The Requirements New Groove: Iterative Definition & Composition Techniques in Jazz

Alfredo Gutiérrez
Senior Manager, IBM WW Rational Technology Events
fredgz@us.ibm.com

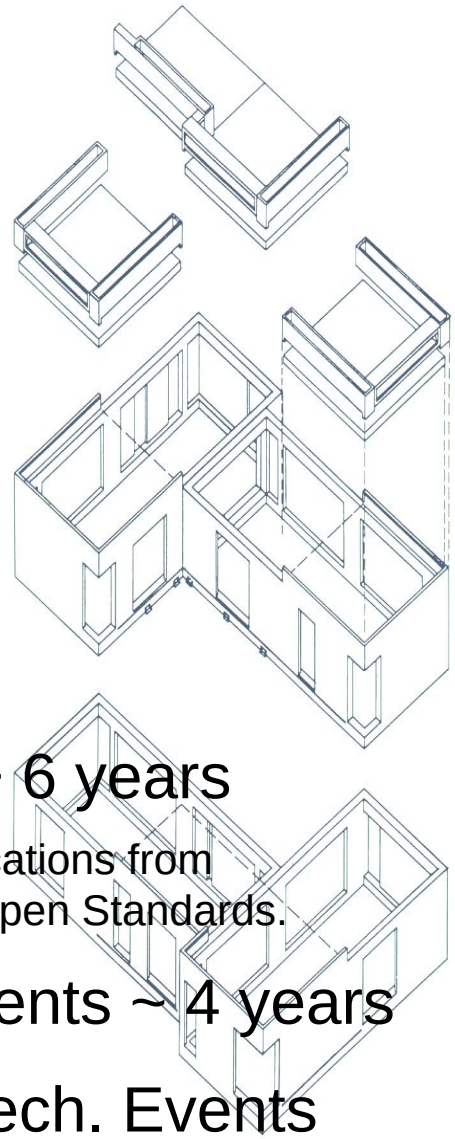
Rational. software

RDM05

© 2009 IBM Corporation

Go to IBM

About me…



- 22+ years in IBM
- Before that: Texas Highway Patrol ~ 7 years
  - Created DB of cases filed
  - Automated warrants procedure
- Operating Systems Development ~ 10 years
  - AS/400 / OS/400 System – Atlanta, Georgia
  - OS/2 Base Operating System – Boca Raton, Florida
  - OS/2 LAN & TCP/IP Networking – Austin, Texas
- Consultant / e-Business Solutions Designer ~ 6 years
  - Migration and transformation of client and business partner applications from client/server model to interconnected Web Applications built on Open Standards.
- Manager, developerWorks WW Technical Events ~ 4 years
- Today: Senior Manager, IBM Rational WW Tech. Events

Agenda

- Collaborative software development vs. Jazz
- The requirements challenge
- Introducing Rational Requirements Composer
- Iterative requirements
- 5 recommended steps to make it happen
- Rational Requirements Composer usage tips
- Demonstrations

Jazz

- "**Jazz**" a musical art form rooted in West African cultural and musical expression and in the African-American Blues tradition, with ***diverse influences over time***
- Jazz spawned subgenres, from New Orleans Dixieland (1910), to big band-swing (30's), bebop (mid-40's), Latin jazz ***fusions*** such as Afro-Cuban & Brazilian jazz (50's, 60's), then jazz-rock (70's, 80's).
- As Jazz spreads globally, it takes on flavors of the local & regional cultures. It a***dapts to its varied environments***; giving rise to many distinctive styles.

Improvisation

- **Improvisational** Jazz
- **Improvisation:** The practice of making and creating, in the moment and in response to the stimulus of one's immediate environment…
  - Adaptations result in the invention of new thought patterns, new practices, new structures or symbols, and new behaviors.
- Business **improvisation** is the ability to access creativity in the moment, under pressure, and in the given environment in order to solve problems.
  - The practice necessary to achieve objectives despite unexpected events or changes.
- We need flexibility and creativity in order to succeed…

The requirements challenge

- Do we really know the true needs early in the project?
- Can the customer really ‘sign-off’ early on the final requirements?
- How do we determine priority?
- Where do we set the baseline?
- Do we understand the real business and technical risks?
- Does anyone ever read those 100+ page documents?

Office tools exemplify the problem

- Microsoft Word
  - Track Changes, Comments, Formatting nightmares, HUGE documents
- Excel spreadsheets for requirement sets
  - Tabular format helps (at first) until the Tabs start to grow
- Visio to model process, flows, screen designs
  - Slightest change requires many manual updates to the same component
- PowerPoint as communication vehicle
  - Dozens of presentations to crawl through for information
- No way to establish relationships and link artifacts
  - Have to have endless meetings to put together the pieces

Q: *Have requirements evolved into what they are today in order to accommodate the functionality of the tools that are being used to represent them?*

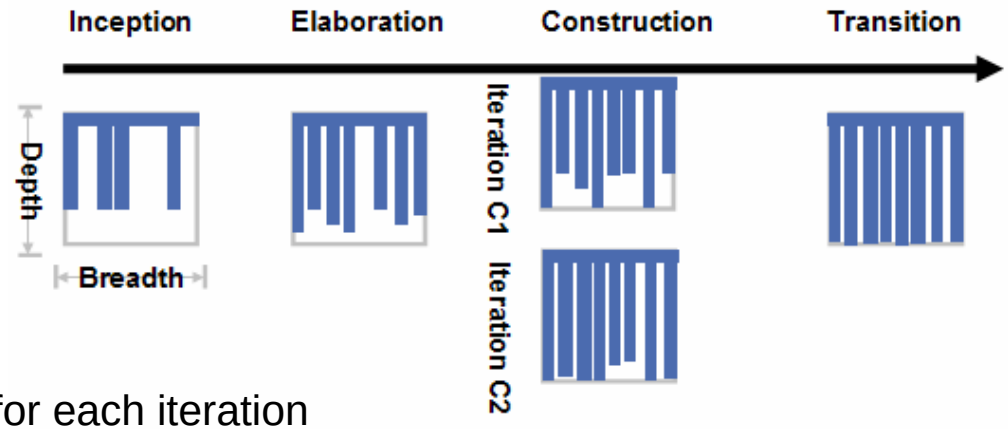
Introducing Requirements Composer

- Music to my ears!
- A bridge across the tool silos
- Effortless communication and collaboration across project teams
- All requirements artifacts can live in the same repository and be accessed via the same requirements solution
- A way to capture, connect, organize, and understand the complex web of requirements

