



IBM Software Group

RUP for Systems Z and other Legacy Systems

Susan M Burk
Senior Managing Consultant
IBM

smburk@us.ibm.com

413-726-9361

Rational software



Agenda

- Objectives
- A Quick Introduction to RUP
- A Few More Terms
- The RUP Big Picture
- RUP and Legacy Development
- Why Consider RUP for Legacy Implementations?
- Where are the Primary Impacts?
- Questions to Consider
- Project Experiences
- Design and Coding Styles for Legacy Development
- Does it matter which style is chosen?
- Finding the Balance
- Rules of Thumb for RUP Adoption for Legacy Implementations
- For More Information...



Objectives

As a result of listening to this session, attendees will gain an awareness of

- What RUP is
- Why and when RUP should be considered for legacy development
- What criteria are used to determine how much of RUP is needed to support teams using legacy languages.



A Quick Introduction to RUP®

- The Rational Unified Process (RUP) is an underlying set of principles for successful software development.
 - ▶ Started with combined ideas from thought leaders in object-oriented analysis and design
- At its core, RUP is
 - ▶ **Architecture-Centric** – positioned for change, reduced maintenance, enables reuse
 - ▶ **Use Case Focused** – functionality is sliced to provide business value
 - ▶ **An Iterative Process** – deliver time-boxed slices, high-risk first to (pre)prod
- RUP is also a collection of content about roles, tasks, and workproducts which are part of the software development lifecycle.
 - ▶ Handy online reference guide of proven software development practices, which can be customized to take it account specialized practices
 - ▶ The Rational Method Composer ® (RMC) is used to publish RUP

