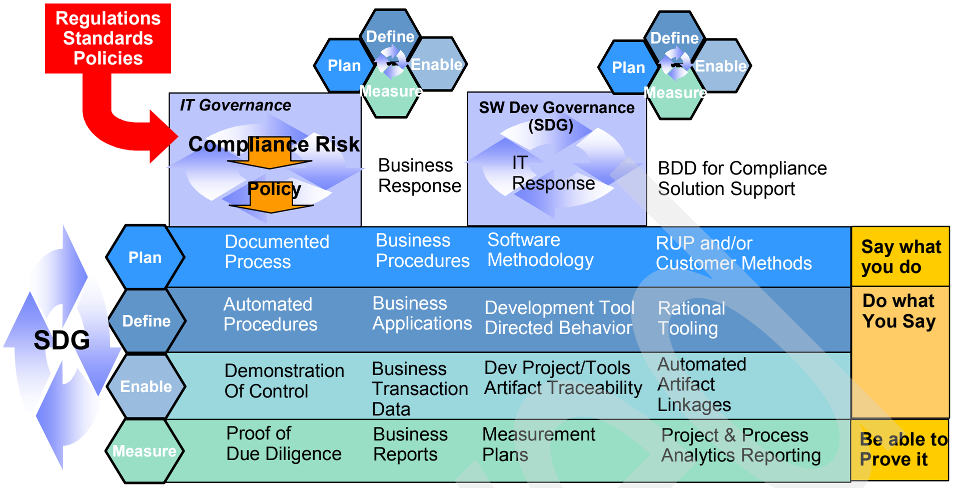
Table 3-7 shows the control principles for documenting results.

Table 3-7 Control principles for documenting results

Control principle	Documenting results
Benefits	► Objective project status. ► All team members can generate comprehensive reports that span across products. ► Documents are never obsolete. ► Always reflect up-to-date status of the system. ► Facilitates collaboration across teams. ► Support to business decision.
Pattern	► Create templates for documents and Web site. ► Real-time documents and reports.
Antipattern	► Manually creating and updating documents.
Evidence	► Transparency of reporting. ► SDLC process maturity and stability. ► Adherence to the defined process by artifact creators/SDLC. ► Linkages between supported tools using tool-directed behavior compliance.

Defining a customized solution for an organization

The IBM BDD for Compliance solution is designed as a modular offering. Due to the extensive nature of this offering, it should be implemented using an iterative approach. This allows organizations to address their highest compliance concerns first. Organizations considering the implementation of this solution should take stock of their compliance issues. After assessing the condition of the business, organizations can select the specific components and tooling to implement a personal solution, as shown in Figure 3-14.



Plan	Documented Process	Business Procedures	Software Methodology	RUP and/or Customer Methods	Say what you do
Define	Automated Procedures	Business Applications	Development Tool Directed Behavior	Rational Tooling	Do what You Say
Enable	Demonstration Of Control	Business Transaction Data	Dev Project/Tools Artifact Traceability	Automated Artifact Linkages	
Measure	Proof of Due Diligence	Business Reports	Measurement Plans	Project & Process Analytics Reporting	Be able to Prove it

Figure 3-14 Compliance problem and solution spaces

Although the IBM BDD for Compliance solution is a complete framework, organizations are able to select those components that best fit their immediate needs for implementation. The solution is designed for incremental adoption, such that each component can be added over time, and yet provide seamless integration to the previously existing components implemented. No matter which tool is selected, the process component always plays an important role, providing guidance on how to use these components all together. Customers can, and have, executed initiatives to implement the entire solution as phased implementation engagements.

Three-phases approach

The minimum number of phases to accomplish such an implementation is three, although it is possible to have as many as ten phases. The difference between these approaches is simply a function of the number of tools selected and implemented at any one time. Success depends upon the technical sophistication of the team, the degree of compliance that currently exists, and the organization's need to change.

Phase I

During a three-phases approach, in phase I you establish a project management plan for the execution of the tasks required to assess your current infrastructure, relative to the current best practice in each of the RUP disciplines.

Your assessment should consider the following areas:

- Software development life-cycle methods and consistency across the organization
- Program and portfolio management best practices
- Requirements gathering and management best practices
- Architectural analysis methods and systems analysis and design
- Application modeling and simulation
- Change request and configuration management best practices
- Build management and environmental configuration
- Systems testing and validation
- Release management and application deployment

The objective is to take an inventory of current practices and compare them to existing industry best practices to evaluate what is working and what might stand for improvement. Provided that there is a desire to begin documenting a new integrated process, this is the time to initiate such an effort.

Phase II

Once sufficient information is gathered from the investigative phase, you can begin to prepare for phase II. During phase II a subsequent project plan is constructed based upon what has been learned, to begin assembly of your integrated compliant process design, and to begin establishing the tooling infrastructure required to support that process. This can be accomplished through one of two methods, typically:

- It can be initiated by consuming the default configuration of Rational products and consumption of the Rational Unified Process as the standard process for the organization.
- Alternately, the tools can be configured to support any existing process, although one should expect that those existing processes that do not conform to current best practice should not be tolerated.

To ensure appropriate deployment of the Rational tooling, the following areas require further investigation and planning based on current organizational activities, release schedules, and so forth:

- *Environment specification*—IBM should work with your management team, administrators, and IT personnel to assess the computing infrastructure; determine where to install software components; and document these decisions in an environment specification report.

- *Usage model definition*—IBM should work with the management team and designated personnel to define how the solution is to be used on specified pilot projects, and document these decisions in the usage model report.
- *Installation and configuration*—IBM should also work with the management team and administrators to install the software components for the prototype solution according to the environment specification report and configure the tools according to the usage model report.
- *Rollout to end-users*—IBM will work with the designated members of the team to train and mentor end users on how to use the solution in their daily work, including not only the basic tool usage, but to incorporate specific workflow training regarding any new processes.
- *Administration planning*—Finally, IBM should work with members of the IT staff and the designated Rational tool administrators to plan for the administration and maintenance of the IBM Rational environment.

Phase III

Phase III is an optional component to the process. During phase III we revisit those areas where the technology is being utilized, and critically examine whether the goals and objectives defined for the implementation are being satisfied. In the event that the organization has found an issue with the operation of the solution, IBM can be called upon to evaluate these issues and discuss approaches for mitigating those risks that have been identified.

Areas examined during a phase III engagement include:

- *Adoption*—How well has the process and associated tooling been adopted?
- *Usage*—How well is the organization adhering to the documented process, and how well is the configuration supporting that process?
- *Practices*—For those practices that cannot be enforced through tool direct behavior, how well is the organization doing in adhering to the documented best practices for these activities?
- *Environment*—How well is the current environment holding up to the expanded usage of the tooling?
- *Administrative practices*—Do the current administrative practices support the infrastructure, as well as the goals and objectives established for the project?

The result of this engagement is the creation of a report outlining the key challenges identified in the use or operation of the solution, along with short-term and long-term recommendations for their mitigation.

Table 3-8 shows some possible combinations of principles and tools that can be selected to deliver the desired evidences.

Table 3-8 Providing evidence through automation of control principles

Principles	Automated by tools	Features	Provide evidences
IT governance	► Portfolio Manager	► Track change progress. ► Manage risk.	► Evidence of SDLC process maturity and stability ► Transparency of reports
Requirement management	► WebSphere Business Modeler ► Monitoring, RequisitePro	► Business needs better mapped into supporting systems. ► Regulations, standards, policies and requirements central repository. ► Change management to policies and regulations. ► Record and trace application requirements for compliance mandates.	► Linkage between artifacts that align with the business process steps
Software change management	► ClearCase+ ClearQuest	► Audit trails. ► Electronic signatures. ► User authentication, authorization, and data control. ► Defect and change management. ► Baseline, reporting, build auditing, project policies and triggers.	► Accountability on the chain of delivery ► Documented and formalized handoffs and signoffs
Deployment management	► ClearCase + ClearQuest + Tivoli Configuration Management	► Automated and controlled deployment.	► Accountability on the chain of delivery ► Documented and formalized handoffs and signoffs ► Documented SDLC process for creating all business artifacts
Software validation	► RequisitePro + CQTM + Manual Tester ► Functional Tester + Performance Tester	► Creates test plans based on requirements. ► Automates functional and performance requirements validation.	► Documented test results to requirements linkage

Principles	Automated by tools	Features	Provide evidences
Validating your compliance process	► Method Composer + Compliance and PSM Plug-in	► Provides guidelines, best practices, and common process. ► Communicates workflow and interactions through diagrams. ► Browser-based, providing accessibility for all users.	► Documented SDLC process for creating all business artifacts ► Process linkages to automated process supported by tool mentors
Documenting results	► Portfolio Manager + SoDA + ProjectConsole	► Define data collection, analysis, and reporting procedures into indicators and reports that are automatically organized and updated on the Web or in Word documents. ► Data collection for reports and documents in a non-intrusive manner. ► Well updated and valid information for decision makers. ► Improve communication by sharing a common project and initiatives, status, and information among team members.	► Transparency of reporting ► SDLC process maturity and stability ► Adherence to the defined process by artifact creators/SDLC ► Linkages between supported tools using tool-directed behavior compliance

Each solution might be customized for each customer. To accelerate its adoption IBM offers a service package described at:

http://www.ibm.com/software/rational/services/professional/index.html

The level of formality for these process components might also vary according to the organization's characteristics and implementation strategy followed (Figure 3-15).

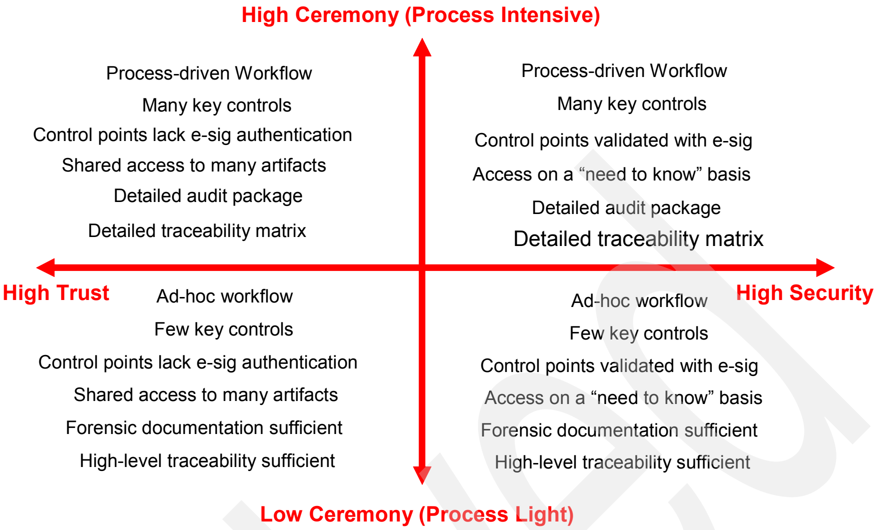

High Ceremony (Process Intensive)
Process-driven Workflow
Many key controls
Control points lack e-sig authentication
Shared access to many artifacts
Detailed audit package
Detailed traceability matrix
Process-driven Workflow
Many key controls
Control points validated with e-sig
Access on a "need to know" basis
Detailed audit package
Detailed traceability matrix
High Trust
High Security
Ad-hoc workflow
Few key controls
Control points lack e-sig authentication
Shared access to many artifacts
Forensic documentation sufficient
High-level traceability sufficient
Ad-hoc workflow
Few key controls
Control points validated with e-sig
Access on a "need to know" basis
Forensic documentation sufficient
High-level traceability sufficient
Low Ceremony (Process Light)

Figure 3-15 Creating the right process for your organization

As more and more customers recognize the value of governance and compliance, they are increasingly interested in moving from environments and operational models in the bottom left quadrant to the upper right quadrant where there is a higher degree of both security and ceremony.

The quadrant that any particular organization will select is dependent upon their particular business needs and available resources to execute. Those who have not done so in the past should consider adopting an iterative approach to implement process solutions like these. The advantage is that it allows the organization to consume technology while simultaneously receiving feedback from auditors during its adoption.

Extensibility

Those organizations that desire to operate at the cutting edge of technology or have needs for extremely formalized procedures may desire to implement customizations on top of the IBM BDD for Compliance solution.

Due to the modularity of the IBM solution custom tailoring can be carried out by either IBM or appropriate IBM business partners.

For more information about Eclipse and IBM Rational tools visit:

http://www.ibm.com/software/rational/eclipse/

4

Roadmap to compliance management

In this chapter we present a solution for compliance management that demonstrates supports for all three dimensions of business-driven compliance for software development.

A solution using Rational tools

By embracing a business-driven development approach to compliance, your organization can replace *ad hoc* or unstructured processes with a streamlined and self-documenting software development life cycle. Your developers can share software assets and project data unconstrained by physical boundaries or burdensome manual documentation requirements, and you can gain the objective data and insight to make better management decisions—all of which can help your team deliver business results.

A business-driven development approach to compliance management can help your organization to:

- Reduce the cost of complying with regulations, standards, and policies.
- Provision policies, and to understand their impact on the IT environment.
- Establish an audit-ready and tamper-resistant software development environment for controlling changes and documenting adherence to internal control structures.
- Manage compliance projects by actively prioritizing, monitoring, and measuring planned activities against results.
- Establish the foundation for effective IT governance and business transformation.