Moving towards a more agile approach

- Link documents, images, diagrams, and artifacts as they are collected and evolving
  - Create meaningful relationships across artifacts to paint the whole picture to the team
- Provide an accessible repository on the web for stakeholders to review and comment as needed
  - Invaluable to projects with GR component – less need for midnight calls!
- Resist the Giant Requirement Up Front (GRUF) temptation
  - Introduce a less rigid, more flexibility requirements process
- Emphasis on the value of modeling
  - Fewer words, more pictures is always a good thing!

Iterative requirements

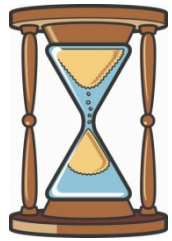


- Enable early user feedback
  - The aim is to have something tangible for each iteration
- Focus on the business and technical risks first
  - Early mitigation = less flux later on
- Testing and integration are continuous
  - Easier/less costly to fix defects earlier
- We don't (and can't) know everything 'up front'
  - Iterations serve to paint the picture incrementally
- Team learning (and therefore effectiveness) is enhanced
  - Apply lessons learned from the previous iteration to the next

Five techniques to make it happen!

1. Timebox
2. Establish breadth
3. Prioritize
4. Go deeper
5. Repeat

1. Timebox



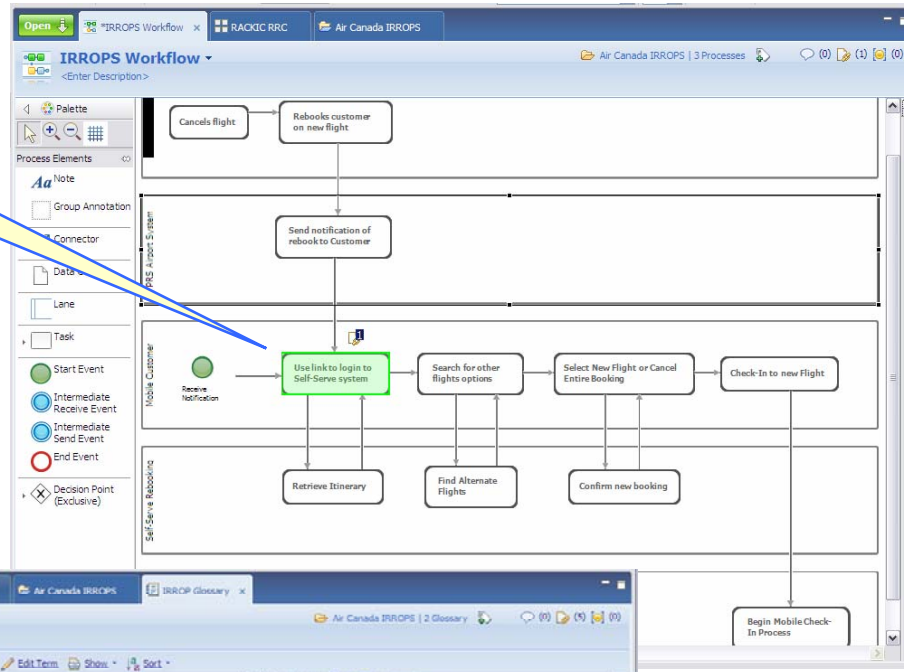
- Direction, not perfection
  - Develop requirements in phases with a time-box for each phase
  - Each requirements cycle is a short, specific time frame
- Make assumptions, then move on
  - Assumptions will either fall out of scope, or become requirements
- Don’t capture too much detail
  - Doing so can lead to analysis paralysis

Composing begins during the early days!

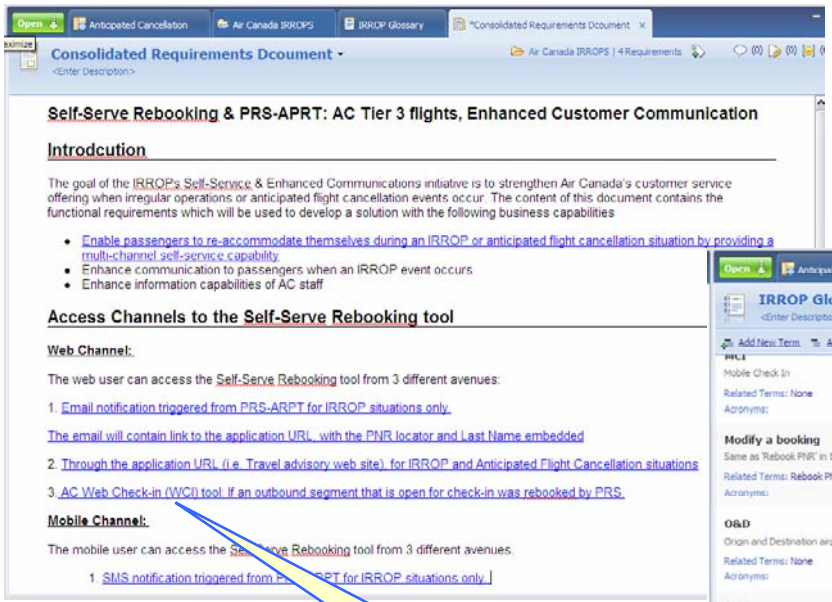
- Capturing high-level stakeholder needs
  - Vision document, Features, Solution Proposal (services), Business Requirements
- Business process sketches, glossary of terms give context
  - Related terms and synonyms are especially helpful for new team members
- Relationships are established by linking
  - Can import PowerPoint, Excel, Word, Images, and even capture e-mails in rich text
- Decisions and assumptions documented
  - So the team does not waste time revisiting again

