IBM Innovate 2011
Using IBM® Rational® Requirements Composer Sketching Tools to Visually Communicate and Elaborate Requirements

Srinivas Mandava
Staff Software Engineer, IBM
smandava@in.ibm.com
RDM-1460

IBM Software
Innovate2011
The Premier Event for Software and Systems Innovation

Software. Everyware.

August 9-11, Bangalore | **August 11,** Delhi

Please Note:

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Agenda

- Rational Requirement Composer features
- What are the “UI sketching tools” within Rational Requirements Composer?
- Benefits of visually communicating requirements
- How we used sketching tools in our work on Rational Requirements Composer
- Tips and techniques for using sketching tools more effectively

IBM Rational Requirements Composer Features

Requirements Management for the Development Lifecycle

Rational Requirements Composer

Definition
- Rich-text documents
- Diagrams: Process, Use Case
- Storyboards, UI sketching & flow
- Project glossaries
- Templates (formal/agile)

Improved!

Visibility
- Customizable dashboards
- Analysis views
- Collections
- Milestone tracking & status

Collaboration
- Review & Approval
- Discussions
- Email Notification

NEW!

Management
- Structure, Attributes/Types
- Traceability, Filtering, Tags
- Baselines, Change History
- Reuse
- Reporting Metrics & Doc.

Improved!

Lifecycle
- Central requirements, test, & development repository
- Common administration and role-based user licensing
- Warehouse reporting

NEW!

Planning
- Integrated planning
- Effort estimation
- Progress tracking

What are the “UI Sketching Tools” within Rational Requirements Composer

IBM Rational Requirements "UI sketching" tools

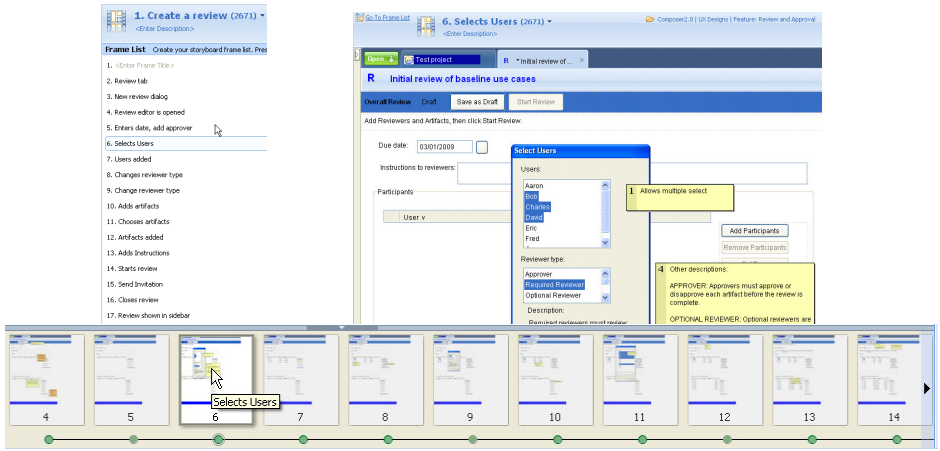
"User interface sketching tools" generically refers to the subset of Rational Requirements Composer capabilities for illustrating the user interface of a software application (storyboards, user interface sketches, and user interface parts).

Primary purpose is to quickly and easily "sketch" the user interface and scenario flow with just enough detail to facilitate conversations about the requirement in a more concrete context.

Not for creating working prototypes or high-fidelity user-interface designs.