The IBM solution-based usage model leveraging Rational tooling to solve the complex and relevant problem of complying with regulations, standards, and policies will achieve the above objectives through:

- Designing custom processes integrated with business industry standards
- Automating those custom processes and validating their design
- Generating project plan templates from the custom process
- Ensuring practitioner conformance through tool-directed behavior (TDB)
- Ensuring appropriate control points via electronic signatures
- End-to-end traceability of nearly all development artifacts
- Validation of implemented application business controls
- Integration of IT operations with development and automate deployment
- Monitoring of project audit-ability and generating metrics

The key capabilities that make this possible through the use of Rational tools are:

- *Life cycle requirements traceability* to help auditors verify that compliance requirements were accurately captured and implemented in key applications
- *Auditable workflow management* capabilities that help ensure and document that all software changes were made by authorized personnel for valid business reasons
- *Flexible metrics and reporting, electronic signature, and audit trail* capabilities that can be tailored to the exact processes and IT controls that govern your development environment
- *Verifiable software builds* to help ensure and document that software developed was actually deployed
- *Automated deployment* that is fully integrated into the development process
- *Continuous validation of compliance mandates* through integrated test management
- *Tool-directed behavior (TDB)* with appropriate product and process quality metrics management
- *A fully integrated process and product audit console solution* for the process and development artifacts

Compliance roadmap

The compliance solution presented here supports all three dimensions of business-driven compliance for software development:

- *Say what you do*—Demonstrate that all compliance mandates are accurately captured and implemented in key applications.
- *Do what you say*—Demonstrate that all software changes are made in a secure, audit-ready environment, subject to appropriate IT controls at relevant life-cycle checkpoints.
- *Be able to prove it*—Demonstrate the effective oversight over compliance remediation projects.

This solution has three main components:

- Roles
- Capability patterns
- Tools

We provide an overview of the main components of this framework next.

Roles

This section provides an overview of the roles within this framework. A role defines the behavior and responsibilities of an individual or a group of individuals working together as a team within the context of a software engineering organization:

- *Process engineer*—Responsible for documenting and defining an IT governance process in accordance with applicable process standards, frameworks, and regulations.
- *Portfolio manager*—Responsible for overall IT governance and how alignment of compliance initiatives with organization strategy and business priorities.
- *Project manager*—Plans, manages, and allocates resources; shapes priorities; coordinates interactions with stakeholder; and keeps the project team focused. The project manager also establishes a set of practices that ensures the integrity and quality of project work products.
- *Policy analyst*—Captures compliance requirements after cataloging and interpreting regulations, standards, and policies to identify internal policies that allow the business to be compliant. Creates traceability between the compliance requirements, the internal policies, and the appropriate regulations, standards, and policies.
- *Business analyst*—Details the compliance project application requirements, and establishes priorities and workflow around the *in scope* requirements.
- *Implementer*—Develops software components and performs developer testing for integration into larger subsystems, in accordance with the project's adopted standards.
- *Technical lead*—A senior designer/implementer for an application. This role has both technical and managerial responsibilities. This role guides and oversees development of application software components. The technical lead is responsible for approving the code changes and completing the change requests.
- *Integrator*—Leads the planning and execution of implementation element integration to produce builds.
- *Release manager*—Responsible for collating the released work products from multiple projects, to produce a release that can be delivered into the production environment. Owns the release management process and has the responsibility and authority for the overall process results. The release manager is responsible for the overall quality of the process. This role is also the main coordinator within this process and is the focal point regarding releases for both the customer and the IT organization.

- *Deployment manager*—Delivers the software release into the production environment. The deployment manager works alongside the release manager and deploys the release under his authority.
- *Tester*—Conducts tests and logs the outcomes of testing.
- *Test manager*—Leads the overall test effort. This includes quality and test advocacy, resource planning and management, and resolution of issues that impede the test effort.
- *Auditor*—Responsible for inspecting the business operation. A primary objective for an auditor is to determine whether the business complies with company policies. An outside auditor may be seeking to determine compliance with regulations, standards, and policies.

Capability patterns

Capabilities patterns express and communicate process knowledge for a key area of interest such as a discipline or a practice and can be directly used by practitioners to guide their work. Examples for capability pattern are:

- Use case-based requirements management
- Use case analysis
- Unit testing

Capability patterns cover the set of activities that is required to deliver an auditable change for the project.

There are eight capability patterns in our solution for compliance. The main objectives for these use cases are:

- *Process modeling and design for compliance*—Document a method for defining an IT governance process in accordance with applicable process standards, frameworks, and regulations.
- *Manage compliance portfolio*—Perform compliance/governance activities while ensuring alignment of compliance initiatives with the organizations' strategic business priorities.
- *Manage projects*—Use templates and embedded process controls and measures to ensure repeatable and auditable project and risk management processes for compliance projects.
- *Initiate compliance project*—Create details of the compliance project application requirements and establish priorities and workflow around the *in scope* requirements.
- *Deliver auditable change*—Establish a foundation for an audit-ready infrastructure.

- *Securely deploy a release*—Build applications and securely move the build from the development environment to test and finally to production while providing traceability from the production back into the development environment.
- *Validate automated business controls*—Activities include validating the business rules and approving or rejecting the release.
- *Manage and generate metrics and reports*—Define organizational metrics for management reporting and performance analysis.

Refer to Appendix A, “Unified Method Architecture” on page 163 for more information about capability patterns.

Rational tools

The IBM Rational tools that are used in the IBM BDD for Compliance solution were introduced in Chapter 3, “IBM Rational's key capabilities for compliance management”:

- “Rational Portfolio Manager” on page 71
- “Rational RequisitePro” on page 75
- “WebSphere Business Modeler” on page 78
- “Rational ClearQuest” on page 81
- “Rational ClearCase” on page 82
- “Rational BuildForge” on page 90
- “Tivoli Provisioning Manager” on page 90
- “Rational ClearQuest Test Manager” on page 94
- “Rational Functional Tester” on page 99
- “Rational Manual Tester” on page 100
- “Rational Performance Tester” on page 100
- “Rational Method Composer” on page 106
- “Rational Unified Process” on page 106
- “RUP Plug-in for Compliance Management” on page 107
- “Rational ProjectConsole” on page 110
- “Rational SoDA” on page 111

Solution details

Now that we have introduced the roles, capability patterns, and provided an overview of the Rational tools used as part of the compliance solution, we are ready to discuss the details of our compliance solution.

For each capability pattern we list the challenges that are addressed and the evidence that must be provided as part of the audit process. We also discuss the Rational tools that we use in our solution to achieve compliance.

Process modeling and design for compliance

This scenario demonstrates a method for the practice of documenting and defining an IT governance process in accordance with applicable process standards, frameworks, and regulations.

The scenario also examines how one might execute the process for defining an IT governance process integrated with the software development life cycle (SDLC). The scenario illustrates the capabilities of RequsitePro to support process analysis, and the use of Rational Method Composer (RMC) to support its documentation.

Finally, we discuss why it is important to have a direct linkage between methods, process, tooling, and workflow automation from a compliance perspective. Figure 4-1 shows the use case diagram.

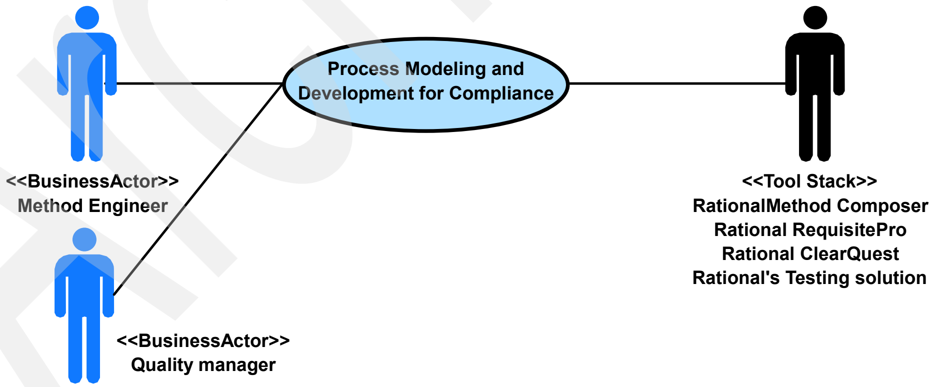


Figure 4-1 Process modeling and development for compliance

Objectives

The objective is to be able to prove compliance by providing the following evidence:

- Evidence of appropriate processes
  - Design and development methodology (for example, written and documented software methodology, or written controls)
  - Documentation and records management—Responsibility for generating, maintaining, and controlling documents
  - Training and education—Formal system to enable all personnel to perform assigned functions
  - Maintenance—Support organization and mechanisms to assist customers
- Evidence of current good practices (CGP) for project management

  Utilization of formalized planning methods for process design and work breakdown structure

Challenges

Some of the main challenges for governance process design and utilization are:

- Absence of a single process that meets the business compliance requirements
- Inability to ensure that practitioners conform to defined methods
- Inability to monitor process execution
- Absence of consolidated reporting on process metrics

One of the greatest challenges in most organizations is the disconnect between the software development organizations and other parts of the organization, such as IT operations.

The challenge from a compliance perspective is that neither auditors nor inspectors care about organizational boundaries, if such boundaries present a risk of malfeasance. Note that the audit is directly aimed at the software development life cycle, as we discussed in Chapter 2, “Compliance guidelines” on page 29. Because critical information and business controls are implemented and stored in the IT applications, nearly all audits turn into a defacto audit of IT applications and package development.

Thus, if an organization is to ensure appropriate protection for its stakeholders, a comprehensive shared process must be defined that crosses the organizational and governance boundaries.

Another significant challenge for most companies is the absence of any process specifically designed to meet their particular needs. Government regulations present no concrete actionable plan for the implementation of process. Furthermore, many pieces of legislation, such as Sarbanes-Oxley, cross multiple industrial domains or verticals. Consequently, differing practices and processes for business operation must be accounted for. This is the reason that most organizations seek out solutions designed to meet the specific, well-recognized, and accepted process frameworks and process standards for their specific industry. COBIT is an example of a well-recognized framework used for SOX, as we discussed in Chapter 1, “A discussion about compliance” on page 1.

In some cases the businesses have to establish several different standards and frameworks such as ITIL, COBIT, and IISO 900x to achieve the integration between SDLC and IT governance.

Discussion

To integrate a process that is designed to meet such requirements using IBM Rational tools, perform these activities:

- *Build and publish an integrated process design.*
  - Define the requirements for the processes in Rational RequisitePro.
  - Trace these processes to each other to determine overlap and satisfaction of the IT governance framework.
  - Perform gap analysis to identify measurements that are critical to determining where processes are lacking against the IT governance framework.
  - Trap your decisions and begin constructing an assembled process containing the components you want.
  - Build the real methodware inside the Rational Method Composer specifically designed for methodware documentation. You can then publish your methodware for public consumption in your business.
- *Automate the process via tool-directed behavior (TDB).*

  Implement documented workflow via automation, such that the defined process cannot be circumvented by practitioners, and defined artifacts can only be created via prescribed tooling.
- *Validate the automated process.*

  Validate the automated process using Rational’s CQTM testing solution, Rational Functional Tester, and Rational Performance Tester, so that you are able to prove to an auditor that the process designed is a legitimate process, designed to concrete specifications.

Activities

Table 4-1 shows the roles and tasks that are performed to achieve the objectives for process modeling and design for compliance.

Table 4-1 Roles and tasks in modeling and design for compliance

Roles	Activities
Process engineer	1. Analyze process requirements. 2. Integrate process and methods. 3. Document methodware and process. 4. Automate process or workflows.
Tester	5. Verify documented process. 6. Validate automated workflow.

Conclusion

Table 4-2 shows Rational tools that can help provide the evidence to prove compliance when documenting and defining an IT governance process, in accordance with applicable process standards, frameworks, and regulations.

Table 4-2 Conclusion: process modeling and design for compliance

Requirements	Rational capability
► Evidence of appropriate processes – Design and development methodology, for example, written and documented software methodology or written controls – Documentation and records management: assigned responsibility for generating, maintaining, and controlling documents – Training and education: formal training to enable all personnel to perform assigned functions – Maintenance: support organization and mechanisms to assist customers	► RUP - RMC configured IT governance w/RUP and ITIL/ITUP – Standards-based configurable methodware ensures documented processes can be found by all roles within the organization – Tool mentors with templates ensuring consistent tool usage among practitioner – Tool-directed behavior ensuring behavior is aligned with method and consistent among users and groups

Requirements	Rational capability
► Evidence of current good practices for project – Management: institution of formal planning methods	► RPM configured project – Exported work breakdown structure (WBS) from RMC ensuring linkages between project plans and roles, tasks and activities – Near real-time project tracking

Manage compliance portfolio

This scenario demonstrates how management performs the portfolio management activities of IT governance and demonstrates how to align compliance initiatives with organization strategy and business priorities.

Through graphical displays and pivot tables, executives will tightly align portfolios with business goals. The scenario demonstrates how, by consolidating multiple projects into a central repository, an organization can gain the visibility needed to plan and balance workload and perform enterprise risk management. Figure 4-2 shows the use case diagram.

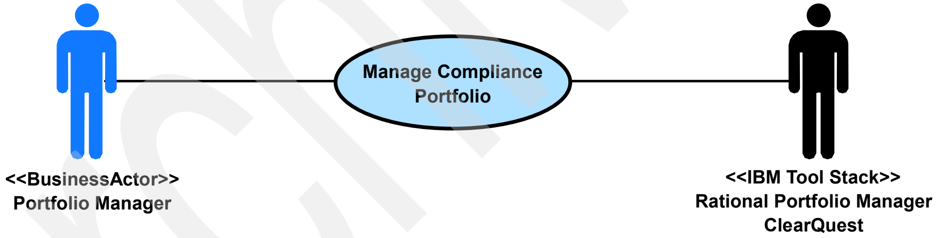


Figure 4-2 Manage compliance portfolio

Objectives

The objective is to be able to prove compliance by providing the following evidence:

- Evidence of appropriate risk management

  Implementation of a risk-based (risk reduction/removal) approach to compliance

- Evidence of project management

  Institution of formal planning methods

- Evidence of process adherence

Implementation of adequate project monitoring and status reporting

Challenges

Some of the main challenges for the portfolio management activities of IT governance are:

- Difficulty assessing application compliance impact on business

  This is the task of determining how much of a problem a given compliance issue will be to the business.

- Difficulty selecting systems and organizations for change

  How to develop a profile of *how well* a particular organization or system satisfies predefined compliance criteria.

- Inability to evaluate proposal and project return on investment (ROI) relative to compliance mandates

  How to measure the potential improvement that could be realized by executing a proposal or continuing to execute upon an existing project.