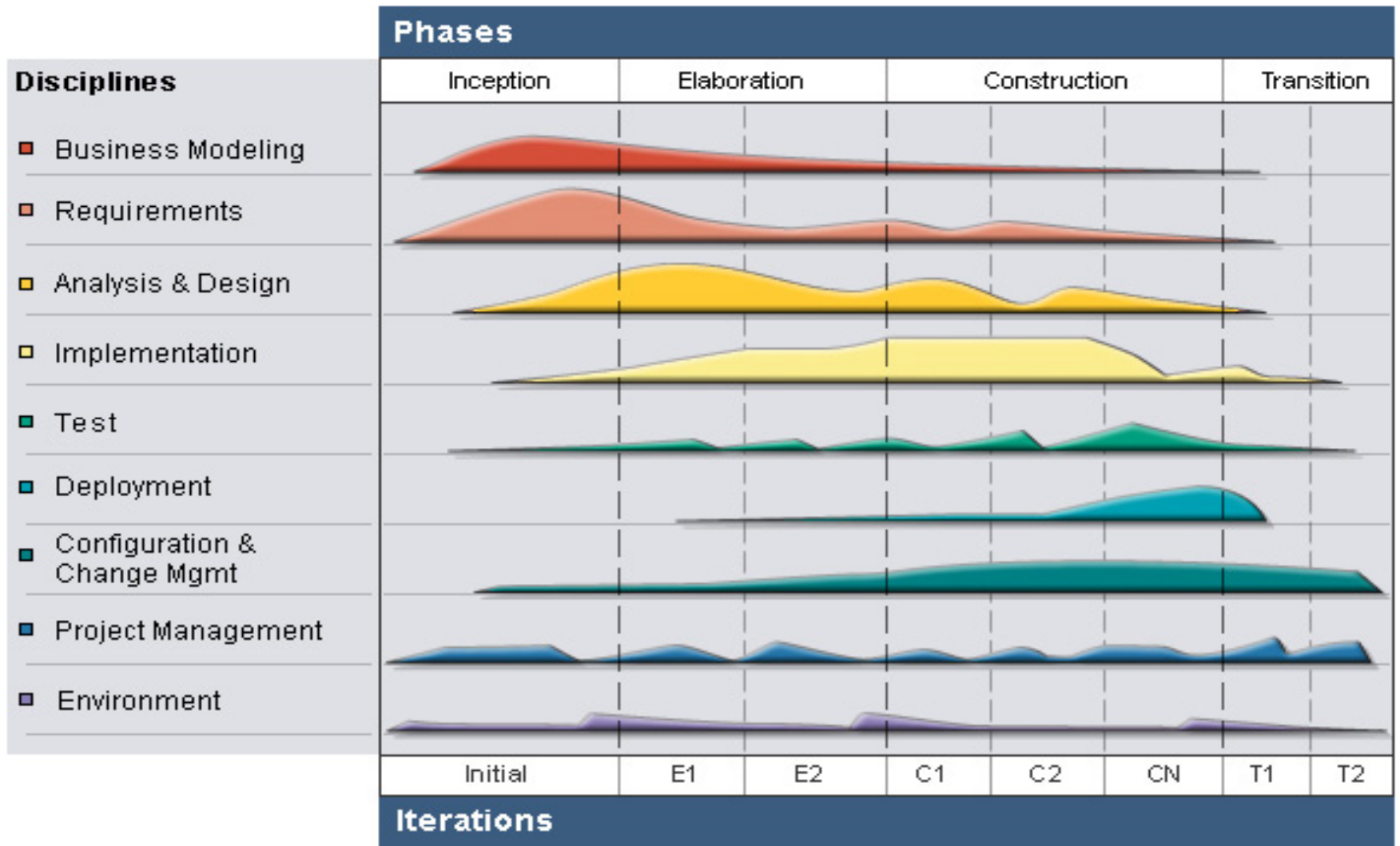


A Few More Terms

- Phase – a time-slice of a project characterized by common objectives.
- Iteration - a time-slice of a phase which has a distinct sequence of activities with a baselined plan and evaluation criteria resulting in an internal or external release.
- Milestone – A business decision point. The completion of a phase is marked by a milestone.
- Discipline - collection of tasks that are all related to a common perspective and skillset.
- Work Product – something which is produced, used, or modified by a task. Work products can be
 - ▶ Thought-aids
 - ▶ Artifacts - subject to version control
 - ▶ Deliverables - subject to version control and having value - material or otherwise - to a customer or other stakeholder
- Model – a work product which visually represents one or more concepts in a way which is meaningful within one or more disciplines.