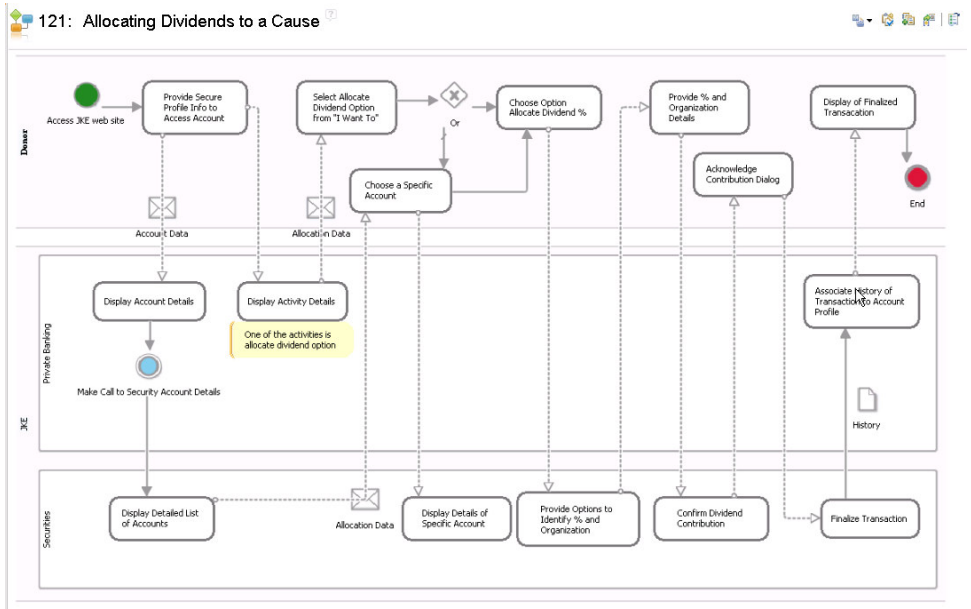
These are useful resources, but not necessary or efficient for higher-level requirement discussions at the elicitation and validation stage.

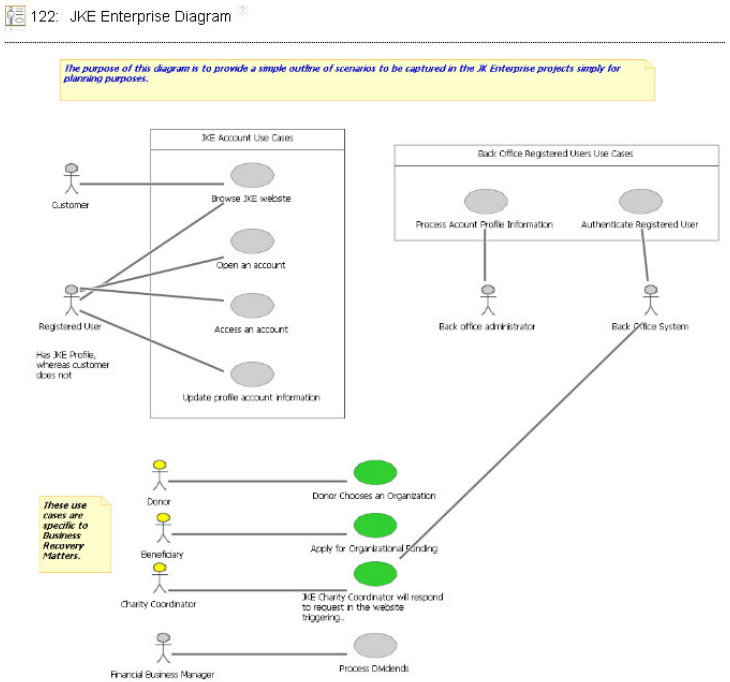
Can also be used to add more detail for direction during implementation.



Other visual techniques in Rational Requirements Composer

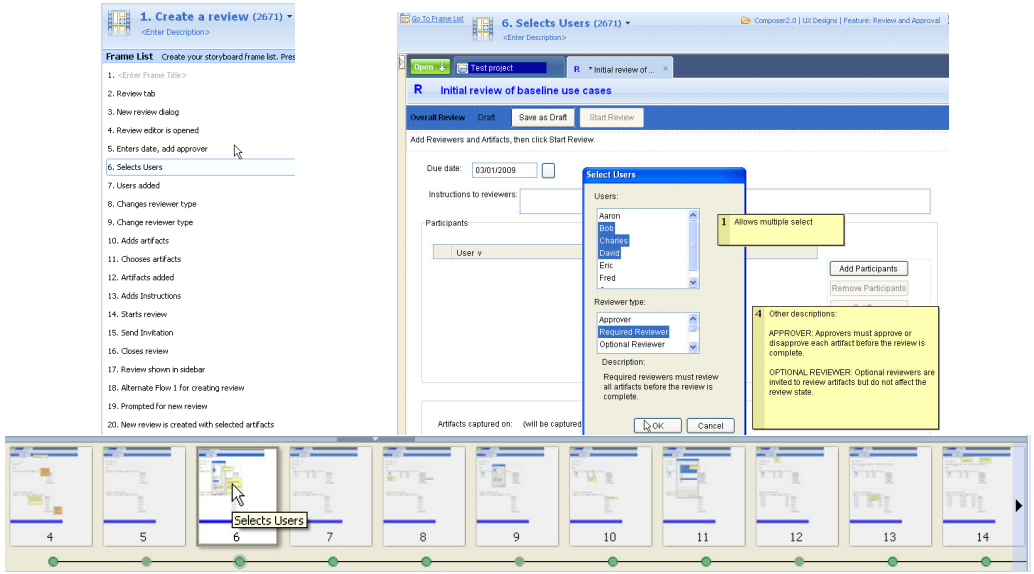
UI sketching tools are just part of the "visual" techniques for defining requirements in Rational Requirements Composer. There are also business process diagrams and use case diagrams for visually communicating and elaborating requirements.