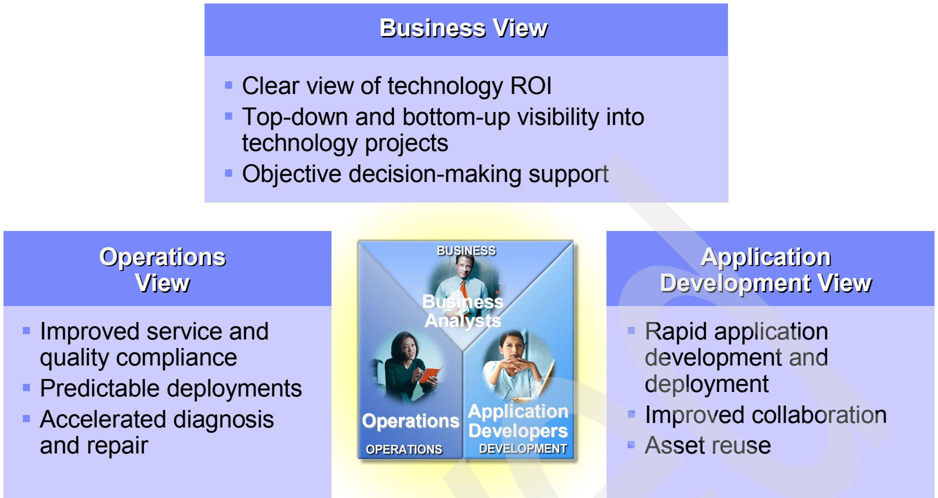
- Inability to counterbalance investments in controlled or uncontrolled efforts

  Every business has a limited amount of resources. Consequently, it can be difficult to determine which projects or proposals you must invest in. This requires a careful balancing of cost and business investment.

Discussion

Successfully adopting a business-driven development approach for compliance requires close alignment of development and IT projects with business priorities, as discussed in Chapter 1, “A discussion about compliance” on page 1, and Chapter 2, “Compliance guidelines” on page 29.

It also requires visibility for executives into the status of assets, whether those assets are projects or applications, so that resources and investments can be applied to the highest priority activities. IT needs the ability to manage with a seamless process from discovery to development to deployment and beyond (Figure 4-3).



Business View

- Clear view of technology ROI
- Top-down and bottom-up visibility into technology projects
- Objective decision-making support

Operations View

- Improved service and quality compliance
- Predictable deployments
- Accelerated diagnosis and repair

Application Development View

- Rapid application development and deployment
- Improved collaboration
- Asset reuse

Figure 4-3 A shared view of the development life cycle

The IBM Rational solution provides the comprehensive visibility that companies need to bridge the gap between business and IT, combining:

- Project portfolio management (PPM), which provides the ability to prioritize, plan, manage, and measure projects as a comprehensive portfolio
- Application life cycle management (ALM), which bridges the gap between development and post-deployment by providing both with a comprehensive set of development tools

This will provide the top-down visibility from PPM with the bottom-up development tool data and application monitoring capabilities of a complete ALM solution (Figure 4-4).

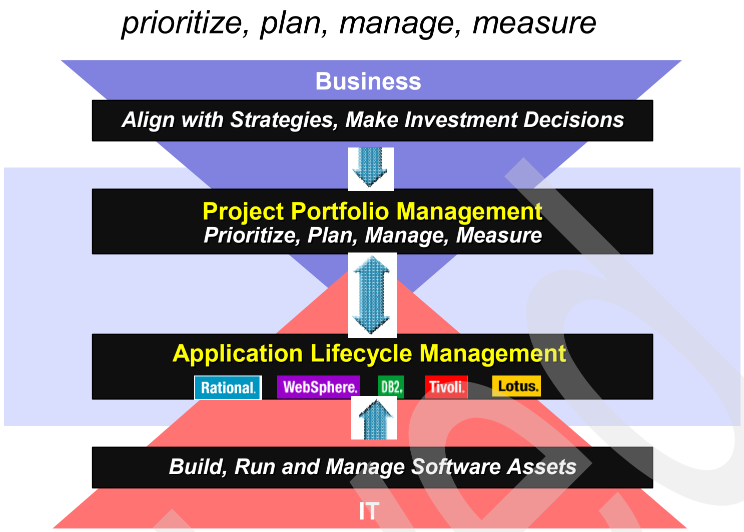

prioritize, plan, manage, measure
Business
Align with Strategies, Make Investment Decisions
Project Portfolio Management
Prioritize, Plan, Manage, Measure
Application Lifecycle Management
Rational.
WebSphere.
DB2.
Tivoli.
Lotus.
Build, Run and Manage Software Assets
IT

Figure 4-4 Connecting business, development, and operations

We observe at least seven process areas that all organizations seek to support with their project and portfolio management efforts:

- Manage *portfolios*—Achieve the business vision.

  Align priorities, projects, and resources with business priorities to meet organizational needs.
- Manage *scope*—Deliver business value.

  Seamlessly evolve ideas, proposals, initiatives, and requirements to measurable programs and projects.
- Manage *work*—Plan a balanced approach.

  Plan, cost, budget, resource, schedule, and report on programs, projects, and procurements.
- Manage *resources*—Optimize your staffing profile.

  Utilize staffing profiles for effective resource assignments and comprehensive capacity planning.
- Manage *financial*—Regulate your financial health.

  Manage charge of accounts with time keeping, expense, and capital expenditure tracking.
- Manage *exceptions*—React to changing needs.

  Capture, quantify, assign, collaborate and communicate, and track actions, issues, risk, constraints, opportunities, and changes in a workflow model.

- Manage *quality*—Track the expected results.

  Ensure that customer priorities get the attention that they deserve with the right process execution.

Some organizations may be more rigorous in some processes than others, and most organizations are challenged by a lack of consistency in how these processes are practiced across projects. They have either failed to codify their best practices for these processes or they have them in a document somewhere that few take the time to read and fully apply.

Rational supports project management process standards such as the Project Management Body of Knowledge (PMBOK) of the Project Management Institute (PMI), the Rational Unified Process, and the IBM Global Services World Wide Project Management Method. Rational Portfolio Manager helps customers facilitate and automate these best practices.

Utilizing scorecards in Rational Portfolio Manager, one can develop a profile of how well a particular organization or system satisfies predefined compliance criteria and be able to forecast how much potential improvement could be realized by executing a proposal or continuing to execute upon an existing project. By combining multiple scorecards for organizations, projects and proposals, and existing project/proposal investments, one can construct a view that would present the optimal combination of investment for compliance and return on investment (ROI).

Similarly, this solution space takes on the individual project efforts in addition to the enterprise portfolio view to minimize risk (Figure 4-5). The PMBOK core processes are identified by some analysts as your base set of functionality areas critical to a PPM tool. We can leverage this collection of processes as a well-known and proven framework for describing our functional capabilities.

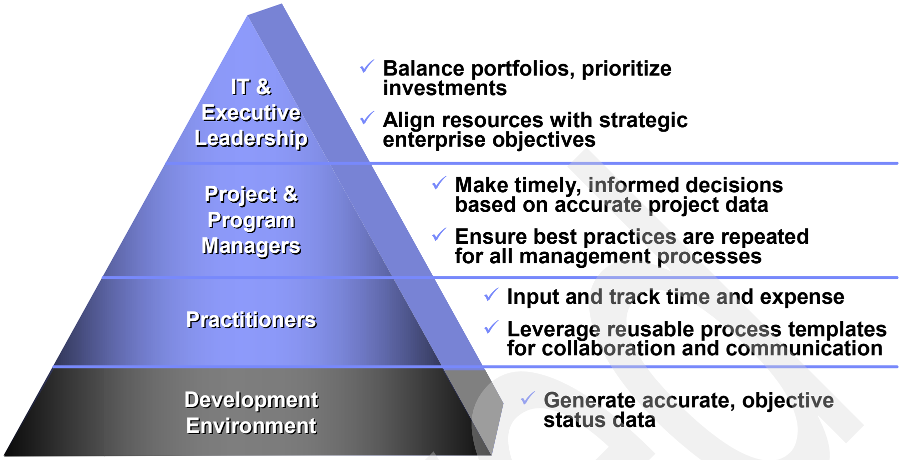

IT & Executive Leadership
Project & Program Managers
Practitioners
Development Environment
✓ Balance portfolios, prioritize investments
✓ Align resources with strategic enterprise objectives
✓ Make timely, informed decisions based on accurate project data
✓ Ensure best practices are repeated for all management processes
✓ Input and track time and expense
✓ Leverage reusable process templates for collaboration and communication
✓ Generate accurate, objective status data

Figure 4-5 RPM aligns priorities, projects, and people

Managing investments and balancing cost across the entire portfolio of an organization's IT assets is an important aspect in IT governance. In recent years, this responsibility has expanded to include executive oversight of the multi-year effort required to implement regulatory changes across the organization, controlling costs, budget, and delivery.

As we discussed in Chapter 1, “A discussion about compliance” on page 1, auditors want to see how you manage oversight for your IT compliance projects. Rational Portfolio Manager (RPM) makes effective IT oversight a reality with the ability to prioritize, track, and manage compliance projects across the IT portfolio by:

- Managing tasks and resources
- Tracking effort expended and maintaining artifacts created
- Managing schedule and milestones against deadlines
- Minimizing overhead and automating development flows

When RPM is used with other IBM Rational tools, it provides organizations with a 360-degree view of their evolving compliance project status.

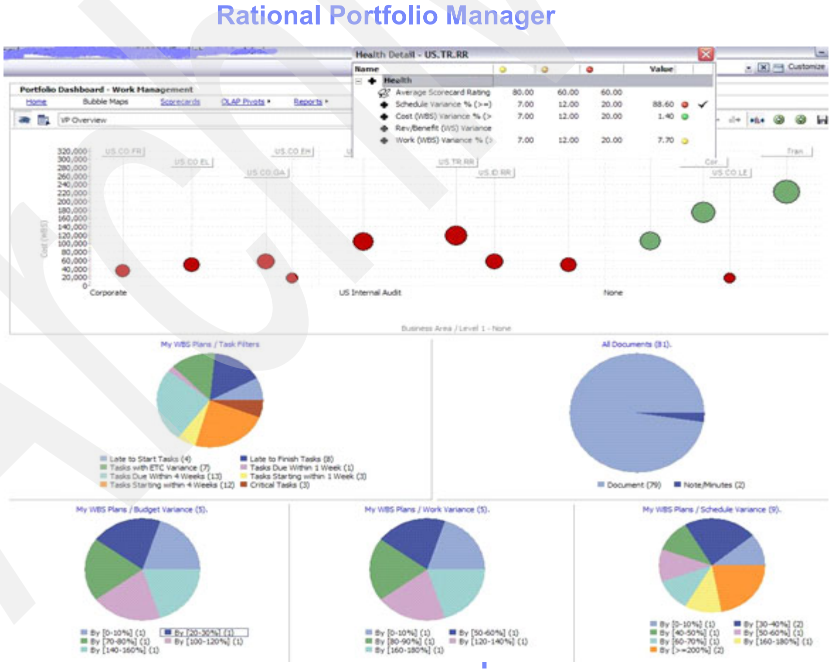
In addition, RPM brings executive, management, and team member processes into one integrated environment to achieve the business vision to:

- Deliver business value.
- Plan a balanced approach.
- Optimize the staffing profile.
- Regulate the financial health.
- React to changing needs.
- Track the expected results.

Finally, RPM can be used to measure the success and health of compliance-related projects:

- Executive oversight of compliance initiatives across divisions, departments, geographies.
- Identify IT projects based on compliance risk.
- Review and compare proposals/projects for ROI.

Figure 4-6 shows an investment map (bubble chart) in RPM that is a graphical depiction of the project status.



Health Detail - US.TR.RR

Name	●	●	●	Value		
Health						
Average Scorecard Rating	80.00	60.00	60.00			
Schedule Variance % (>=)	7.00	12.00	20.00	88.60	●	✓
Cost (WBS) Variance % (>	7.00	12.00	20.00	1.40	●	
Rev/Benefit (WS) Variance						
Work (WBS) Variance % (>	7.00	12.00	20.00	7.70	●	

Figure 4-6 Measure the success and health of the project

Figure 4-6 shows several dimensions including size in dollars, size in relation to other projects, and current health (red, yellow, or green). The settings for the investment maps have many options, and multiple maps can be built. The lower portion of the screen is pie charts showing budget, work, and schedule variances. In the live system, clicking any of these charts will take you directly to the details of the underlying item.

Activities

Table 4-3 shows the tasks that are performed to achieve the objectives for process modeling and design for compliance.

Table 4-3 Roles and tasks in portfolio for compliance

Roles	Activities
Portfolio manager	1. Acknowledge business issue. Precondition: “Raising a business issue” by the auditor 2. Analyze stakeholder request. 3. Develop and rank proposals. 4. Convert proposals to projects. 5. Close business issues. Precondition: “Deployment to production” by the deployment manager

Conclusion

Table 4-4 shows that Rational tools can help provide the evidence to prove compliance when aligning compliance initiatives with organization strategy and business priorities.

Table 4-4 Conclusion: managing compliance portfolio

Requirements	Rational capability
► Evidence of appropriate risk management Implementation of a risk reduction approach to compliance	► RPM configured – Scorecards for organizational units – Scorecards for projects and proposals – Portfolio analytics for compliance
► Evidence of project management Institution of formal planning methods	► RPM-configured project Exported work breakdown structure (WBS) from RMC ensuring linkages between project plans and roles, tasks and activities

Requirements	Rational capability
► Evidence of process adherence Implementation of adequate project monitoring and status	► RPM configured project Near real-time project tracking, leveraging timecards linked to method-based tasks

Manage projects

This scenario describes how a project manager uses templates, embedded process controls, and measures to ensure repeatable and auditable projects and management processes for compliance projects. Figure 4-7 shows the use case diagram.