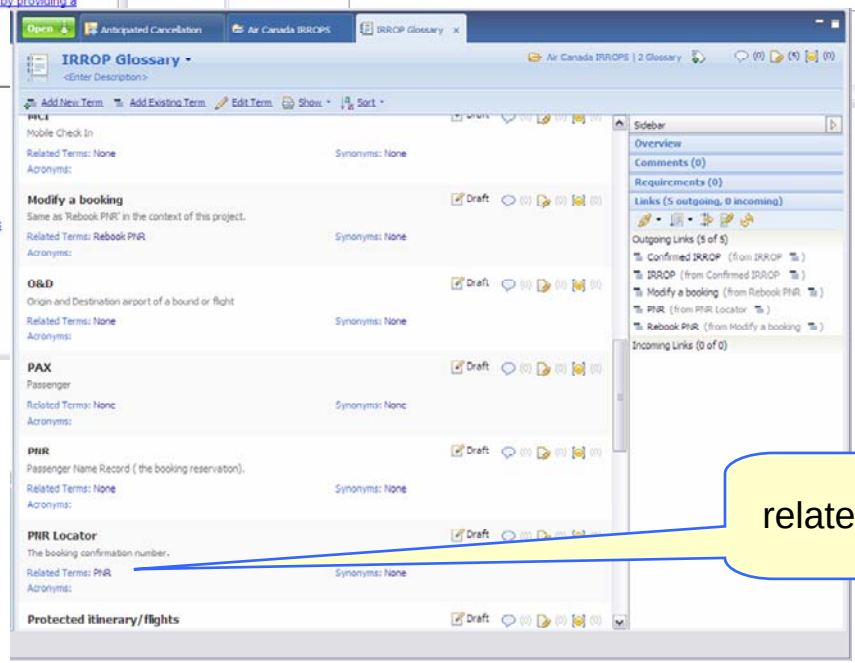
✓ Business process

highlighted comments



✓ Rich text documents





linked requirements

related terms

✓ Project glossary

Discover a new requirement and document it

Discover new requirement

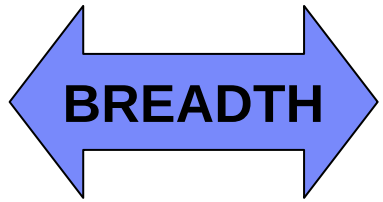
Add information to project document

Tie it to existing glossary information

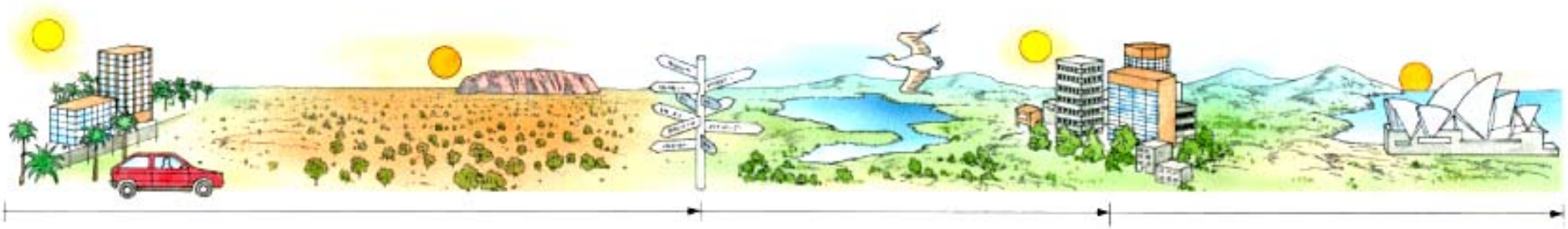
Add a comment for other project members

Identify the information with a tag

2. Establish breadth



- Understand the ‘lay of the land’
- Identifying the boundary: scope of automation
  - High-level requirements covering the *entire* breadth of the system
  - Staying within the time-box: whatever gets missed here can move to the next iteration
- Begin with a list of use cases and actors
  - Aim for a complete actor and Use Case set, start with outlines, details then get added incrementally