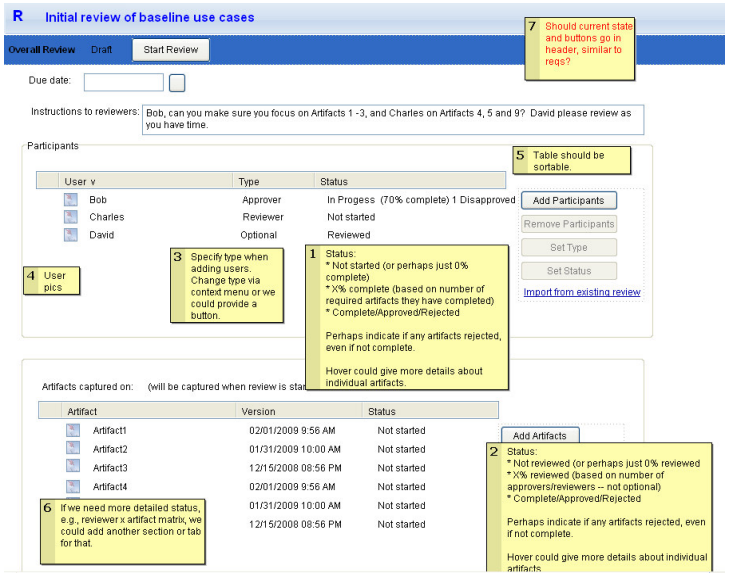
Storyboards

- Used to illustrate user scenarios and user interactions.
- Visually expresses or illustrates requirements that have a user-interface and user-interaction component.
- Uses a series of “frames” to show the progression of user-interface screens and actions performed by the user while performing primary tasks.
- Each frame can be constructed from previous frames, sketches, or parts, or by adding individual user-interface elements directly to the frame.