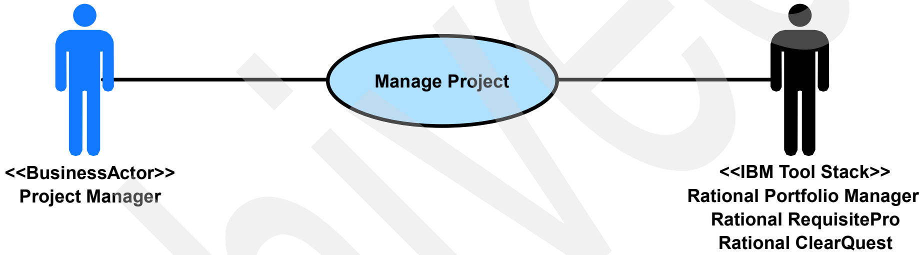


Figure 4-7 Manage project

Objectives

The objective is to be able to prove compliance by providing the following evidence:

- Evidence of management of development process for a project

  Collecting, validating, triaging, and prioritizing change request including enhancement requests and defects

- Evidence of project management

  Institution of formal planning methods

- Evidence of plan adherence

  Implementation of adequate project monitoring and status reporting

Challenges

Some of the main challenges for managing compliance projects are in the area of:

- Managing a controlled change process
- Enforcing approvals and controls
- Traceability across the life cycle of the original change requests, to the compliance requirements imposed on the application, to the implementation and associated testing
- Capturing compliance metrics and controls
- Enforcement of policies through phase gates

Discussion

The basic workflow for this use case is discussed next.

Establish project management structure

- Scheduling and costing guidelines
- Reporting and tracking metrics
- Estimating and forecasting guidelines
- Deliverable report formats and templates
- Change control procedures
- Stage gate reviews criteria

Note that the use of predefined compliance project templates ensures a consistent approach for different compliance projects.

Initiate projects

- Identify and assign a project team (affect at project level).
- Identify and implement all mandatory reporting and tracking requirements.
- Set up all required links and projects in other Rational tools:
  - Rational Portfolio Manager
  - Rational ClearQuest
  - Rational RequisitePro

Plan and release the projects

- Review and adapt the project work breakdown structure (WBS) to support specific requirements of the compliance project.
  - Define project phase, summary tasks, and tasks and milestones.
  - Validate all deviations from predefined templates and mandatory reporting and tracking requirements.
- Assign required resources against project tasks.

- Calculate and level project schedule.
- Validate project duration and costs against expected budgets and targets.
- Perform planning stage gate approval workflow.
- Baseline project schedule, resources, and costs.
- Copy proposed plan and commit resources.

Execute and control projects

- Define required monitoring and tracking metrics.
  - Define variance thresholds.
  - Set up automated alerts and triggers.
- Implement predefined views, settings, and layouts.
  - Resource utilization pivot
  - Scope management pivot
  - Standard project review report
- Conduct periodic project reviews.
- Review and validate risk list and mitigation plans.

  Produce required reports and analysis.
- Archive project reports and analysis, as required.

Close out projects

- Archive all work products of the compliance project:
  - Project deliverables
  - Status reports and artifacts
- Conduct lessons learned:
  - Review of WBS and deliverable templates
  - Review of resource consumption profiles
  - Review of mandatory metrics and reporting requirements
- Embed enhancements and recommendations in compliance project templates.
- Validate template modifications against compliance standards and policies.
- Release updated compliance project templates.

Activities

Table 4-5 shows the tasks that are performed to manage projects.

Table 4-5 Roles and tasks for managing projects

Roles	Activities
Project manager	1. Initiate a project. 2. Plan a project. 3. Execute and control a project. 4. Close a project.

Conclusion

Table 4-6 shows Rational tools that can help provide the evidence for repeatable and auditable project and management processes for compliance projects.

Table 4-6 Conclusion: Manage projects

Requirements	Rational capability
► Evidence of management of development process for a project. Collecting, validating, triaging, and prioritizing change request including enhancement requests and defect.	► ClearQuest configure project is used for change management including: – Collecting change requests including enhancement requests and defects. – Validating, triaging, and prioritizing change requests. ► Establishing traceability between compliance requirements and ClearQuest request for enhancement (RFE). Use Rational RequisitePro and ClearQuest integration.
► Evidence of project management Institution of formal planning methods	► RPM configured project Exported work breakdown structure (WBS) from RMC ensures linkages between project plans and roles, tasks and activities.

Requirements	Rational capability
► Evidence of plan adherence Implementation of adequate project monitoring and status reporting	► Using Rational Portfolio Manager – Monitor predefined compliance conditions using reports and automated triggers. – Authorize any template or workflow changes that affect the compliance model. – Enforce the policies controlling the phase gates by management in RPM.

Initiate compliance project

At the division level IBM Rational RequisitePro empowers software teams to track the definition, design, and implementation of these requirements from requirements elicitation through deployment.

We demonstrate how the business analyst's role details the compliance project application requirements, establishes priorities, and establishes a ClearQuest workflow around the in-scope requirements by using a release object. Figure 4-8 shows the use case diagram.

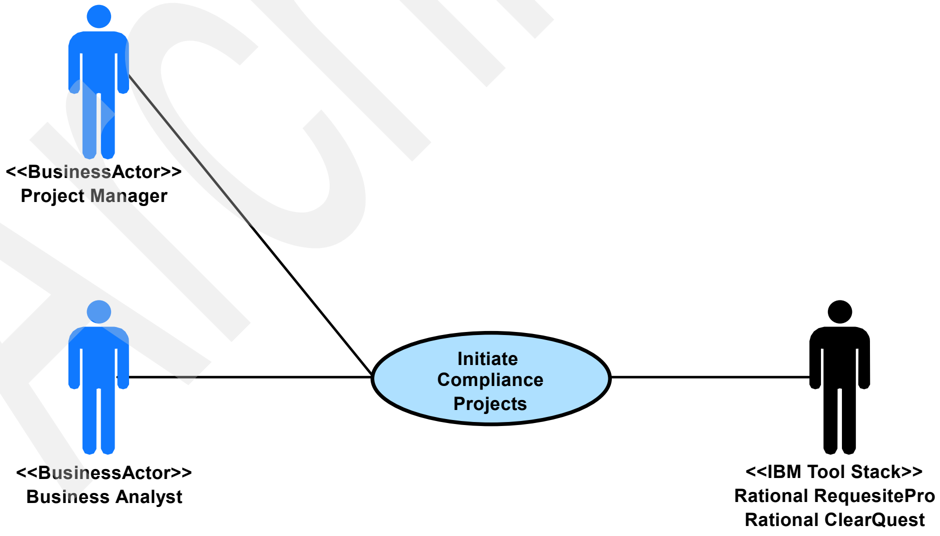


Figure 4-8 Initiate compliance projects

Objectives

The objective is to be able to prove compliance by providing the following evidence:

- Evidence of processes conformance
  - Implementation of workflow automation
  - Tool-directed behavior (TDB)
  - Artifact approval
  - Artifact quality gating
- Evidence of artifact linkage
  - Tool-directed artifact linkages
  - Chain of traceability on completed artifacts

Challenges

Some of the main challenges in this area are:

- Processes are documented, but nobody will adhere to them.
- Different organizations or departments make up their own procedures.
- Cannot locate the sign-offs for specific artifacts.
- Cannot locate even the latest versions of requirements, designs, and changes that correlate.

As companies seek to find more streamlined methods for operation, the reasons for some business controls can get lost over time. Although significant documentation exists in most organizations that outline standard procedures and sign-off of critical business controls, these are more often viewed as unnecessary bureaucracy rather than protections for the business and employees.

Discussion

Through the implementation of workflow automation that implements tool-directed behavior, you protect the business from lackluster controls or circumvention by practitioners.

TDB is a term used in regulatory domains described as the implementation of documented workflow via automation, such that the defined process cannot be circumvented, and defined artifacts can only be created through prescribed tooling.

Automated workflow can do several good things, besides ensuring that workflow is adhered to:

- TDB ensures that appropriate authorization points are not bypassed, and it can be used later to demonstrate to auditors that certain tasks were actually started or completed, as a result of what is documented.
- TDB and automated workflow can also ensure proper linkage of differing artifacts in the correct sequence. This becomes important downstream, when it can become necessary to reconstruct sequences of events in order to determine the what and when on a specific transaction
- Create artifact relationships that cannot be created any other way. For example, this is the case between the RequisitePro and ClearQuest integration. Unless the TDB enforced by the ClearQuest integration server is invoked, the data can never be applied to either RequisitePro or ClearQuest.

Activities

Table 4-7 shows the roles and tasks that are performed to achieve the objectives for process modeling and design for compliance.

Table 4-7 Roles and tasks in initiate compliance project

Roles	Activities
Project manager	1. Initiate development. Precondition: “Project approval” by portfolio manager 2. Schedule release. Precondition: “Create change request” by business analyst 3. Schedule change request
Business analyst	1. Analyze stakeholder requests. Precondition: “Initiate development” by project manager 2. Design automated controls. 3. Create change requests.

Conclusion

Table 4-8 shows Rational tools that can help provide the evidence to prove compliance when tracking the definition, design, and implementation of the compliance requirements from requirements elicitation through deployment.

Table 4-8 Conclusion: Initiate compliance project

Requirements	Rational capability
► Evidence of processes conformance – Implementation of workflow automation – Tool-directed behavior in workflow – Controlled artifact approval – Artifact quality gating – Tool directed artifact linkages – Chain of traceability on completed artifacts	► ClearQuest configured – Workflow designed to match method workflow. – Integration to requirements tooling ensures artifact linkage can only occur one way. – Workflow approval with role separation in ClearQuest. ► RequistePro and ClearQuest Integration Artifacts linkages created via tool integration
► Evidence of requirements satisfaction – Evidence that system is fit for its intended use – Documented linkages between business drivers and business controls	► RPM configured project – Exported work breakdown structure (WBS) from RMC ensures linkages between project plans and roles, tasks and activities. – Near real-time project tracking.

Deliver auditable change

This scenario demonstrates how IBM Rational enterprise change management tools provide a foundation for an audit-ready infrastructure. The main goal is to demonstrate how:

- ClearCase provides management and control of software assets through version control, audit trails and traceability, user authentication, baselines, and reporting.
- ClearQuest automates workflow management to enforce process, audit trails, and electronic signatures that allow organizations to document authorizations and sign-offs at key stages in the life cycle.

Figure 4-9 shows the use case diagram.

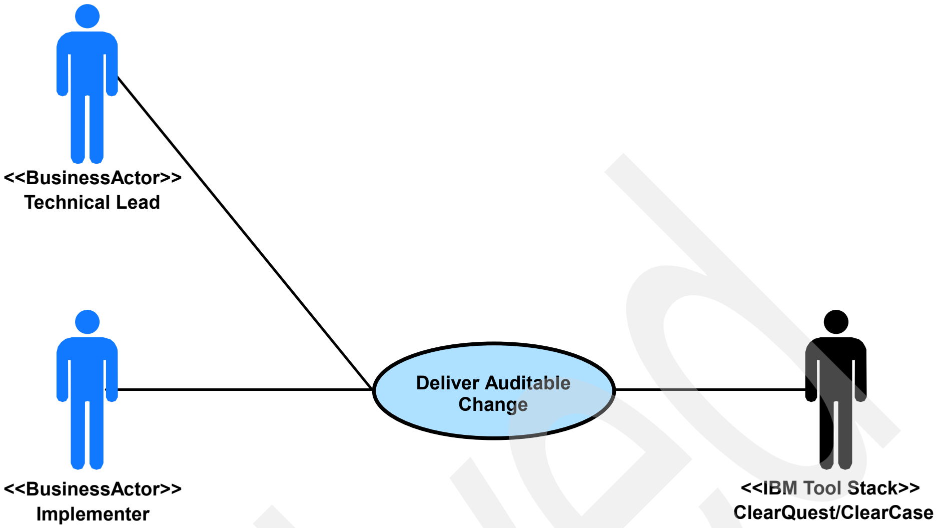


Figure 4-9 Deliver auditable change

Objectives

The objective is to be able to prove compliance by providing the following evidence:

- Evidence of appropriate configuration management and change control process
  - Documented methods for configuration, change, and version management
  - Implementation of appropriate role-based structures and separation of responsibilities
  - Management of releases, configuration items, source, and documentation
- Evidence of adherence to change management procedures
  - Process metrics, evidence of control points, such as sign-off or code walkthrough
  - Process metrics gathering
- Evidence of metrics for process monitoring and optimization

Challenges

Some of the main challenges in the area of configuration management and change control are:

- Inability to manage multiple concurrent releases, deployments, and platforms simultaneously with high quality
- Inconsistent change control procedures, a team using multiple processes and likely multiple tools
- Inability to change control file system structures
- Absence of integration between IT operations and development
- Inability to prove what code versions were deployed into production

In typical non-regulated environments you hear plenty of stories about developers crushing changes implemented in production, regressions in product, and the inadvertent releases of non-production quality code. In regulated environments this is unacceptable.

Discussion

Industrial strength configuration management procedures and processes must be implemented to protect the interests of all of the stakeholders of the business. These stakeholders include product or service consumers, senior executives or managers accountable for business controls, or shareholders of a publicly traded company.

At the heart of it all is the need to demonstrate due diligence in the design, development, testing, and handling of the application assets that implement automated business controls for the company. These controls are simply the bits of logic embedded in these applications.

Therefore, in order to demonstrate due diligence and control over these bits of logic, a company must show appropriate configuration management and change control over all of the assets used in the implementation of the automated systems.

Furthermore, it is not enough to simply implement a version control system. Version control simply implements a librarian function against code which can help in locating a problem after it has occurred. Appropriate integrated change management and configuration management processes are designed to reduce or eliminate the possibility that such catastrophes occur in the first place.

Figure 4-10 shows a strategy for instituting release management, deployment control, configuration management, and change request management including the three points of control, as discussed in “The three points of control strategy” on page 54.