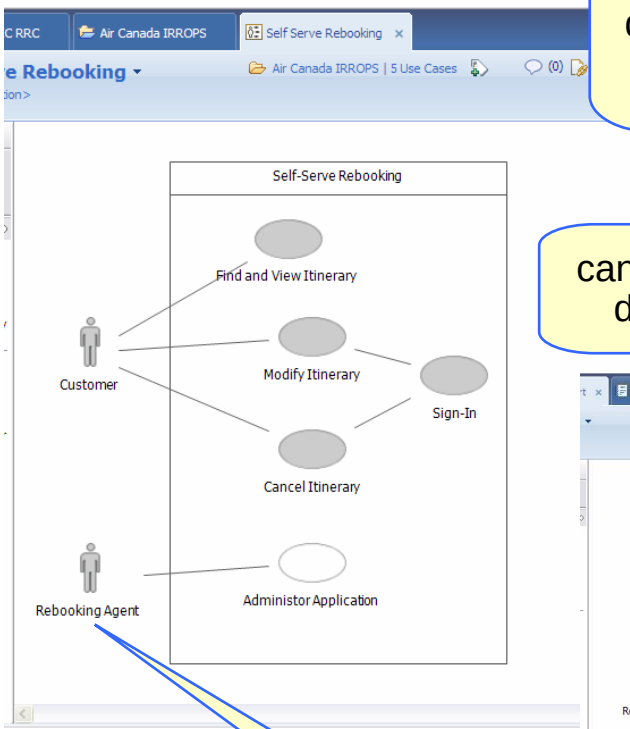


The use case model in Composer

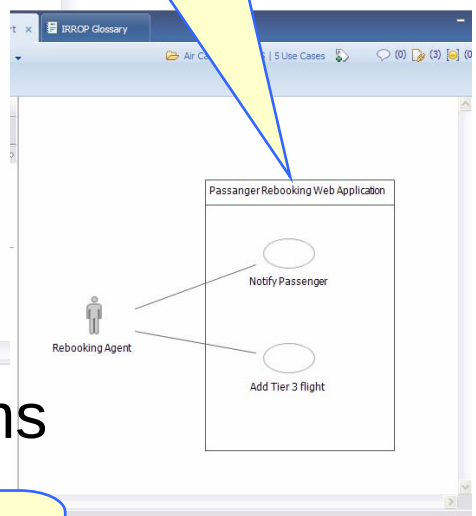
- ‘Automated’ tasks in business process model
  - Helps determine the scope of the system – black box for system use cases
- Aim is to ‘discover’ all use cases
  - Usually do not get it right the first time - need to combine or merge as needed later
- Sketch of the use case diagram
  - UML notation showing system boundary, actors & use case relationships
  - Initially start with a sketch, can turn into reusable components later
- Outline of use cases and actor descriptions
  - Use Case document template- provide brief description and identify major flows only