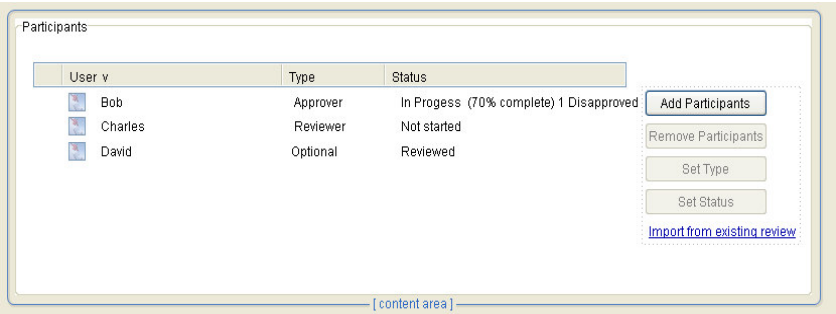
User Interface Sketches

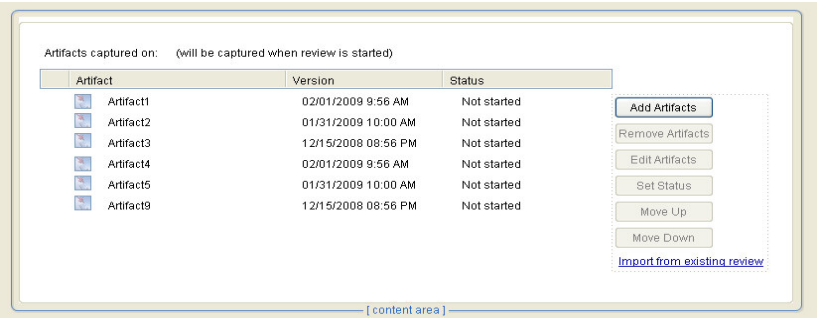
- For illustrating a single web page, screen or state.
- Can be used to visually express or illustrate requirements that have a user-interface component, but where there's not much user interaction
- Can be constructed from other sketches, parts or by adding individual user-interface elements directly to the sketch.



User Interface Parts

- For creating reusable components that are used in sketches and storyboards.
- Can be constructed of other parts or by adding individual user-interface elements.
- Key to rapid iteration of storyboards.





Benefits of visually communicating requirements

Why "visual communication"?

- To ensure that everybody is on the same page in understanding the requirement
- Bridges the communication gap between interested parties (stakeholders, business analysts, user experience, development, testing, etc.)
- Visual representations can often communicate requirements more efficiently and more effectively than just text. Provides a concrete context that customers/stakeholders can relate to.
- Forces thinking in terms of scenario and user interaction.
- Is helpful in identifying additional, related requirements, constraints, etc.
- Can help clarify scope of requirement

Why visual communications with Rational Requirements

- Integration with other requirement artifacts
  - Linking/traceability to related requirement artifacts (user requirements, use cases, business process diagrams, business rules, etc.)
  - Embedding visual artifacts within textual artifacts to create composite documents
  - Manage all requirement artifacts in the same system
- Integration with Rational Lifecycle Management applications
  - Relate artifacts with development work items in Rational Team Concert and test plans and results in Rational Quality Manager.

Why visual communications with Rational Requirements

- Ease of creating and modifying storyboards, sketches and parts
  - Employs reusable parts, making it easier to build up storyboards and ensure consistency
  - “Inheritance”, along with reusable parts, makes it easy to make changes to large number of related artifacts.
- Benefit from other Rational Requirements Composer features
  - On the web
  - Reviewing
  - Commenting
  - Artifact audit history
  - Artifact filtering and tagging
  - Traceability analysis
  - Reporting

How RRC product team used sketching tools in our work on Rational Requirements Composer