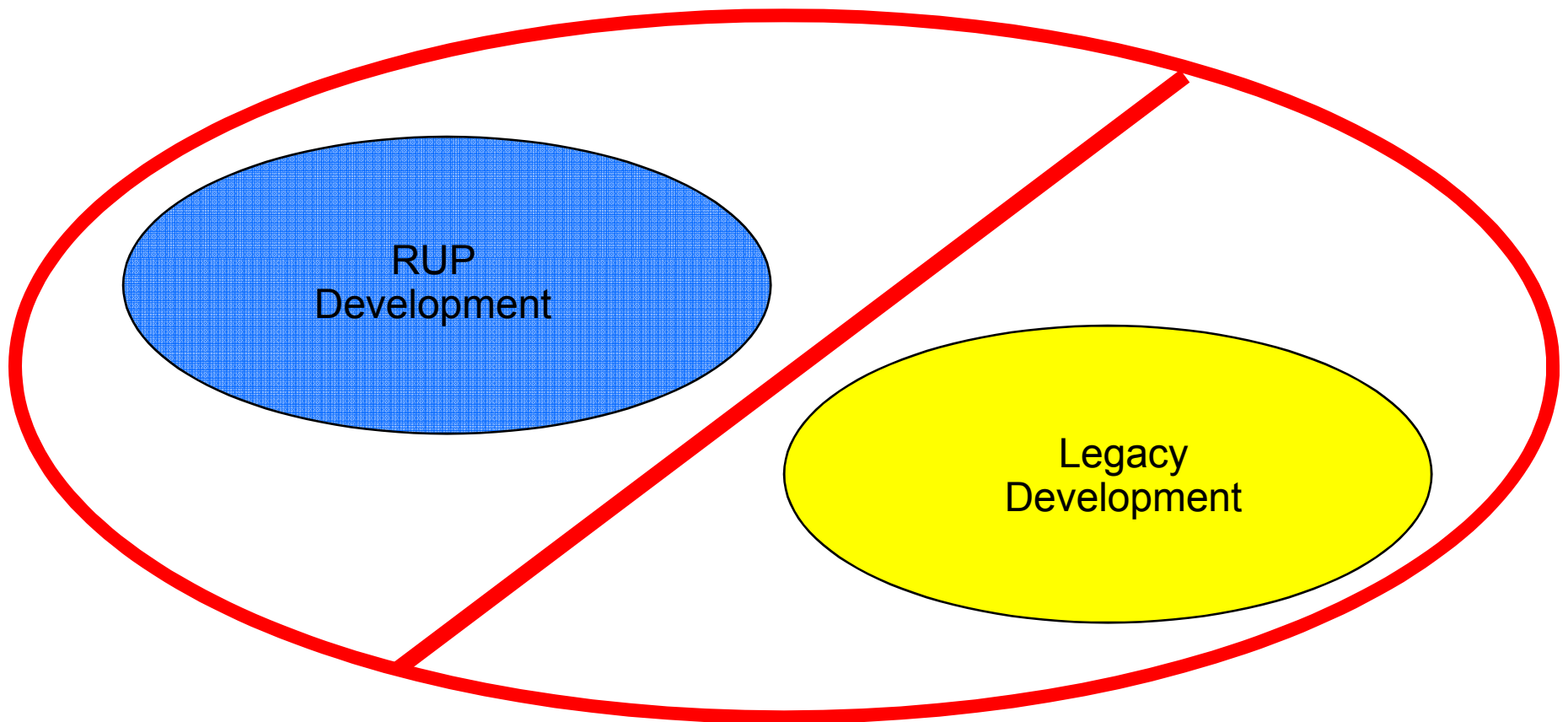
The RUP Big Picture



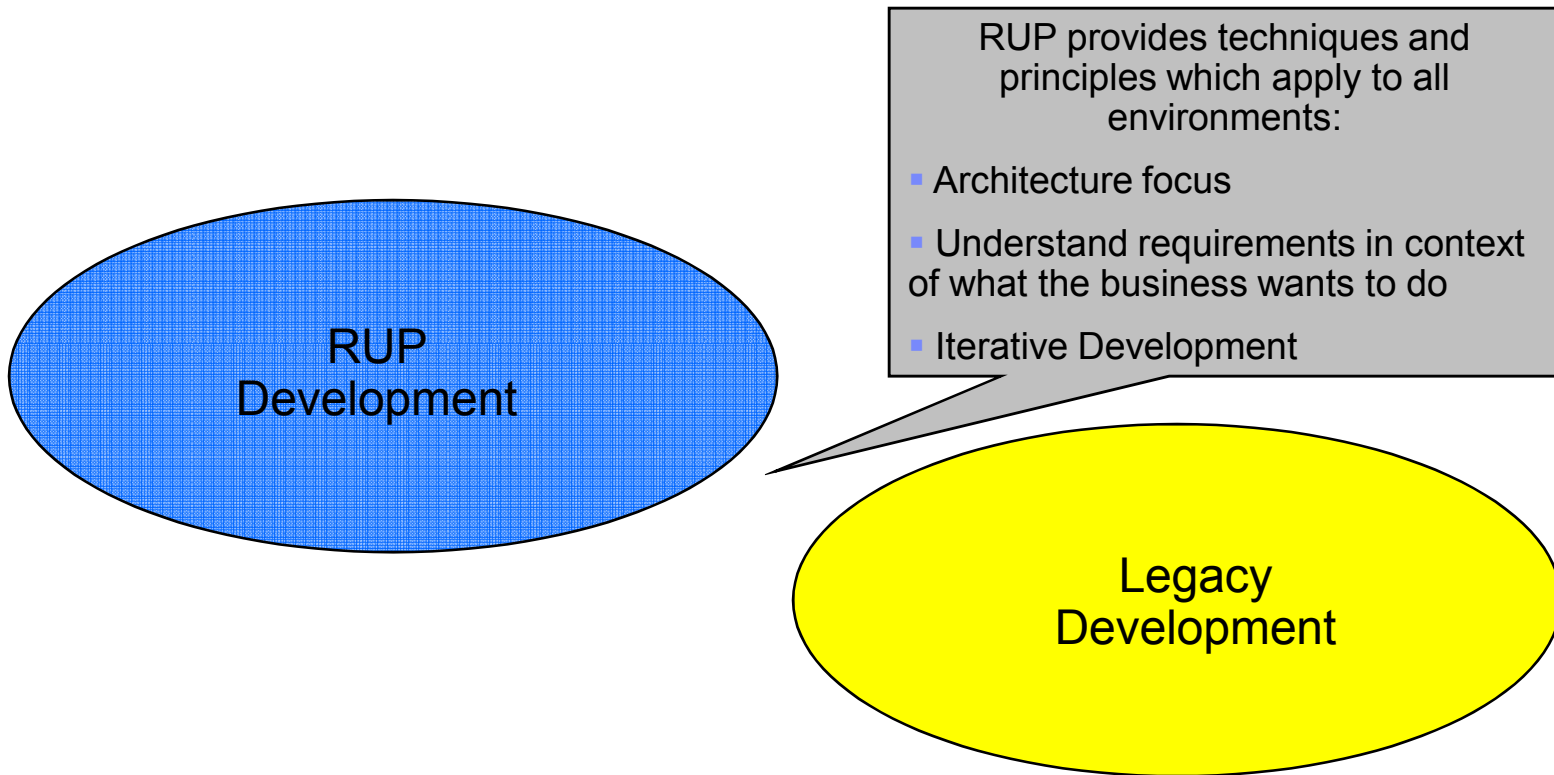
Time



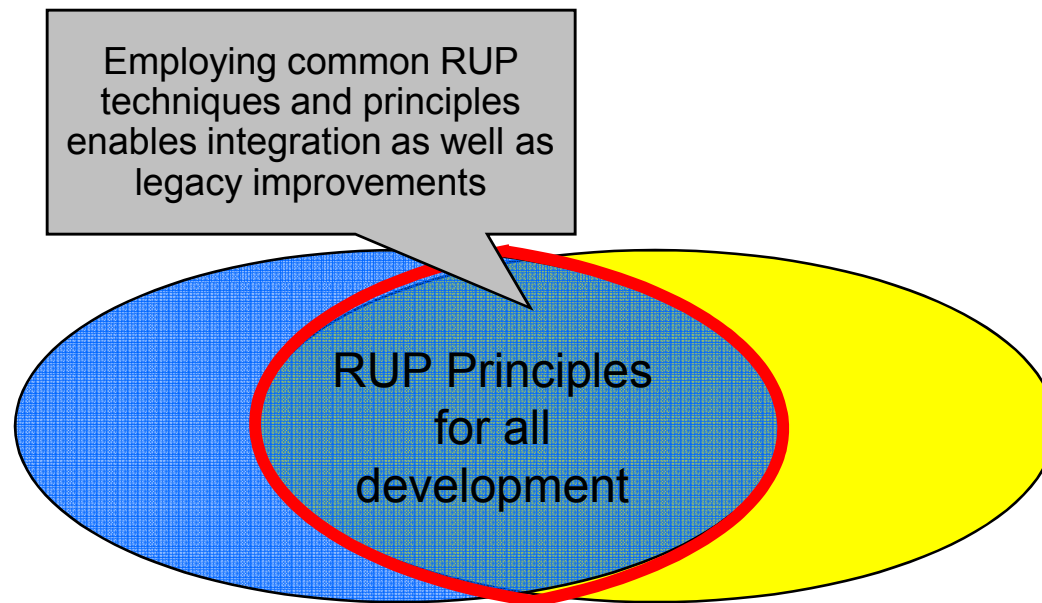
RUP and Legacy Development



RUP and Legacy Development



RUP and Legacy Development



The extent of adoption is governed by both the need and ability to invest in future development