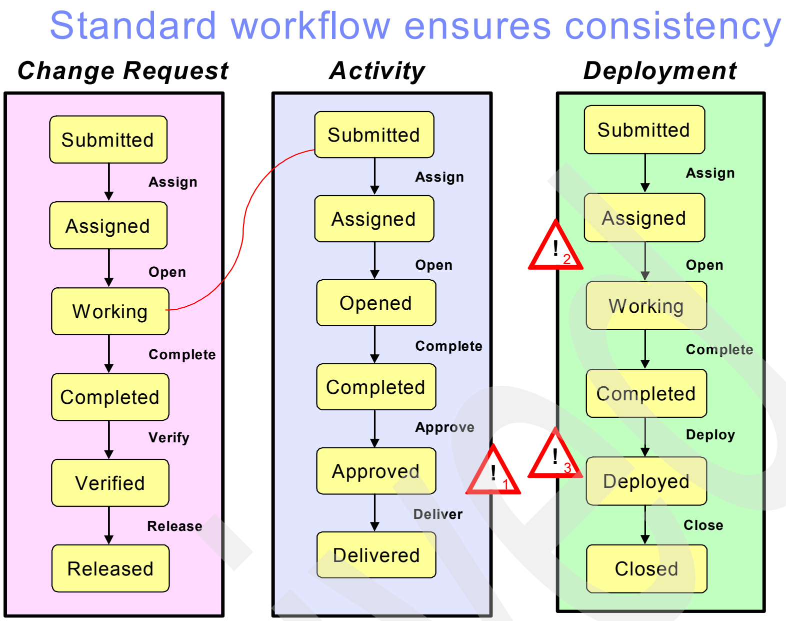


Figure 4-10 Three points of control

The three points of control are:

- *Release management*—Deals with the coordination of multiple concurrent enhancements, defect corrections, or other changes to a target application.
- *Deployment control*—Deals with the synchronized release of applications to the product environment itself, taking into account dissimilarities between the various platforms that a given application must operate on in a target technical environment.

  For example, a given financial clearing system used at the Mercantile Exchanges in New York and Chicago has client interfaces operating on Windows XP, MQSeries® middleware running on DEC Alphas and IBM z/Series, databases implemented under DB2®, and various extended client platform trading integrations via multiple Web servers. Each of these components of the application architecture requires a separate release, with coordinated deployment across these disparate platforms.
- *Change requests*—Deals with high-level requests to modify the system, which may span the application architecture. These requests must then be broken up into one or more change requests, or activities as part of the Unified Change Management (UCM) workflow.

This style of comprehensive control ensures auditors and inspectors that the business understands the potential risks presented to stakeholders by inadequate code and configuration control.

As we discussed previously, automated workflow can ensure proper linkage of differing artifacts in the correct sequence using Rational tools, as shown in Figure 4-11.

These are *must have* capabilities when reconstructing sequences of events, for example, to determine what transactions took place at specific times.

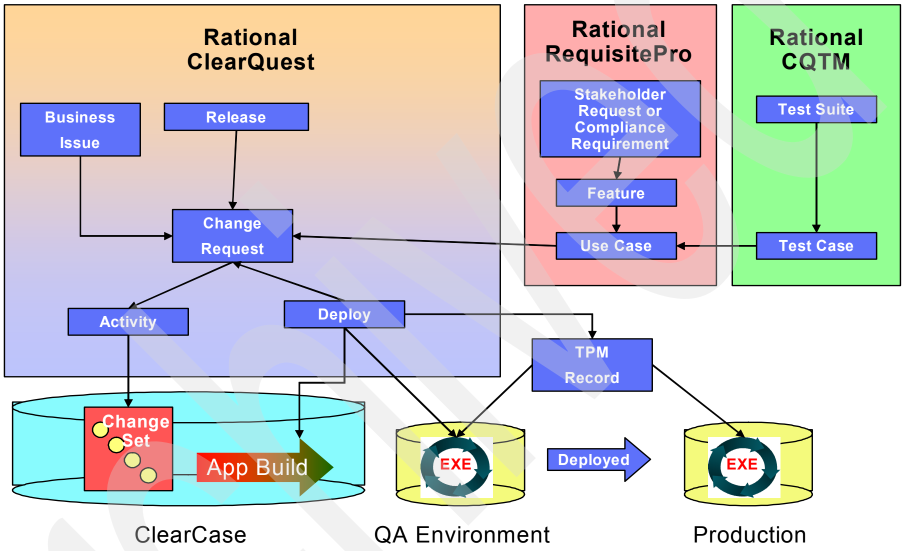


Figure 4-11 Traceability object diagram

Activities

Table 4-9 shows the roles and tasks that are performed to achieve the objectives to deliver auditable change for compliance.

Table 4-9 Roles and tasks from deliver auditable change for compliance

Roles	Activities
Implementer Technical lead	► Code, review, and approve automated control. Precondition: Schedule change request by project manager.

Conclusion

Table 4-10 shows Rational tools that can help provide the evidence necessary for compliance by providing a foundation for an audit-ready software development infrastructure.

Table 4-10 Conclusion: Deliver auditable change

Requirements	Rational capability
► Evidence of appropriate configuration management and change control process – Documented methods for configuration, change, and version management – Implementation of appropriate role-based structures, separation of responsibilities – Management of releases, configuration items, source, documentation, and so on ► Evidence of adherence to change management procedures – Process metrics, control points such as sign-off or code walkthrough – Process metrics gathering ► Evidence of metrics for process monitoring and optimization	► Rational Unified Process for ITG Appropriate documentation for all aspects of change are found in the exiting methodware and can be extended to include methods from other frameworks. ► ClearCase with unified change management – Automated project and component definition ensures project containment and precludes development stream pollution. – Workspace ownership enforces activity delivery controls. – Version control at the file system level ensures that all artifacts must be checked-out for any changes to occur. ► ClearQuest configured for compliance Process monitoring and metrics can be derived utilizing the ClearQuest reporting interface. External reporting is supported through rational software documentation automation tool, RPM, Rational project console, or via crystal reports.

Securely deploy a release

This scenario allows the IT operations team to build applications and securely move the build from the development environment to quality assurance and to the production environment while providing traceability from the production back into development. The primary goal is to discuss how to bridge the gap between software builds and deployments.

This use case also demonstrates how using ClearCase and ClearQuest change management software along with Tivoli Provisioning Manager for distribution software can help automate, streamline, and accelerate the software build/deploy process. We also show the use of a release record and e-signature capability. Figure 4-12 shows the use case diagram.

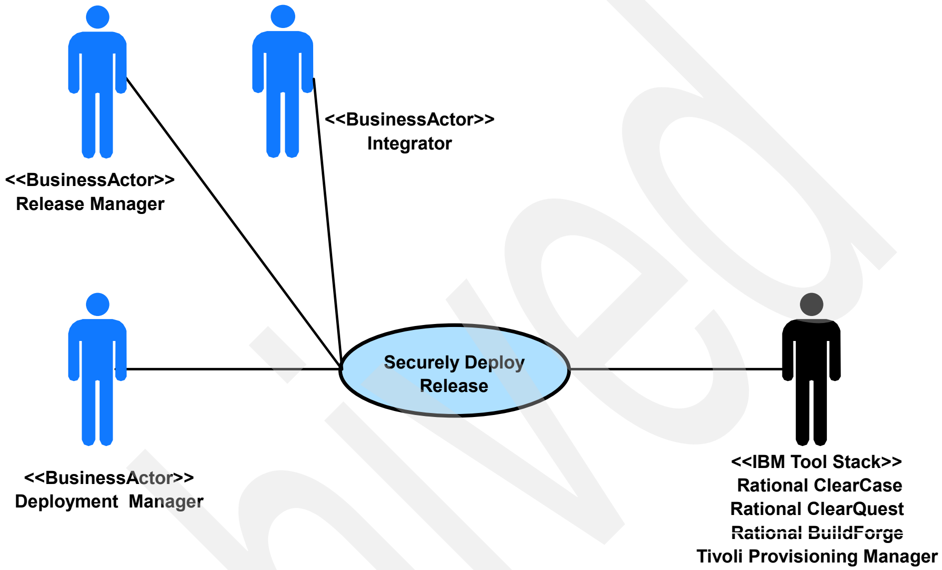


Figure 4-12 Securely deploy a release

Objectives

The objective is to be able to prove compliance by providing the following evidence:

- Evidence of secure deployment
  - Presence of consistent best practices for deployment
  - Consistent tracking of platform configurations, patch levels, dependent software, and layered products
  - Control of products deployed to secure regions including through multi-tiered firewalls
- Evidence of tamper-resistant production environment

Challenges

Some of the main challenges in the area of secure deployment of a release are:

- Absence of consistent best practices for deployment

- Inconsistent platform configurations, patch levels, dependent software, and layered products
- Difficulties with secure deployment of code beyond multi-tiered firewalls
- Deployment of servers from bare metal chassis
- Deployment to pervasive devices, such as palmOS, pocketPC, or mobile phone

Discussion

To overcome these challenges, we must capture best practices in automated deployment within the organization. Using Rational ClearQuest, ClearCase, BuildForge, and Tivoli Provisioning Manager together facilitates the build and deployment process and minimizes errors and inconsistencies that can arise during the course of your deployment.

You can implement a process for managing the deployment of your release through your test environments using a Rational ClearQuest deployment record. A Rational ClearQuest deployment record enables you to track the set of build artifacts that you want to deploy through a link to a Rational ClearCase deployment unit (DU). Each time you build your release, you create a new DU in Rational ClearCase that contains a list of which versions of the build artifacts will be deployed. The Rational ClearQuest deployment record (DR) uses the Rational ClearCase DU to track which build artifacts are deployed to which test environment at any point in the project life cycle.

BuildForge provides continuous monitoring of the IBM Rational ClearCase source repository, and executes builds automatically either when a change occurs or on a scheduled basis. The connection quickly and easily enables complete visibility and documentation of the code-build cycle. The adaptor gathers and reports source metrics for each software build. The adaptor interfaces directly with the IBM Rational ClearCase system through user-defined view specifications that are defined in the BuildForge user interface.

Once the IBM Rational ClearCase views are established, they can be linked to BuildForge projects. For each of the views defined, BuildForge scans the specified list of VOBs at scheduled intervals for changed files since the last successful build for a project, and collects the file differences and check-in comments. The change data is stored in the BuildForge repository and is associated with the executed build. A document view of each build session is captured and recorded in BuildForge, called a bill of materials (BOM), which includes the associated IBM Rational ClearCase information.

BuildForge can automatically detect IBM Rational ClearQuest activities that are part of a current build session by listing the activities that are associated with the designated view where the code resides. These activities are recorded during the build process and are part of the build record stored in BuildForge. Upon successful build exit status, BuildForge can automatically advance the state of the associated IBM Rational ClearQuest activities to a resolved status, and include a note from the build session. The BuildForge adaptor for IBM Rational ClearQuest also supports e-mail notification.

Rational ClearCase and ClearQuest enable you to track your software deployment through your specified test environments and to approve the deployment as needed, providing audit trails for your software assets that satisfy regulatory compliance requirements. You could then deploy your application to a production system by integrating Rational ClearCase and Rational ClearQuest with a software provisioning solution such as Tivoli Provisioning Manager. This integration automates your deployment process for your specific servers.

Tivoli Provisioning Manager is a resource management solution that provides core automated provisioning capability for corporate and Internet data centers. It coordinates and provisions assets based on your defined business processes. You can use Tivoli Provisioning Manager to provision, configure, and deploy a release into your production environment.

Tivoli Provisioning Manager introduces the concept of workflows. You can use workflows to automate and consistently repeat configuration and deployment tasks that you currently perform manually. Tivoli Provisioning Manager provides workflows that you can customize to support your existing tools and processes. You can also create new workflows. Workflows can automate processes from configuring and allocating servers, to installing, configuring, and patching software, and can be either large and complex or can consist of a single command.

The workflow framework enables you to manage your data center infrastructure. A data center infrastructure is made up of devices and integrated systems, called data center assets. Data center assets include both physical and logical assets:

- Servers that support managed applications
- Network devices such as switches, routers, load balancers, and firewalls that support communication between servers and applications
- Logical assets such as subnetworks and ports
- Service access points and access control lists that support secure data transmission
- Software products, software stacks, and software service releases
- Storage devices

The Rational ClearCase integration with Tivoli Provisioning Manager enables you to extract versioned file elements from Rational ClearCase and import them into the Tivoli Provisioning Manager deployment environment, ensuring that the correct versions of your files are deployed.

The Rational ClearQuest integration with Tivoli Provisioning Manager does the following:

- Enables Tivoli Provisioning Manager to check for deployment approval status in Rational ClearQuest, ensuring that your deployed release meets established quality standards
- Enables workflows to be represented as Rational ClearQuest records, facilitating deployment tracking and reporting
- Enables traceability between workflows and Rational ClearQuest deployment records, providing an audit trail that helps satisfy your organization's process and regulatory compliance requirements

Activities

Table 4-11 shows the roles and tasks that are performed to achieve the objectives for securely deploying a release for compliance.

Table 4-11 Roles and tasks for securely deploying release

Roles	Activities
Integrator Release manager Deployment manager	1. Build application. Precondition: "Code automated controls" by implementer/technical lead 2. Deploy to test environment (sign-off). 3. Complete release (sign-off). 4. Deploy to production (sign-off). Precondition: "Verify release" by Test manager

Conclusion

Table 4-12 shows Rational tools that can help provide the evidence necessary for compliance by providing a foundation to securely deploy a release.

Table 4-12 Conclusion: securely deploying a release

Requirements	Rational capability
► Evidence of secure deployment: – Documented best practices for deployment – Configuration accounting, irrespective of platform configurations, patch levels, dependent software and layered products – Control of products deployed to secure regions including through multi-tiered firewalls	► Rational ClearCase, ClearQuest, and BuildForge – Facilitates the build and deployment process and minimizes errors and inconsistencies – Provides traceability between build record and deployment record – Provides control points for approval before deploying to QA and production ► Tivoli Provisioning Manager – Automated deployment utilizing software package blocks for reusable deployment scripts. – Documented deployment logic via automated logic blocks. – Control over third-party deployment mechanisms, for example, Microsoft automatic deployment services and remote installation services – Enterprise directory support providing software distribution and inventory operations to be targeted by user
► Evidence of tamper-resistant production environment: Implementation of appropriate security measures on core production components	► Tivoli Provisioning Manager Monitors deployed resources and will automatically redeploy modified elements should they be changed in any form

Validate automated business controls

The focus of this scenario is to provide an overview of test management activities and benefits of Rational's test solutions, in particular, the benefits from RequisitePro, ClearQuest TestManager, Manual Tester, and ClearQuest with e-signature capability. Figure 4-13 shows the use case diagram.