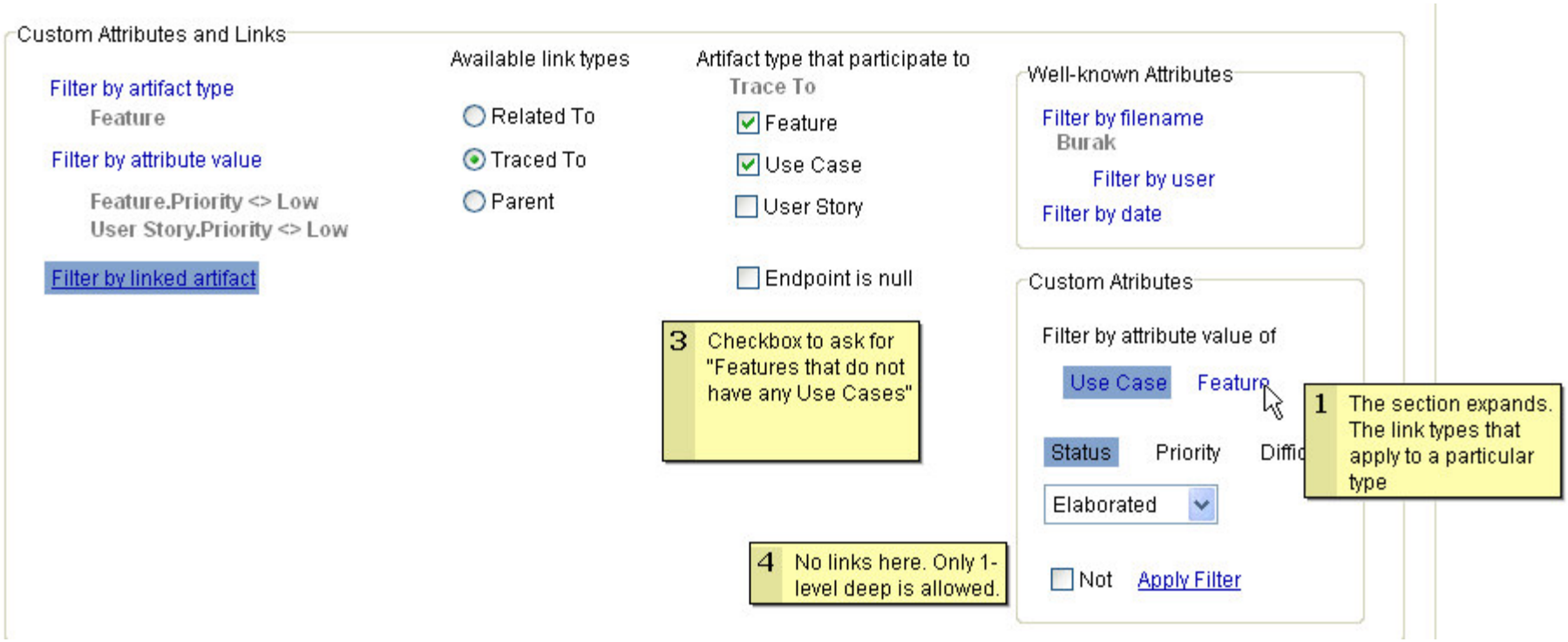
Our primary uses of sketching tools

- Understanding and elaborating on requirements
- Exploring the user interface and user experience design
- Proposing the user interface and user interaction
- Specifying the user interface design
- Helping frame and resolve issues during development
- Illustrate and capture UI patterns and guidelines

Understanding and elaborating on requirements

- To initially understand, elaborate and agree on the requirement
- Very early in the process. Usually based on a short text description of the requirement.
- This step is especially important for more vague and large-scope items. Text descriptions alone can leave a lot of room for misinterpretation.
- Usually created by business or user experience team, sometimes dev team
- Not elaborate or detailed. *Just enough to communicate the idea*.
- Helps facilitate discussion and elaboration of requirement
- Input for user experience designs, initial development estimates

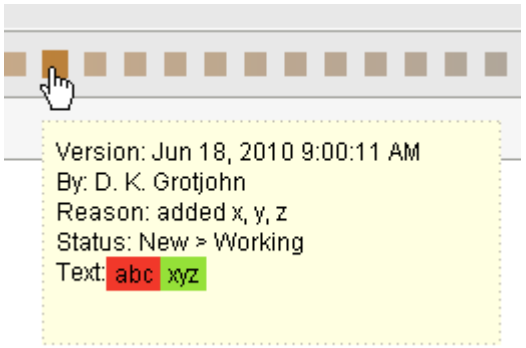
Example

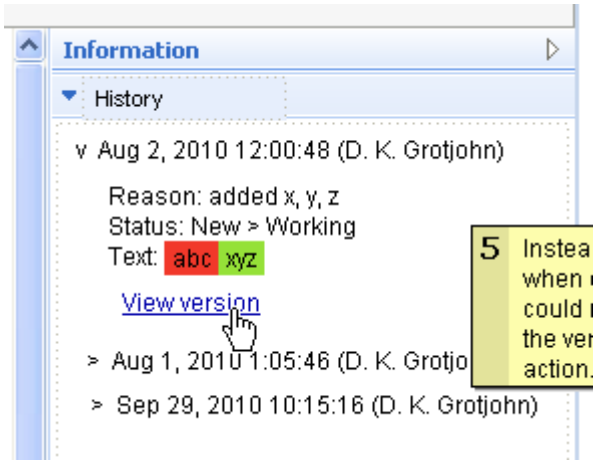


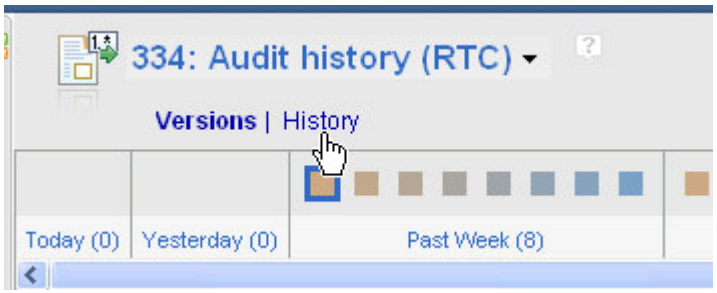
Exploring the user interface and user experience design