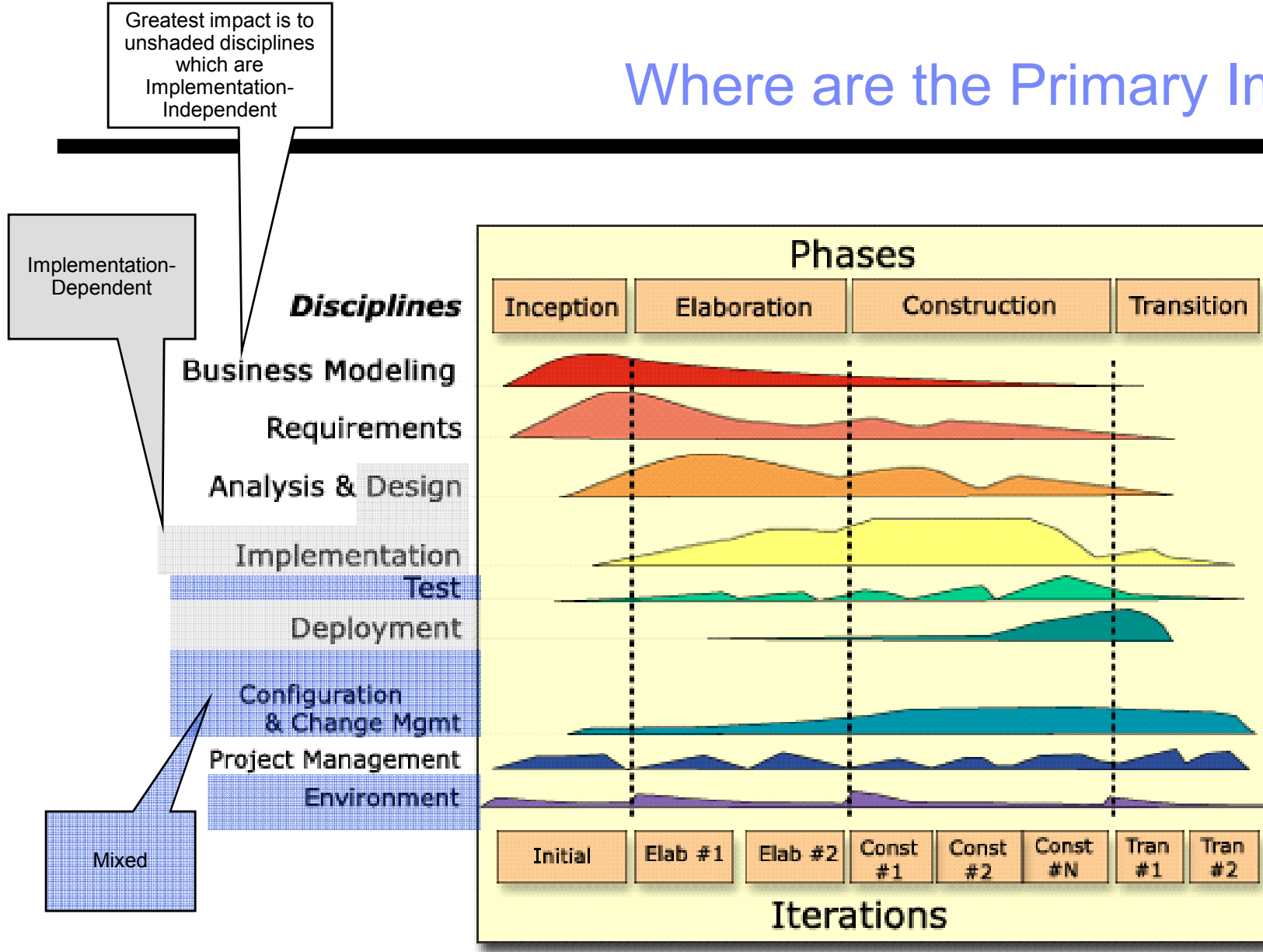


Why Consider RUP for Legacy Implementations?

- We all want to ensure good communication between:
 - ▶ Business and technical areas
 - ▶ Diverse technical areas
 - ▶ Globally distributed team members
- Most projects involve some level of integration between components and/or applications.
- The RUP principles are implementation-independent
- Development Languages are "just" an implementation concern. Each language has specific best practices for development, but the overlying analysis, architecture and design principles - and the thought-communication-techniques and tools – don't have to be different than what is contained in RUP
 - ▶ RUP provides a great way to keep an entire project “on the same page”



Where are the Primary Impacts?



Questions to Consider

- What is your appetite for tackling concerns of implementations which span languages and/or environments?
 - ▶ Communication of business requirements “across the divide”
 - ▶ Laying out an architecture so that everyone understands it
 - ▶ Laying out an initial design so each area understands what it does, what it passes (and to whom), and what it returns
 - ▶ Usage of an iterative approach
 - ▶ Modularity (or increased modularity) when a legacy language is the development language
 - ▶ Reuse of assets and thought process
 - ▶ Change impact analysis
 - ▶ Outsourcing of development or integration of purchased software