✓ Use case specifications

double clicking on diagram opens use case description

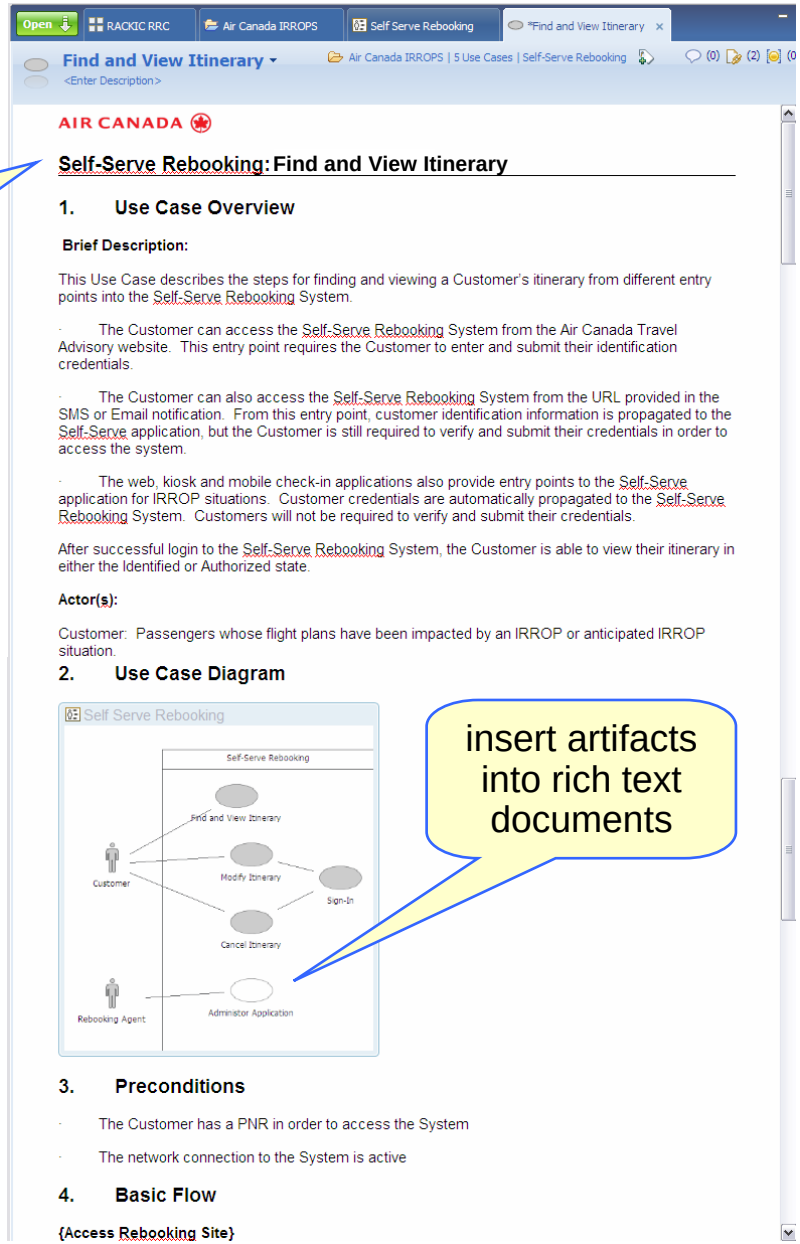


can sketch first, define later



✓ Use case diagrams

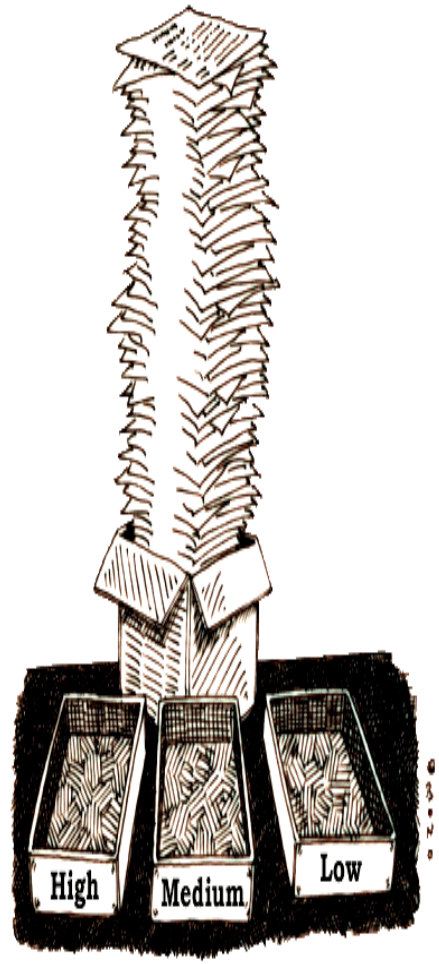
reusable components



insert artifacts into rich text documents

3. Prioritize

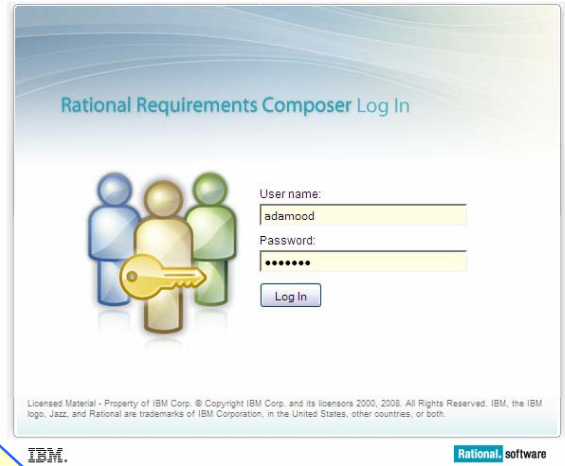
- According to RISK
  - Technical risk: working with architect to determine architectural risks
  - Business risk: good to challenge the customer
- According to BUSINESS VALUE
  - Which UCs and functions will deliver the biggest bang for the buck?
  - Pareto's Principle - **80/20 rule**
- Be ready to do it again! Will have to re-prioritize requirements in the next iteration…



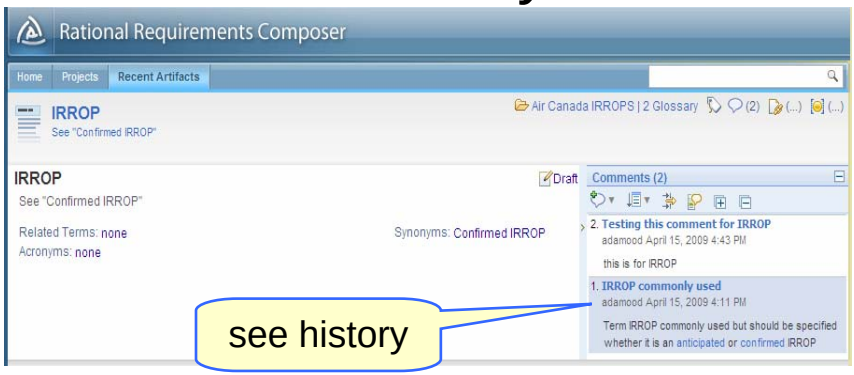
Composer can help!

- Define and set attributes (priority, difficulty, risk)
  - These will not be lost and can later be imported to RequisitePro for management
- Commenting features to communicate and help achieve consensus
  - Each stakeholder may have their own idea/agenda – facilitate difficult discussions
- Business partners, stakeholders, development teams that do not have client access can take advantage of the web
  - Easy to use and navigate as needed – can send URL links to specific artifacts
- Define a strategy for how you will use attributes and tags consistently in RRC