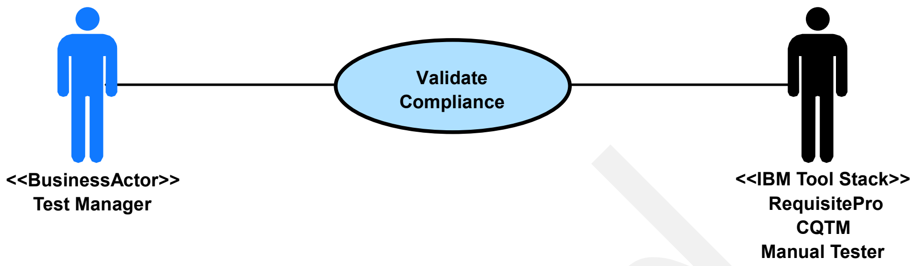


Figure 4-13 Validate compliance

Objectives

The objective is to be able to prove compliance by providing the following evidence:

- Evidence of adequate testing procedures

  Documentation of testing procedures

- Evidence that systems are fit for their intended use
  - Linkages between system requirements to test results
  - Examination of test results sign-off

Challenges

Some of the main challenges for the automated validation of business controls are:

- Inadequate ability to determine test motivation or to document requirements requiring tests
- Lower test quality because of rapid development of applications and aggressive schedules
- Inadequate management mechanisms for the support of multi-platform applications and configurations for each test

Discussion

A first important question to answer is “Why should you create and run a given test?” In other words, “What is the driving force for the test?“

The most common reason is because there is a system requirement that says “The system should do X”, therefore you should have a test for X.

It is essential to establish the following linkage when a feature or use case supports a compliance initiative:

- Establish a link between the test and the motivator.

  The motivator can be requirements in RequisitePro, for example. The objective is to be able to identify and to re-evaluate the test whose motivator has changed. This can be facilitated by having a link between the motivator and the test artifacts such as test plan, test case, and configured test case.

- Establishing a link enables requirements coverage reporting.

  Establishing the reason for the test case also enables you to do requirement coverage reporting. With the link established, you are able to see which of your requirements pass, which requirements fail, and which requirements are not tested. This type of information is invaluable to have when it comes to release time.

Rational CQTM queries can be used to identify tests requiring re-evaluation by determining if a new revision of requirements has been established. CQTM out-of-the-box query called *requirements coverage* can be used to show requirements and their related test artifacts for requirement coverage reporting.

The next question to answer is “Where should you run a test?” That is, on what workstation should the test execute and what are the specific attributes that specify the makeup of a workstation?

Configuration testing can be very difficult to manage. Testers often have hundreds of tests that must be executed on a variety of different workstations and, in some cases, the details of the test cases are different depending on the workstation where the test is executing.

An important strength of Rational’s testing solution is the ability to maintain the information regarding the location of test execution. Rational’s solution allows testers to define the configuration information for the workstations that they must test, including attributes such as operating system, browsers, disk space, and memory. They then select which configurations a test case, or group of test cases, must execute on. Next, a tester can specify, if needed, different test case variations for different types of workstations. Finally, ClearQuest TestManager (CQTM), Rational’s newest test management solution, acts as a gate keeper, ensuring that the correct test case is executed on the correct configuration.

Perhaps the more important question about a test case to answer is, “What test cases run first?” In other words, what are the guidelines for prioritizing test cases?

CQTM allows users to assign priority to tests. The higher the priority, the more important the test. CQTM also recognizes that while a test case requires some status such as pass, fail, or blocked for reporting purposes, it also recognizes that a test is rarely a 100% pass or fail. Results are not often so black and white. As a example, sometimes a test case will pass, but the tester may spot some small flaw along the way. To handle these types of situations, CQTM can be configured by adding additional fields to be used to indicate how much of a pass or fail a test really is.

Activities

Table 4-13 shows the roles and tasks that are performed to achieve the objectives for validating automated business controls for compliance.

Table 4-13 Roles and tasks for validating automated business controls

Roles	Activities
Test manager	1. Perform tests. Precondition: "complete release" by release manager 2. Verify change requests. 3. Verify release (sign-off).

Conclusion

Table 4-14 shows Rational tools that can help provide the evidence necessary for compliance by providing a foundation to validate automated business controls.

Table 4-14 Conclusion: Validate automated business control

Requirements	Rational Capability
► Evidence of adequate testing procedures Documentation of testing procedures ► Evidence that systems are fit for their intended use – Linkages between system requirements to test results – Examination of test results sign-off	► Rational software quality solutions – Tool mentors document best practices for use of Rational testing tools with rational methods. – Utilization of CQTM integrations to RequisitePro, ClearQuest ensure appropriate documentation of test motivation. – Support for both manual and automated tests with aggregation of test results in a single interface ensures complete coverage of business controls testing. – Use of Rational ClearQuest workflow can serve as the sign-off required to allow applications to move forward toward production.

Manage and generate metrics and reports

This scenario allows the project manager's role to define organizational metrics for management reporting and performance analysis. Here are some potential sample requests for real-time reporting:

- Number of enhancement requests (ER) by electronic signature state, for example, charts showing the number of enhancement requests per open, working on, closed state of the e-signature
- Number of test cases
- Number of defects
- Number of tested requirements
- Measurement of earned value
- Change requests initiated, accepted, and executed or rejected
- Number of exceptions and their status
- Number of controls and their status (verified, not verified)
- List of policies, applications, and verified compliance policy requirements
- Report on traceability chain
- Audit package

Figure 4-14 shows the use case diagram.

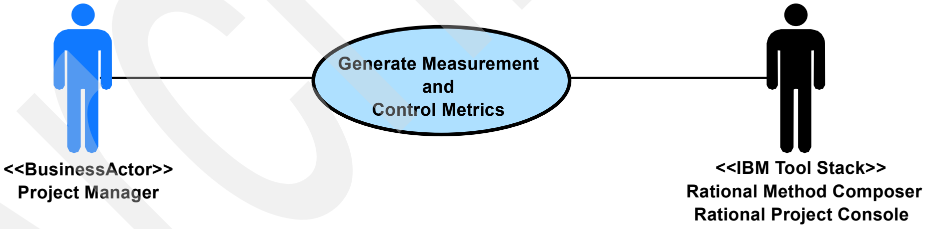


Figure 4-14 Generate measurement and control metrics

Objectives

The objectives are:

- Evidence of appropriate measurement plans

  Documented measurement plan, controls, and processes

- Evidence of quality metrics program

  Consistent data gathering and reporting program including distribution analysis, trend analysis, and aging analysis

Challenges

Some of the main challenges for the automated validation of business controls are:

- Absence of a single process that meets the compliance requirements
- Inability to ensure that practitioners conform to defined methods
- Inability to monitor process execution
- Absence of consolidated reporting on process metrics

Discussion

To demonstrate to an auditor that you have appropriate controls in place, you must do more than just show that you have measures of quality.

For example, configuration management and change control on both your business processes as well as the measurement process itself is instrumental to moving to the CMMI Level 5 standard. Also, demonstrating due diligence for process optimization is another important component of any compliance solution.

As we previously stated, to be in compliance with a regulation generally means that you can achieve both the performance and procedural requirements of a law:

- Performance requirements—Your ability to deliver functionality or tasks that support a specific regulation, standard, or policy
- Procedural requirements—Your ability to document adherence to an established operational process

It is not enough to only have plans and processes in place. A company must faithfully execute the plans and adhere to the processes, and put in place sufficient resources, time, training, and material to ensure their continual success. In other words, if a process is left unmonitored and unenforced, it is just a matter of time before it decays into anarchy.

The three main inputs to a software project are:

- Product requirements
- Project plan
- Software process

The measurements and reports for project management are therefore guided by:

- Identifying and tracking requirements
- Identifying the achievable objectives
- Tracking the status and progress of work performed to achieve the objectives

The same measurement data, as well as additional measurement of the process itself, provides the ability to control, fine-tune, and improve the software development process common framework for establishing and sustaining the measurement activities, as illustrated in Figure 4-15.

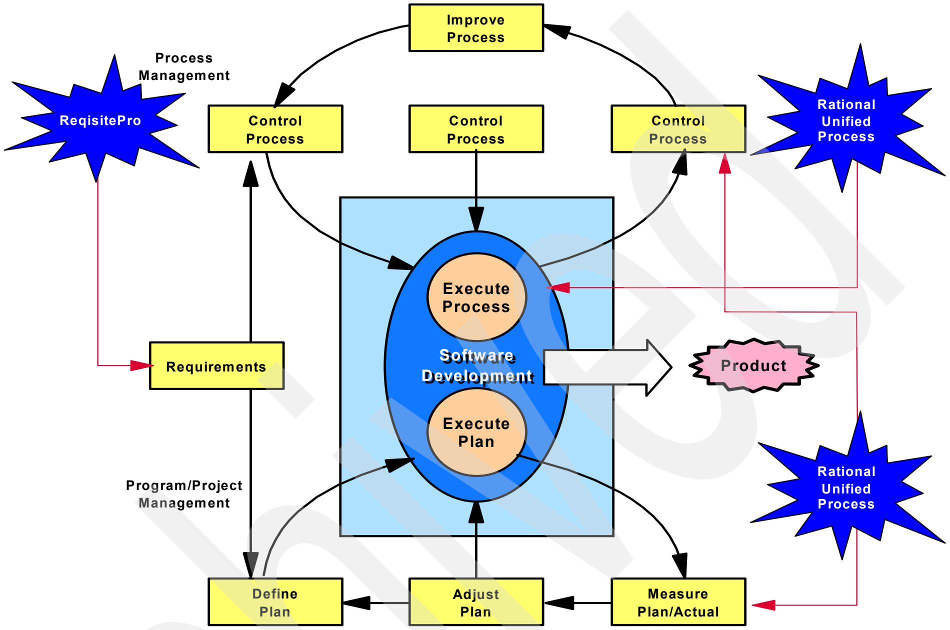


Figure 4-15 Common framework and process measurement

Table 4-15 shows some examples of the types of measurement to perform to monitor and improve the process. The *fitness* targets the requisites for a successful process execution, such as skills, experience, personnel, facilities, training, tools, and documented procedures. The *use* targets the execution of the process and the usage of the requisites, such as tools, and documented process.

Table 4-15 Fitness and use

Fitness	Use
► People skills – Experience – Training – Quantity	► People – Effort – Time
► Tools accessibility – Adequacy – Utility – Skills	► Tools – How widely – How frequently – How long

Fitness	Use
► Procedures coverage – Sufficiency – Quality – Documentation	► Procedures awareness – How widely – How frequently – How long
► Facilities space – Computers – Technical support – Sufficiency	► Facilities – How widely – How frequently – How long
► Work plan targets – Work breakdown – Structure – Applicability – Understandable	► Work plan awareness – How widely – How frequently – How long

Using Rational ProjectConsole and its integration with Rational tools or other third-party tools, a company can automate measurement of process execution against the defined plans. The result of these measurements can then be used to improve the plans and the processes incrementally (Figure 4-16).

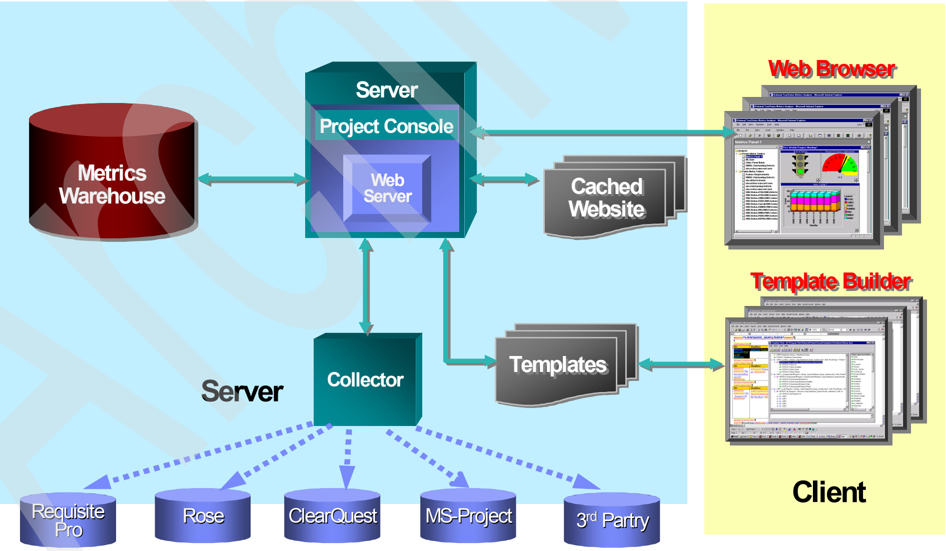


Figure 4-16 Typical configuration for PJC

Activities

Table 4-16 shows the roles and tasks that are performed to achieve the objectives for generating metrics and reports for compliance.

Table 4-16 Roles and tasks to manage and generate metrics and reports

Roles	Activities
Project manager	1. Define and implement the development processes and the associated control points. 2. Implement the types of queries for each policy and process instantiated in the workflow engines.

Conclusion

Table 4-17 shows Rational tools that can help provide the evidence necessary for compliance by providing a foundation to generate metrics and reports.

Table 4-17 Conclusion: Manage and generate metrics and reports

Requirements	Rational capability
► Evidence of appropriate measurement processes – Documented measurement plan – Documented key measures and data collection strategy – Documented strategy for data analysis ► Evidence of sustained commitment to the plan	► Rational Method Composer Implemented practical software and systems measurement (PSM) plug-in defines a program for appropriate measurement and documents roles, activities, and disciplines for execution. ► Rational ProjectConsole – Defined measures and control mechanisms document strategy for data collection. – PJC dashboard demonstrates metrics for consistent reporting of data analysis and summary. – Consistency of measures against plan, illustrates sustained commitment to optimizing processes for product quality. ► Rational SoDA Provide automated traceability, as well as automated artifact reporting.