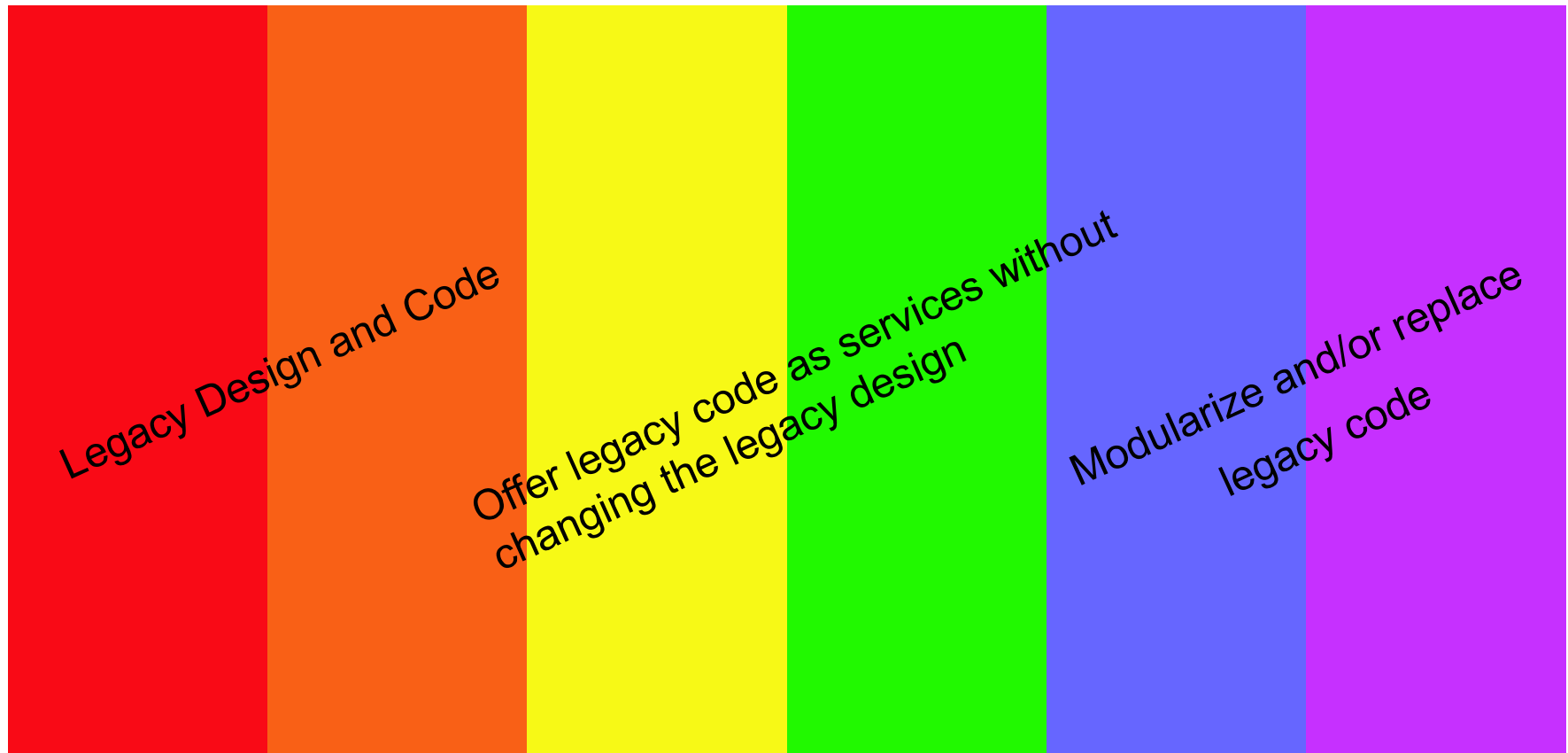
Project Experiences

- It is difficult to tackle all of these concerns at once
- Cultural change is difficult
- There's great success to be had – and everyone is happier, too!
- Keep in mind that RUP with legacy languages not one-size-fits-all
- Project experiences in which COBOL was one of the target languages
 - ▶ Requirements with use cases and/or activity diagrams
 - Test cases from the use cases
 - ▶ Analysis level sequence diagrams
 - ▶ Architecture and design with sequence diagrams across platforms
- Project experiences with RPG
 - ▶ The same!
- Projects – and organizations – do what they had “the will and the means” to do



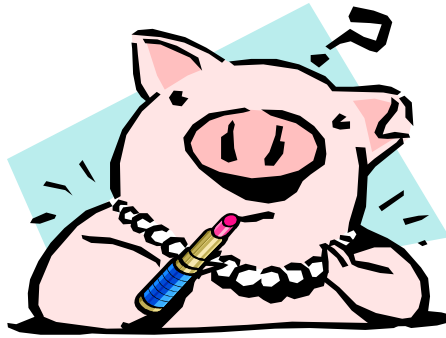
Design and Coding Styles for Legacy Development

“Spectrum” of options



Does it matter which style is chosen?

Encapsulating a brittle code-base only partially removes its constraints



- Signs of acceptable legacy design and code:
 - ▶ Stable or...
 - ▶ Can tolerate change
 - Table-driven or parameter-driven business rules and processing
 - Well-structured data - “one fact per field, one field per fact”
 - Modular code
 - Limited use of code structures which require extensive testing on change

Finding the Balance for Design and Code



"Stealth RUP"

Rules of Thumb for RUP Adoption With Legacy Implementations

Importance of Modularity, architecture,
Change impact analysis, reuse →

<p>Select more techniques from RUP to add clarity</p> <ul style="list-style-type: none"> • Use Cases • Activity Diagrams • Components as "classes" • Sequence Diagrams 	<ul style="list-style-type: none"> • Follow all RUP Principles • Special focus on RUP for business modeling, analysis, architecture, and testing • Pick appropriate design and coding style for legacy development with RUP
<p>Select techniques from RUP to add clarity</p> <ul style="list-style-type: none"> • Use Cases • Activity Diagrams 	<p>Select techniques from RUP to add clarity</p> <ul style="list-style-type: none"> • Use Cases • Iterative Development

Will and means to effect change →



Making RUP Legacy-Friendly?