- Initial designs of the user interface and user experience
- Focuses on primary screens and primary user tasks
- More elaborate and detailed than initial sketches, but still low fidelity
- Includes many design options and iterations
- Primarily a user experience activity, but may require some input from business or development team
- Dependencies and interaction issues can start to be identified at this stage
- Taking advantage of re-use and inheritance is especially helpful here
- User experience identifies preferred design

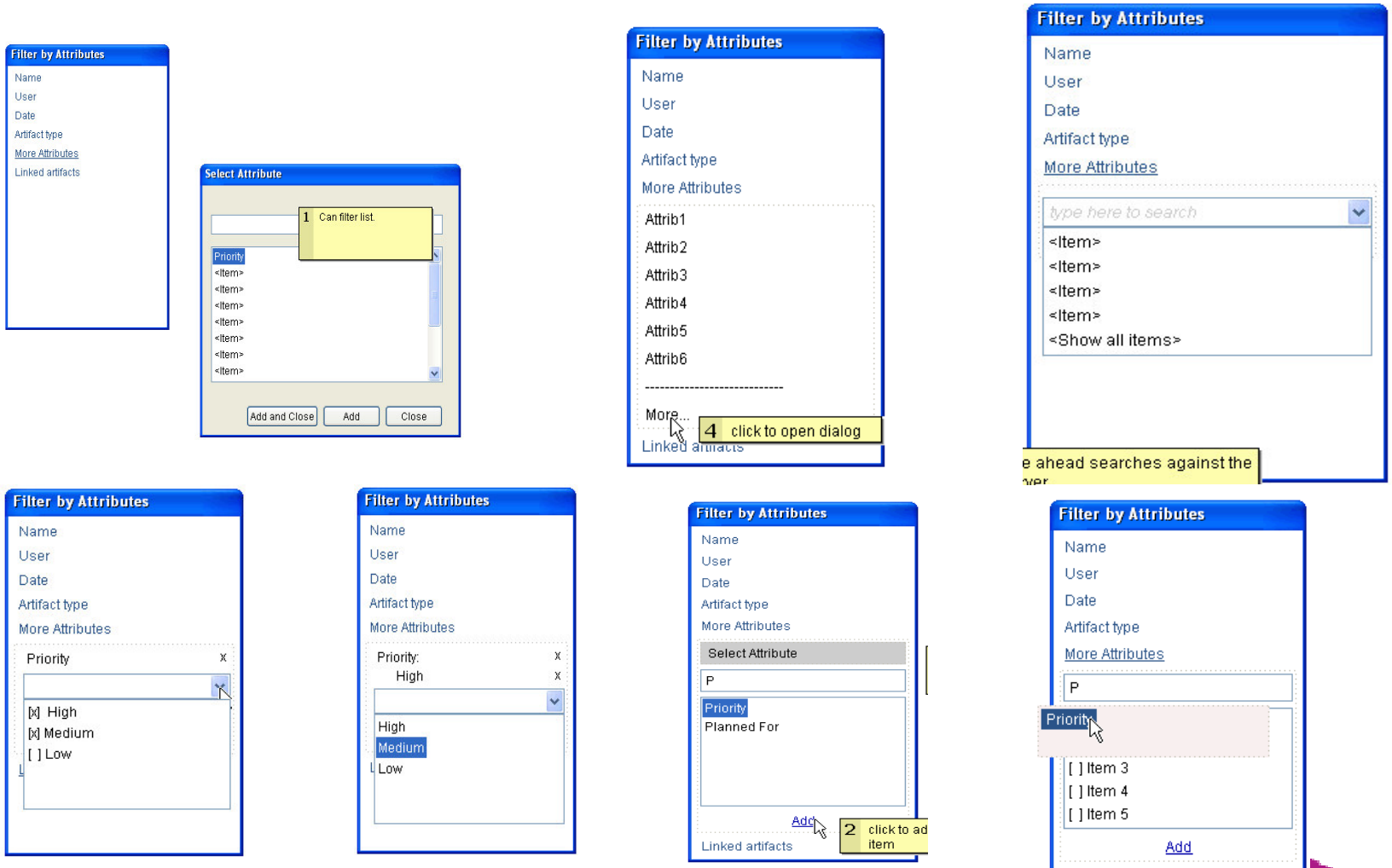
Examples: exploring options for audit history