- ✓ Read-only web access



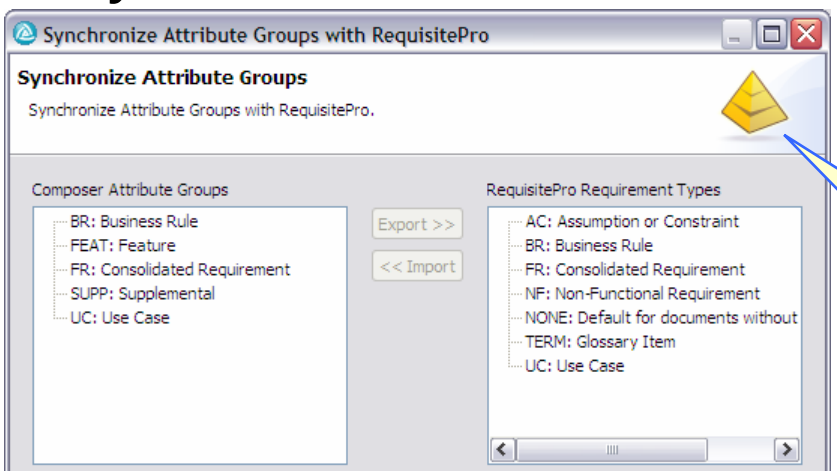
- ✓ Comment on any artifact

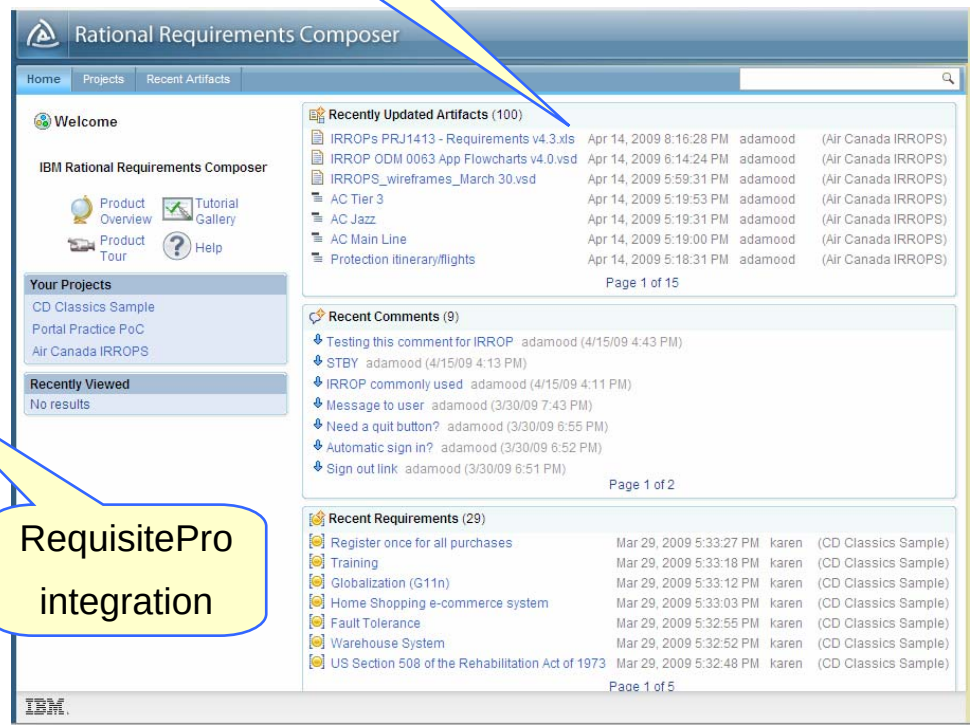


dashboard style home page: click on link to open related artifact

see history

- ✓ Synchronize attributes





RequisitePro integration

4. Go deeper

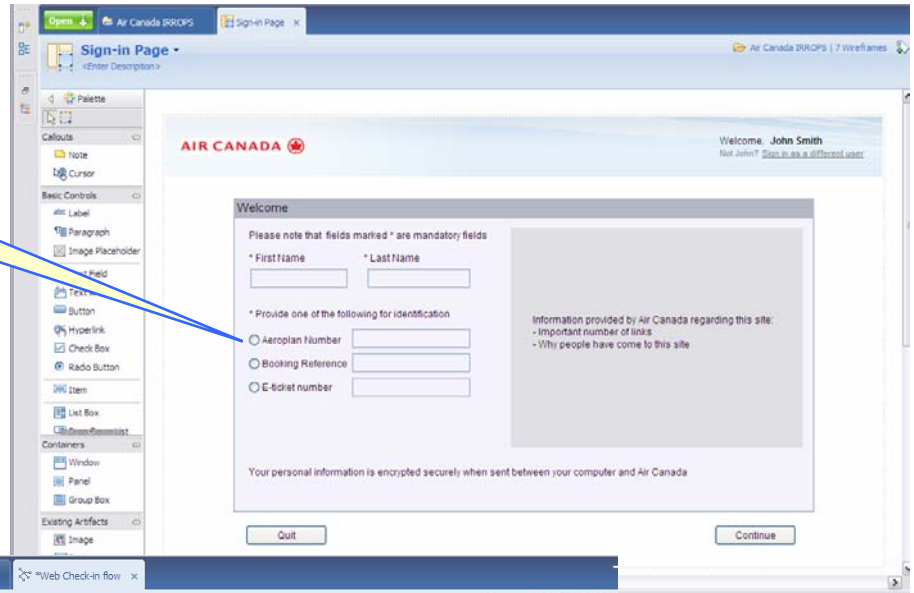
- Defining depth on the features / functions that are at the top of the list
  - The first few, highest priority UCs are addressed first – business rules & messages are referenced as well
- Moving from UC outlines to detailing all flows and major scenarios
  - Going through this process, it is not uncommon to discover that one UC should actually be two, or vice versa
- Scenarios are invaluable!
  - At the heart of the iterative approach, are the end to end user scenarios that can be coded and tested for that iteration…

Detailing requirements in Composer

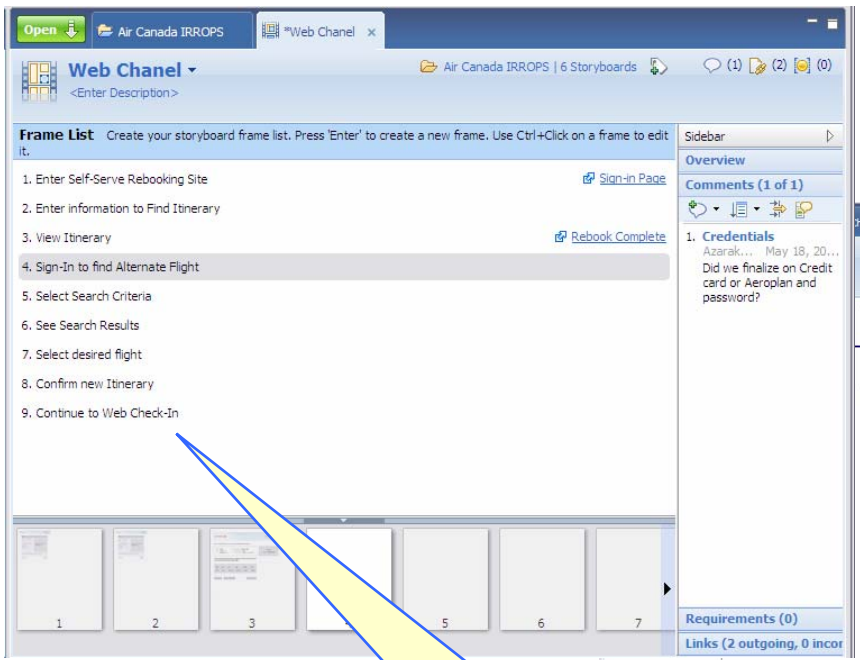
- High priority (or riskiest) use cases first
  - Business Rules are captured separately and referred to at various step in the flows
- Meaningful end to end scenarios
  - Like telling a story - great way to communicate in a way everyone can understand
- Wireframe sketches and user interface components
  - Wireframes are an excellent ‘communication vehicle’ – should be used as such ☺
- Storyboards help visualize the scenarios
  - Walking through a flow using mockups will help to ask the right questions