Summary

In this chapter we described the benefits of Rational tools and their relevance to the regulatory compliance.

A

Unified Method Architecture

This appendix describes the Unified Method Architecture as implemented in the Rational Method Composer plug-in for compliance.

Unified Method Architecture defined

The Unified Method Architecture (UMA) is a process engineering meta-model that defines schema and terminology for representing methods consisting of method content and processes.

Fundamental principles of UMA

UMA is based on the following fundamental separations of concern:

- The separation of core method content versus the application of method content in processes
- The definition of an optional extensibility mechanism in the method for large-scale management of method and process repositories
- Packaging and configuration of method content, processes, and plug-ins in method libraries
- A separation of recommended method and guidance description fields
- A separation of semantic elements from their notation in process diagrams

The basic elements of UMA

The most fundamental principle of UMA is the separation of reusable core method content from its application in processes, and almost all of UMA's elements are categorized along these separations.

UMA separates reusable core method content from its application in processes:

- *Method content* describes what is to be produced, the necessary skills required, and the step-by-step explanation describing how specific development goals are achieved, independently of the placement of these items within a development life cycle.
- *Processes* take these method elements and relate them into semi-ordered sequences that are customized to specific types of projects. For example, a software development project that develops an application from scratch performs development tasks such as *Develop Vision* or *Use Case Design* similar to a project that extends an existing software system. However, the two projects will perform the tasks at different points in time with a different emphasis, that is, they will perform the steps of these tasks at different points in time and perhaps apply individual variations and additions.

Figure A-1 shows the difference between method content and process by representing them as two different dimensions:

- Method content describing how development work is being performed is categorized by disciplines. Each discipline comprises tasks (not visible in the figure) that provide step-by-step descriptions of how specific development goals are achieved.
- For a process, tasks have been referenced by the process from the method content and placed into breakdown structures and workflows ready for instantiation by allocating resources to perform the work and having real work products as the inputs and outputs of the tasks.

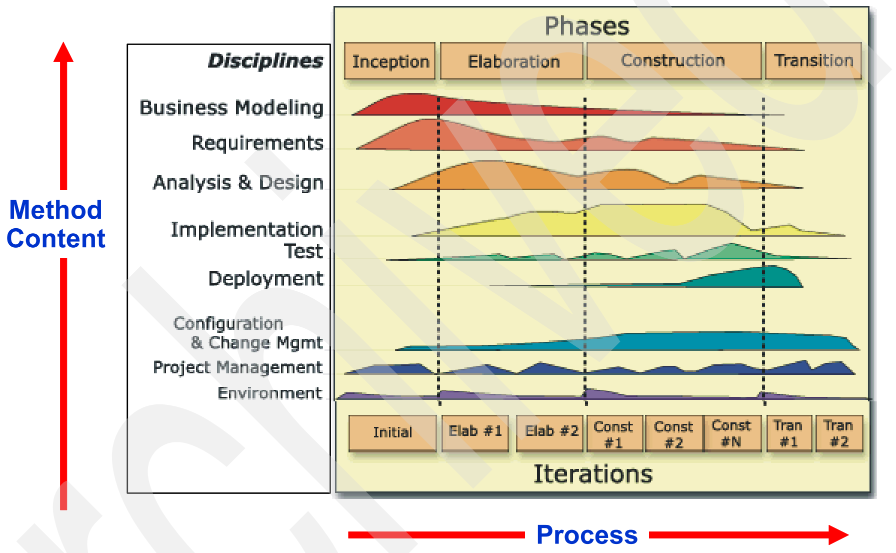


Figure A-1 Method content and process

UMA's key concepts reflect this separation of method content from process, as shown in Figure A-2. It shows that a method (also referred to as a method framework) contains method content, described with concepts such as work products, roles, tasks, and categories, as well as processes, described with activities, capability patterns, or delivery processes.

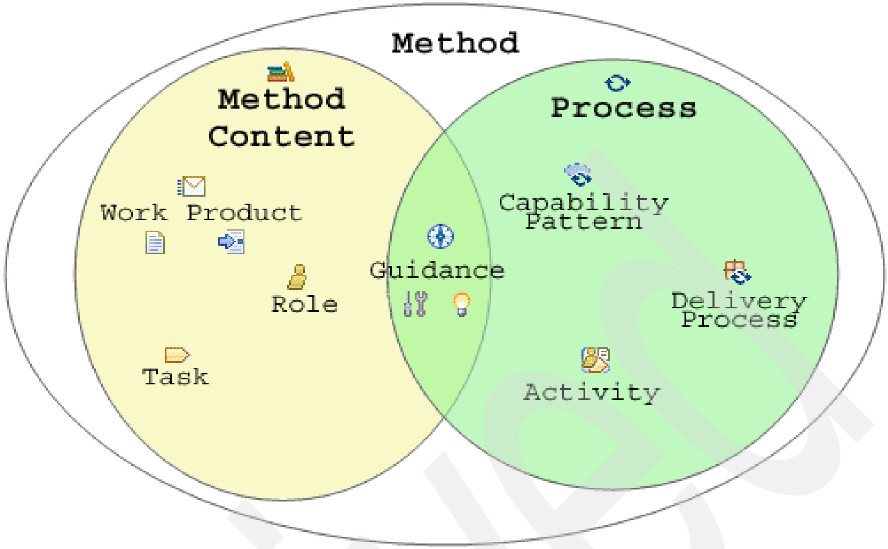


Figure A-2 Method framework

Key concepts of UMA

In this section we provide an overview of how the key UMA concepts are positioned based on whether they represent method content or process.

Method content

Method content is fundamentally described by defining tasks described with steps that have work products such as input and output and that are performed by roles. Roles also define important responsibility relationships to work products.

The key method content elements are:

- Work product
- Role
- Task

Development process

A development process defines the structured work definitions that have to be performed to develop a system, for example, by performing a project that follows the process. Such structured work definitions delineate the work to be performed along a timeline or life cycle and organize it in so-called breakdown structures or activity diagrams.

A process takes reusable core method elements, such as tasks and work products, and relates them into semi-ordered sequences that are then customized to specific types of projects.

Fundamental concepts used in UMA to define processes include:

- Key process elements
  - Activity
  - Capability patterns
  - Delivery process
- Guidance comes in many types:
  - Checklist
  - Concept
  - Example
  - Guideline
  - Practice
  - Report
  - Reusable asset
  - Roadmap
  - Supporting material
  - Template
  - Term definition
  - Tool mentor

The UMA meta-model has been developed as a unification of different method and process engineering languages, such as the Software Process Engineering Metamodel (SPEM) extension to the UML for software process engineering, and the languages used for RUP V2003, Unified Process, IBM Global Services Method, and IBM Rational Summit® Ascendant. As such it provides concepts and capabilities from all of these source models unifying them in a consistent way, but still allowing the expression of each of these source methods with their specific characteristics. This concept provides you with a general overview of UMA capabilities.

Separation of method content and process

UMA provides a clear separation of method content definitions from its application in processes. This is accomplished by separately defining:

- Reusable core method content, in the form of general content descriptions, such as roles, tasks, work products, and guidance
- Project-type specific applications of method content in context in the form of process descriptions that reference method content

Method content provides step-by-step explanations of how specific development goals are achieved independent of the placement of these steps within a development life cycle. Processes take these method content elements and organize them into a sequence that can be customized to specific types of projects. For example, a software development project that develops an application from scratch performs development steps similar to a project that extends an existing software system. However, the two projects will perform similar steps at different points in time with a different emphasis and perhaps individual variations.

Content reuse

UMA allows each process to reference common method content from a common method content pool. Because of these references, changes in the method contents will automatically be reflected in all processes using it. However, UMA still allows overwriting certain method-related content within a process as well as defining individual process-specific relationships for each process element (such as adding an additional input work product to a task, renaming a role, or removing steps that should not be performed from a task).

Process families

UMA's goal is to not only support the representation of one specific development process or the maintenance of several unrelated processes, but to provide process engineers with a tool set to consistently and effectively manage whole families of related processes.

UMA realizes this by defining the concepts of *capability patterns* and *delivery processes*, as well as specific reuse relationships between these types of processes. These concepts allow a process engineer to maintain consistent families of delivery processes that are project type specific and are variations of the same base method content and capability patterns. The result is different variants of specific processes built up by dynamically reusing the same method content and patterns, but applied with different levels of detail and scale; for example, process variants for small-scale versus large-scale development projects.

Multiple life cycles

A general method architecture must support different varieties and even combinations of life-cycle models for process definitions. These include waterfall, iterative, incremental, evolutionary, and so forth. The UMA meta-model is designed to accommodate multiple approaches. It provides a rich set of concepts and customization attributes for specifying temporal semantics for process elements such as phases, iterations, dependencies, and ongoing or event-driven work.

Flexible extensibility and plug-in mechanisms

UMA's method plug-ins provide a unique way of customizing method content and processes without directly modifying the original content. Instead, they just described the differences (additions referred to as contributions and replacements) relative to the original. This plug-in concept allows users to easily upgrade to newer versions of method content without losing their customizations.

Multiple process views

UMA defines multiple and consistently maintained views on processes. These views allow process engineers to approach process authoring based on their personal preferences. A process engineer can choose to define her processes with a focus on any of the following:

- Work breakdown—This is a work-centric view that defines tasks associated with a particular high-level activity.
- Work product usage—This is a results-based view that defines the state of certain deliverables and artifacts at various points in the process.
- Team allocation—This is a responsibility-based view that defines needed roles and their work product responsibilities.

UMA provides consistency between all these views, because they are all based on one integrated object structure. Changes in one view will immediately be reflected in the other views.

Reusable process patterns

UMA's capability patterns are reusable building blocks for creating new development processes. Selecting and applying a capability pattern can be done in one of two flexible ways:

- A pattern can be applied in a sophisticated copy and modify operation, which allows the process engineer to individually customize the pattern's content to his needs during the pattern application.
- A pattern can be applied via dynamic binding. This unique new way of reusing process knowledge allows commonly reoccurring activities to be factored out into patterns that can then be applied over and over again in a process. When the pattern is being revised or updated, all changes will automatically be reflected in all processes that applied that pattern.

Abbreviations and acronyms

ALM	Application life cycle management
BCS	Business Consulting Services
BDD	Business driven development
BOM	Bill of materials
CCB	Change control board
CEO	Chief Executive Officer
CFO	Chief Financial Officer
CFR	Code of Federal Regulations
CGP	Current good practices
CMM	Capability maturity model
CMMI	Capability maturity model integrated
CMS	Center for Medicare and Medicaid Services
COBIT	Control Objectives for Information and related Technology
COSO	Committee of Sponsoring Organizations of the Treadway Commission
COTS	Commercial off the shelf
CQ	ClearQuest
CQTM	ClaerQuest Test Manager
CRO	Chief Risk Officer
DEC	Digital Equipment Corporation
DFSS	Design for Six Sigma
DMADV	Define, measure, analyze, design, verify
DMAIC	Define, measure, analyze, improve, control
DPMO	Deviations per million opportunities
DR	Deployment record
DU	Deployment unit
EASHW	European Agency for Safety and Health at Work
ER	Enhancement request
FDA	Food and Drug Administration
FTE	Full time employee
GAAP	Generally Accepted Accounting Principles
GUI	Graphical user interface
HIPAA	Health Insurance Portability & Accountability Act
HTML	Hypertext Markup Language
IBM	International Business Machines
IDC	Subsidiary of International Data Group (IDG)
IDE	Integrated development environment
ISO	International Organization for Standardization
IT	Information technology
ITG	IT governance
ITIL	IT Infrastructure Library
ITLM	IT life cycle management
ITSO	International Technical Support Organization
LDAP	Leightweight Directory Access Protocol
M&A	Merger and acquisition
OGC	Office of Government Commerce in the UK
OLAP	Online analytical processing
OS	Operating system
OSHA	Occupational Safety & Health Administration

PJC	ProjectConsole
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PPM	Project portfolio management
PVOB	Project VOB
QA	Quality assurance
RFP	Request for proposal
RMC	Rational Method Composer
ROI	Return on investment
RPM	Rational Portfolio Manager
RUP	Rational Unified Process
SCLM	Software configuration library management
SCM	Software configuration management
SDLC	Software development life cycle
SEC	Securities and Exchange Commission
SEI	Software Engineering Institute
SLA	Service level agreement
SOX	Sarbanes-Oxley
SPEM	Software Process Engineering Metamodel
SPICE	Software Process Improvement and Capability DEtermination
TDB	Tool-directed behavior
TMTP	Tivoli Monitoring for Transaction Performance
TPTP	Tools Platform Project
TQM	Total quality management
TUP	Team Unifying Platform
UCM	Unified change management
UMA	Unified Method Architecture
UML	Unified Modeling Language
URL	Uniform Resource Locator
VOB	versioned object base
VOC	voice of the customer
WBS	work breakdown structure

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbook.

IBM Redbooks

For information about ordering these publications, see "How to get IBM Redbooks" on page 175. Note that some of the documents referenced here may be available in softcopy only.

- *Software Configuration Management: A Clear Case for IBM Rational ClearCase and ClearQuest UCM,* SG24-6399
- *Deploying Applications Using IBM Rational ClearCase and IBM Tivoli Provisioning Manager*, REDP-4105
- *Rational Application Developer V6 Programming Guide*, SG24-6449
- *Using a Single Business Pattern with the Rational Unified Process (RUP)*, REDP-3877
- *Patterns: Model-Driven Development Using IBM Rational Software Architect*, SG24-7105
- *Build a Business Process Solution Using Rational and WebSphere Tools*, SG24-6636

Online resources

These Web sites and URLs are also relevant as further information sources.