A few words about the **Eclipse** Process Framework, OpenUp, and Legacy Implementations

- The Eclipse Process Framework (EPF) is an Open Source process framework consisting of a method authoring and publishing tool and foundational content for creating and/or adapting a large variety of processes to address IT needs
 - ▶ The Rational Method Composer was built on EPF
- OpenUP is the Open Unified Process
 - ▶ IBM donated a subset of RUP's content to provide initial content for OpenUp
- The core principles of OpenUP and those of RUP are well-aligned
 - ▶ Application of an iterative lifecycle that mitigates risk early and often, and shows results early and often
 - ▶ Focus on the collaboration within a development team including the product stakeholders to maximize results
 - ▶ Management of requirements in a form that represents stakeholder value and drives the development process
 - ▶ Cognizance of architecture as a means to increase quality and technical understandability



Making RUP Legacy-Friendly?

A few words about the **Eclipse** Process Framework, OpenUp, and Legacy Implementations

- What could EPF and OpenUP mean for implementation-dependent content for Legacy Implementations?



- EPF is supported by 20+ companies, including IBM
- Implementation-dependent content could be developed and made available as plug-ins for OpenUP (and RUP)

For more information...

- Within RUP or RMC, please check out
 - ▶ Concept: Legacy Evolution (in RUP 7.1 and beyond)
 - ▶ RUP for Systems Engineering (RUP SE) – Plug-in
 - ▶ RUP for Model Driven Systems Development (MDSD) – Plug-in; successor to RUP SE
 - ▶ RUP for Systems Z – plug-in
- For more information on the Eclipse Process Framework and OpenUP, please check out
 - ▶ <http://www.eclipse.org/epf/>
 - ▶ <http://epf.eclipse.org/wikis/openup/>



For more information...

- Selected articles of interest
 - ▶ Philippe Kruchten, “Using the RUP to evolve a legacy system”
<http://www-128.ibm.com/developerworks/rational/library/389.html>
 - ▶ Gary Evans, “Agile RUP for non-object-oriented projects”
http://www128.ibm.com/developerworks/rational/library/content/RationalEdge/sep03/m_rupagility_ge.pdf
 - ▶ Pan-Wei Ng. “Automated Modeling of Legacy Systems Using the UML”
<http://www.128.ibm.com/developerworks/rational/library/content/RationalEdge/sep02/AutomatedSep02.pdf>
 - ▶ Paul Reed, “Reference Architecture: The best of best practices”
<http://www-128.ibm.com/developerworks/rational/library/2774.html>
 - ▶ Anthony Crain, “The simple artifacts of Analysis and Design”,
<http://www-128.ibm.com/developerworks/rational/library/4871.html>
 - ▶ DJ de Villiers, “Introducing the RUP into an organization”
<http://www-128.ibm.com/developerworks/rational/library/916.html>
 - ▶ Dave Brown, “How does RUP SE apply to a system of systems?”
<http://www-128.ibm.com/developerworks/rational/library/2766.html>



THANK YOU



Learn more at:

- [IBM Rational software](#)
- [IBM Rational Software Delivery Platform](#)
- [Process and portfolio management](#)
- [Change and release management](#)
- [Quality management](#)
- [Architecture management](#)
- [Rational trial downloads](#)
- [Leading Innovation Web site](#)
- [developerWorks Rational](#)
- [IBM Rational TV](#)
- [IBM Rational Business Partners](#)

© Copyright IBM Corporation 2008. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, the on-demand business logo, Rational, the Rational logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