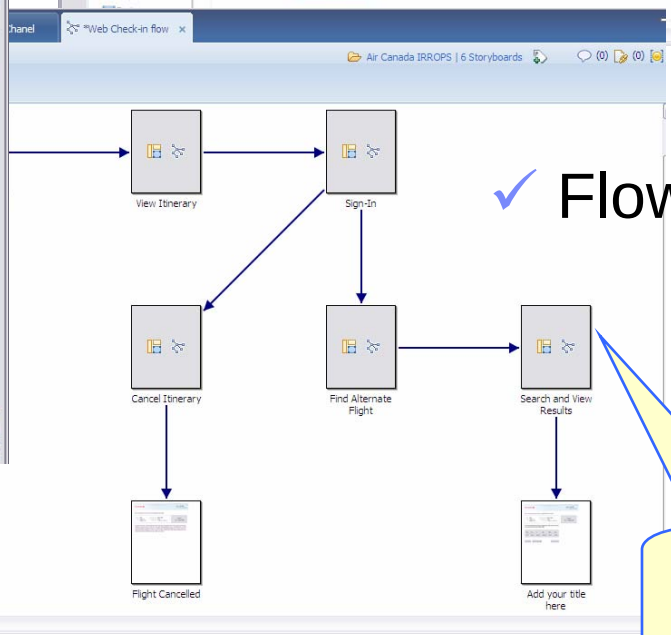
✓ Wireframes

reusable components



✓ Storyboarding





✓ Flow Diagram

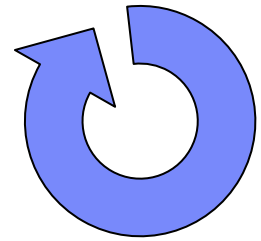
visualize step by step scenarios

wireframe placeholders

Show necessary changes to the process

See project activity and comments → Identify process changes → Create new sub-process → Link information → Tie to term in glossary

5. Repeat



- The number of iterations may vary, but each time through we still follow the four previous steps
  - Timebox
  - Breadth
  - Prioritize
  - Depth

- As we iterate, the emphasis shifts away from the breadth of the system to developing more depth
  - The emphasis also begins to shift from definition to management of the existing requirement set
  - Managing change becomes a big component in later iterations

Requirements identification and creation

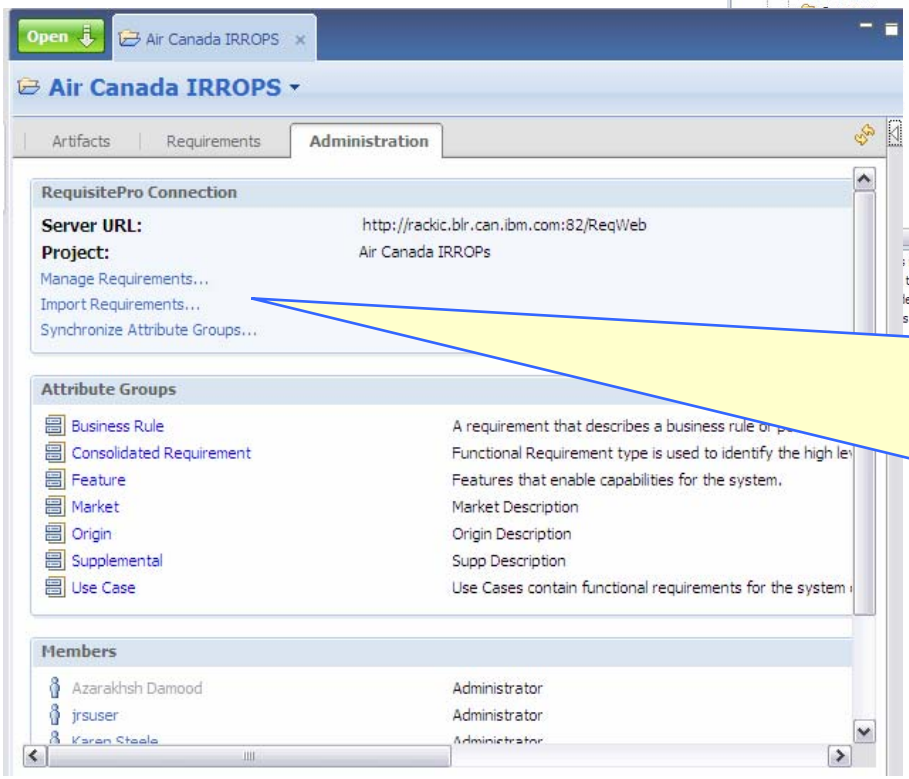
- As we repeat …consistency is key…
  - Requirements Definition Plan helps to plan upfront what will be ‘outcome’ from Composer
- Requirements created in Composer based on any artifact
  - Each requirement creates a rich text document
  - Can imbed or link other artifacts into the requirement (ie. Screenshot/wireframes)