Rational

- Rational software

  http://www-306.ibm.com/software/rational/
  http://www-306.ibm.com/software/rational/offerings/lifecycle.html

- Rational products

  http://www-306.ibm.com/software/rational/sw-atoz/
  http://www-306.ibm.com/software/awdtools/clearcase/
  http://www-306.ibm.com/software/awdtools/clearquest/

http://www-306.ibm.com/software/awdtools/reqpro/
http://www-306.ibm.com/software/awdtools/portfolio/
http://www-306.ibm.com/software/awdtools/rmc/
http://www-306.ibm.com/software/awdtools/tester/functional/
http://www-306.ibm.com/software/awdtools/tester/manual/
http://www-306.ibm.com/software/awdtools/tester/performance/
http://www-306.ibm.com/software/awdtools/soda/
http://www-306.ibm.com/software/rational/buildmanagement/
http://www-306.ibm.com/software/awdtools/team/
http://www-306.ibm.com/software/awdtools/team/client/
http://www-306.ibm.com/software/awdtools/developer/application/
http://www-306.ibm.com/software/awdtools/studioapplicationmonitor/
http://www-306.ibm.com/software/rational/toolkit/ipo_toolkit.html

- Rational on developerWorks

  http://www-128.ibm.com/developerworks/rational/library/

- Rational services

  http://www.ibm.com/software/rational/services/professional/

Tivoli

- Tivoli Provisioning Manager and Monitoring for transaction Performance

  http://www-306.ibm.com/software/tivoli/products/prov-mgr/
  http://www-306.ibm.com/software/tivoli/products/monitor-transaction/

Compliance regulations

- Sarbanes-Oxley

  http://www.sec.gov/news/studies/principlesbasedstand.htm

- USA Patriot Act

  http://thomas.loc.gov/cgi-bin/query/z?c107:H.R.3162.ENR:

- Basel II

  http://www.bis.org/publ/bcbs118.htm

- CFR 11

  http://www.access.gpo.gov/cgi-bin/cfrassemble.cgi?title=200521
  http://www.access.gpo.gov/nara/cfr/waisidx_05/21cfr11_05.html

- HIPAA

  http://www.cms.hhs.gov/HIPAAGenInfo

- Gramm-Leach-Bliley Act

  http://www.ftc.gov/privacy/glbact/glbsub1.htm

- Committee of Sponsoring Organizations of the Treadway Commission (COSO)

  http://www.coso.org

- Control Objectives for Information and related Technology (COBIT)

  http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/TaggedPageDisplay.cfm&TPLID=55&ContentID=7981

- Information Technology Infrastructure Library (ITIL)

  http://www.itil.org/itil_e/index_e.html

- Software Process Improvement and Capability DEtermination (SPICE)

  http://www.isospice.typepad.com/isospice_spice_project

- ISO 9000 and ISO 14999

  http://www.iso.ch/iso/en/iso9000-14000/understand/inbrief.html

- Six Sigma

  http://www.motorola.com/motorolauniversity.jsp
  http://en.wikipedia.org/wiki/Six_sigma

- Capability Maturity Model Integrated (CMMI)

  http://www.sei.cmu.edu/cmmi/cmmi.html

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

activity record 55
ADApT
  methodology 7
application deployment 114
approval 43
  financial 43
  technical 43
artifact
  traceability 12
audit 1
  data 37
  trail 44, 81, 121
auditable change 123, 144
auditor 10, 123
authentication 45, 82
  biometric 45
authorization 45
automated
  deployment 121
  tests 97
  workflow 39, 46

B

backlog 39
Bank Secrecy Act 7
Basel accord 7
Basel II 7
be able to prove it 62
best practices 39–40, 114
bidirectional traceability 85
bio sciences 3
biometric authentication 45
build management 83, 85, 87, 114
BuildForge 90, 151
  Adaptors 90
  FullControl 90
  FullThrottle 90
  Prism 90
business
  analyst 122
  artifact 10
  compliance 30
  controls 1, 124, 154
  environment 25
  philosophy 18
  process 4
  regulated 2
  rules and policies 30
  transactions 30
  transformation 13, 120
Business Driven Development for Compliance solution 59

C

capabilities patterns 123
Capability Maturity Model Integrated
  see CMMI
capability patterns 167
Carnegie Mellon University 19, 101
CCB 12
Center for Medicare and Medicaid Services 8
ceremony 51
CFR 8, 44
chain of traceability 5
chains of authority 69
chains of responsibility 32, 69
change
  control board 12
  management 79
  request 55, 147
change process 138
ClearCase 82, 144, 150
ClearQuest 81, 103, 141, 144, 150
CMM 19
CMMI 14, 19, 101, 159
COBIT 15–16, 127
Code of Federal Regulations 8, 44
commercial methods 40
Committee of Sponsoring Organizations of the Treadway Commission 16
compliance
  architecture 13
  audit 30
  challenges 1, 22
  considerations 51

controls 38
cost 38
environment 1
guidelines 29
initiatives 22, 68
management 9, 29, 59
roadmap 119
meaning 3
metrics 138
opportunity 13
portfolio 123, 129
problem and solution spaces 64
process 35
project 123, 141
requirements 76
roadmap 121
sustainable 9
validation 101
compliant software development process 53
configuration management 83, 145
best practices 114
Control Objectives for Information and related Technology 15
see COBIT
control principles 65
automation 116
build and deployment management 91
compliance validation 107
documenting results 112
IT governance 73
requirements management 79
software change management 84
software validation 100
control strategies 50, 52
corporate
policies 36
policy 4
COSO 16
COTS 26
CQTM 94, 127, 156
critical data file 6
CTQ 19
cultural change 27

D
DB2 147
defect 18
deployment
automatic 121
control 147
management 85
manager 123
record 55
release 149
development
governance 66
process 167
DFSS 19
DMADV 19
DMAIC 19
do what you say 62
documentation 46
documented
handoff 12
process 11
test results 12
documenting
results 108

E
EASHW 2
Eclipse 82
Eclipse Test & Performance Tools Platform Project 94
electronic signature 43, 45, 81, 121
enhancement request 158
environment specification 114
e-signature 45, 150
European Agency for Safety and Health at Work
see EASHW
evidence
documented process 11
process compliance 12
process maturity 11
extensibility 40, 118

F
FDA 2, 44
financial
approvals 43
report 3
reporting 6
Food and Drug Administration 2
Functional Tester 99

G

GAAP 6
gap analysis 25
Generally Accepted Accounting Principles 6
governance 31
  components 33
  dashboard 70
  development 66
  measures 32
  policy 32
  roles 32
government
  agency 10
Government's Office of Government Commerce 17
governmental
  regulations 2
Gramm-Leach-Bliley 9

H

Health Insurance Portability & Accountability Act 8
HIPAA 8, 14

I

identity management 46
implementer 57, 122
infrastructure 50
inspector 10
integrator 57, 122
International Organization for Standardization
  see ISO
ISO 18
ISO 9000 14
ISO 900x 18, 127
IT governance 13, 66
  framework 127
  process 125
IT infrastructure 24
IT Infrastructure Library 15
ITIL 15, 17, 67

L

laundering 7
LDAP 46
legislation 24
life cycle 64
  business driven development 70
  management 82, 131
  requirements traceability 121
life sciences 3
Lightweight Directory Access Protocol
  see LDAP

M

M&A 26
mandatory regulation 2
manual process 38
Manual Tester 100
maturity 20
measurable attributes 48
measures 48
medical device 3
mergers and acquisitions 26
method content 165
methodology
  ADApT 7
metrics 47–48, 145
  characteristic 48
  flexible 121
  primitives 48
  selection 48
Microsoft Visual Basic .NET 100
Microsoft Visual Studio .NET 82, 99
money laundering 7
MQSeries 147

N

National Commission on Fraudulent Financial Reporting 16
non-certifiable regulation 3
non-directive regulation 2
non-functional requirements 92
non-repudiation 43, 45, 64

O

Object Management Group 71, 103
Occupational Safety & Health Administration
  see OSHA
Office of Thrift Supervision 7
OGC 17
OLAP 72
opportunity 13
organizational
  metrics 49
  policies 36

  - position 52
- OSHA 2
- outsourcing 26

P

- parallel development 83
- performance requirements 3, 9
- Performance Tester 100
- pharmaceutical 3
- Plug-in for Compliance Management 107
- points of control 46, 54
- policy
  - analyst 122
  - artifacts 4
  - corporate 4
  - impact 67
  - internal 68
  - management 1
  - statement 4
- portfolio
  - management 68, 131
    - best practices 114
  - manager 122
- procedural requirements 9
- process
  - compliance 12
  - conformance 50
  - design 42
  - engineer 122
  - exception 10
  - improvement 17
  - integrity 31
  - linkage 64
  - manual 38
  - measurement 160
  - modeling 123, 125
  - standards 15
- program
  - manager 17
- project
  - management 82, 137
  - manager 56, 122
  - metrics 49
  - status 110
- Project Management Body of Knowledge 133
- Project Management Institute 133
- ProjectConsole 110
- Provisioning Manager 90

Q

- quality
  - assurance 12, 55
  - audit 19
  - critical 19
  - management 18

R

- Rational
  - product offerings 60
  - tools 59
- Rational Application Developer 82, 99
- Rational BuildForge 90
- Rational ClearCase 82
- Rational ClearQuest 77, 81, 99, 138
- Rational ClearQuest Test Manager 94
- Rational Functional Tester 97, 99, 127
- Rational Manual Tester 97, 100
- Rational Method Composer 71, 103, 106, 125, 127, 163
- Rational Performance Tester 99–100, 127
- Rational Portfolio Manager 63, 71, 104, 133, 138
- Rational ProjectConsole 110, 161
- Rational RequisitePro 75, 95, 127, 138, 141
- Rational SoDA 111
- Rational Software Architect 82
- Rational Team Unifying Platform 110–111
- Rational Unified Process 106
  - see RUP
- recertification 46
- records
  - management 24
  - retention 47
- Redbooks Web site 175
  - Contact us xiv
- regulation 1
  - business controls 3
  - examples 5
  - IT impact 3
  - mandatory 2
  - non-certifiable 3
  - non-directive 2
  - support 82
- regulatory requirements 61
- release 88
  - deployment 124, 149
  - management 114, 147
  - manager 56–57, 122

    - record 54
- reports 6, 46
- requirements
    - best practices 114
    - compliance 76
    - management 73
    - non-functional 92
    - performance 3, 9
    - procedural 3, 9
    - taxonomy 77
- RequisitePro 75, 154
- resource
    - manager 72
    - utilization 139
- retention policy 47
- return on investment 19
- RFI 34
- RFP 34
- risk and control consciousness 16
- roadmap 121
- ROI 19
- roles 122
- RPM 71
- RUP 20, 49, 67, 73–74, 84, 106
    - framework 21
    - phases 21
    - Plug-in for Compliance Management 107
    - plug-ins 106
    - process framework 106

S

- Sarbanes-Oxley
    - see SOX
- say what you do 62
- Say what you do, do what you say, and be able to prove it 60
- SCLM Advanced Edition for z/OS 82
- SCM integration 82
- SDLC 11, 64, 125
- security 23, 51
- separation of duties 42, 80
- service
    - delivery 17
    - desk 17
    - level agreement 4, 99
- service-oriented architecture 104
- Six Sigma 18
- SOA 104
- SoDA 111
- software
    - artifacts 37
    - change management 79
    - change questions 53
    - development
        - assets 82
        - audit trails 44
        - compliance 30
        - governance 31, 35
        - process 29
        - process documentation 42
    - process assessment 17
    - project questions 53
    - supplier 17
    - validation 92
- software development life cycle 11, 64
- Software Development Platform 78
- Software Process Engineering Metamodel 167
- Software Process Improvement and Capability dE-termination 17
- SOX 6, 14, 127
- SPICE 17
- stakeholder 49
- standards 14
- status reporting 137
- sustainable compliance 9

T

- TDB 10, 39, 65, 120–121, 127, 142
- Team Unifying Platform 110
- technical
    - lead 57, 122
    - metrics 49
- technology approvals 44
- test
    - manager 123
    - plan 96
    - progress 98
- tester 57, 123
- three points of control strategy 54
    - roles and responsibilties 56
- Title 21 CFR 11 8
- Title 21 Part 11 2, 45
- Tivoli Configuration Manager 91
- Tivoli Monitoring for Transaction Performance 99
- Tivoli Provisioning Manager 90, 150
- TMTP 99

tollgate 19
tool directed behavior
  see TDB
tool stack 41
tools 59
TPTP 94
TQM 18
traceability 5, 37, 41, 64
  artifact 12
  bidirectional 85
trust 51
TUP 110

U
UCM 83, 163
UMA 71, 103, 164
  capability patterns 170
  concepts 166
  elements 164
  meta-model 167
  method plug-ins 169
UML 71
Unified Change Management
  see UCM
Unified Method Architecture
  see UMA
USA Patriot Act 7
usage model 115
user authentication 82

V
validation 102
version control 83
VOC 19

W
Websphere Business Modeler 78
WebSphere Integration Developer 78, 82
WebSphere MQ Workflow 78
WebSphere Process Server 78
WebSphere Studio Application Monitor 99
workflow 46, 50
  automation 41
  management 82, 121
Workplace for Business Controls and Reporting 63, 67
workspace management 83

IBM Redbooks

Rational Business Driven Development for Compliance

Rational Business Driven Development for Compliance

Say what you do, do what you say, and be able to prove it

Manage compliance using Rational tools and processes

Leverage compliance for business advantage

This IBM Redbook is intended to help you design your processes and solutions using Rational products to manage compliance. It shows how Rational solutions can be applied by an organization to realize a controlled, auditable software development environment to help alleviate specific challenges imposed by standards, policies, and regulations.

This book provides a usage model and product configuration guidance to help a tools administrator implement and configure some or all of the Rational tools to address compliance challenges. This book is intended help you and your partners understand the design and deployment of IBM Rational's Business Driven Development for Compliance solution.

This book focuses on problems addresses by Rational products. The emphasis is on auditable change processes, in particular using ClearQuest and ClearCase. We provide additional coverage of requirements for compliance using RequisitePro, and compliance attestation using ClearQuest Test Manager, IT governance using Rational Portfolio Manager, Rational Method Composer, and the Rational Unified Process (RUP).

This book presents a solution for compliance management that demonstrates support of all three dimensions of business driven development of software for compliance, say what you do, do what you say, and be able to prove it.

IBM®

Redbooks

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7244-00 ISBN 073849657X