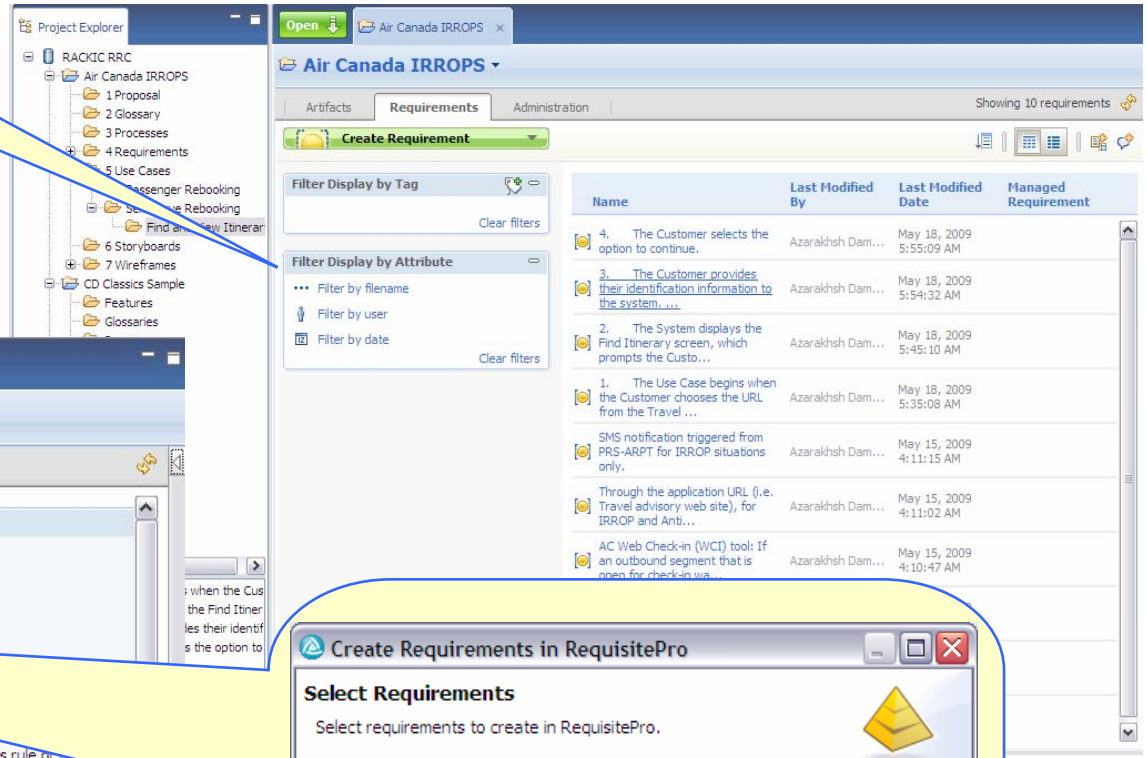
- For large projects, RequisitePro needed for management
  - RequisitePro has many features to help with more complex reporting, traceability needs
- Early lifecycle adoption identified as key success criteria
  - Requirements definition happens during proposal phase on most services engagements
- Can trace to test cases in RQM and tasks in RTC

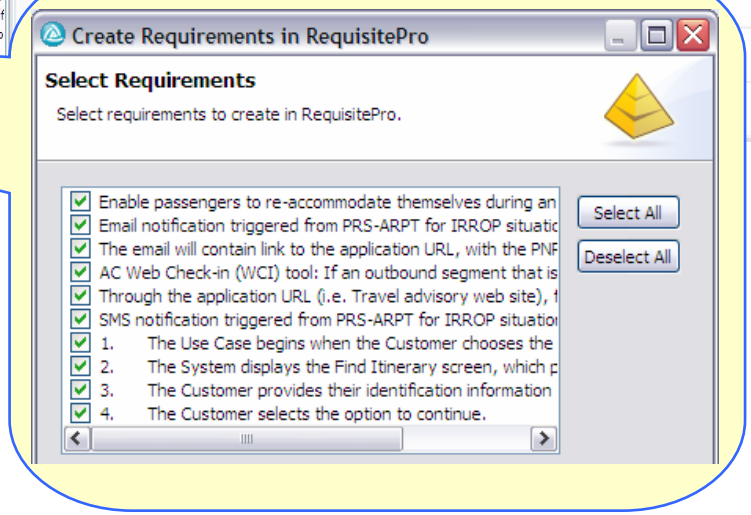
filter requirement sets by attribute

✓ Requirements Management

✓ Rational RequisitePro







Real return for our investment dollars

- Improved communication saves time, reduces rework, promotes consistency, and improves quality
  - Also speeds up the on-boarding process for new team members
- “Requirements management remains single biggest reason for project failure” according to industry reports
  - Composer embodies leading requirements practices for delivery success

So, I hope you are ready to …

- Try a more agile, incremental requirements definition and elaboration process on your next engagement or project
- Discover the benefits of enhanced collaboration and communication among your team (and stakeholders)
- Explore this flexible tool to provide a more complete, contextual, holistic view of your project requirements
- Find your organization's requirements' groove!

Questions

Helpful links & resources

- Rational Requirements Composer
  - Information Page: http://www-01.ibm.com/software/awdtools/rrc/
  - Rational Requirements Composer Overview Demo: http://www.ibm.com/developerworks/offers/lp/demos/summary/r-rrcoverview.html
  - Rational Requirements Composer & RequisitePro Integration Demo:
  - http://www.ibm.com/developerworks/offers/lp/demos/summary/r-rrcreqpro.html
- Developer Works discussion forums
  - Rational Requirements Composer Forum: http://www.ibm.com/developerworks/forums/forum.jspa?forumID=1546
  - Rational RequisitePro Forum: http://www.ibm.com/developerworks/forums/forum.jspa?forumID=340
- IBM Global Business Services
  - http://www-935.ibm.com/services/us/gbs/bus/html/bcs_index.html

© Copyright IBM Corporation 2009. 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, Rational, the Rational logo, Telelogic, the Telelogic 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.