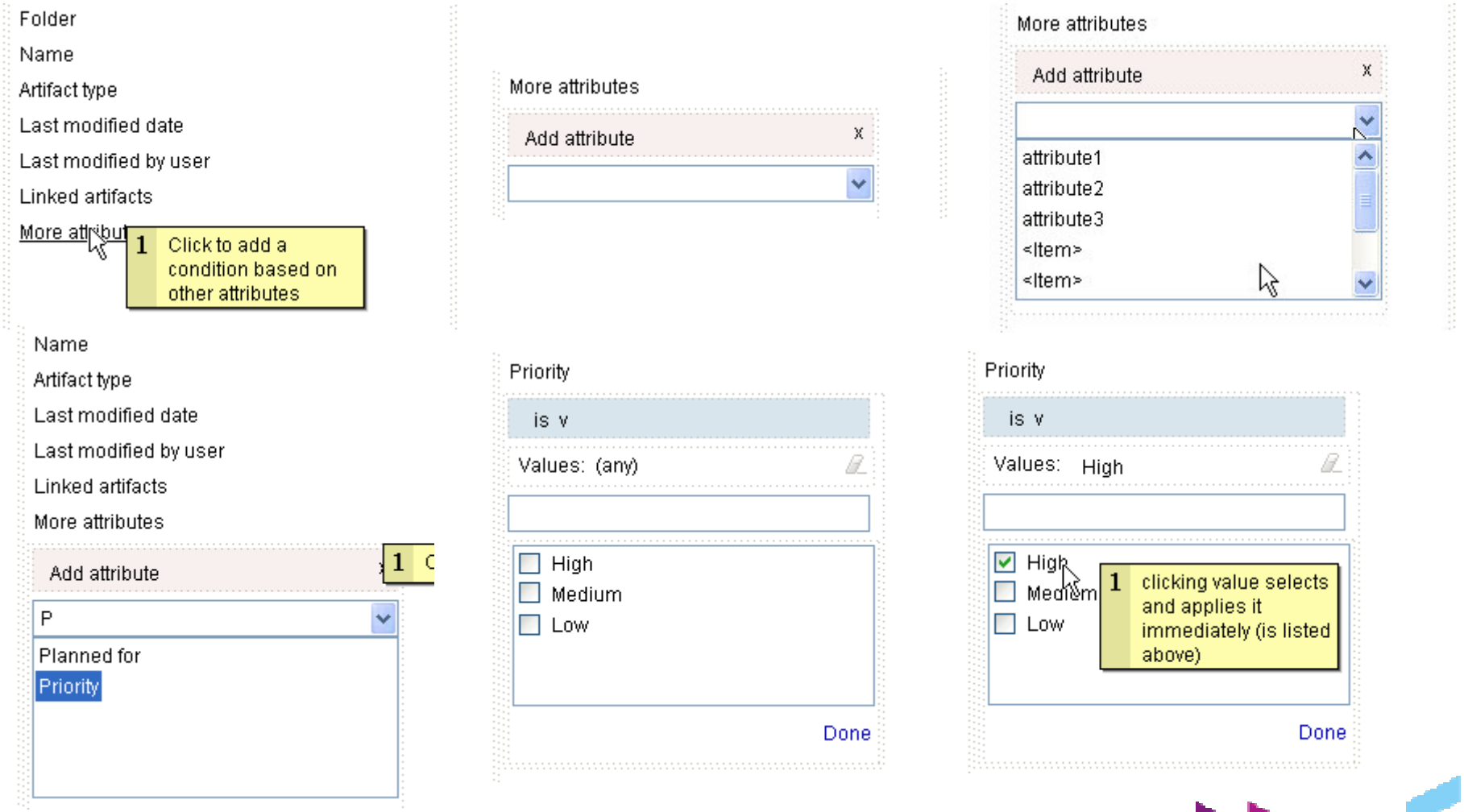
Examples: controls for attribute and attribute value filters



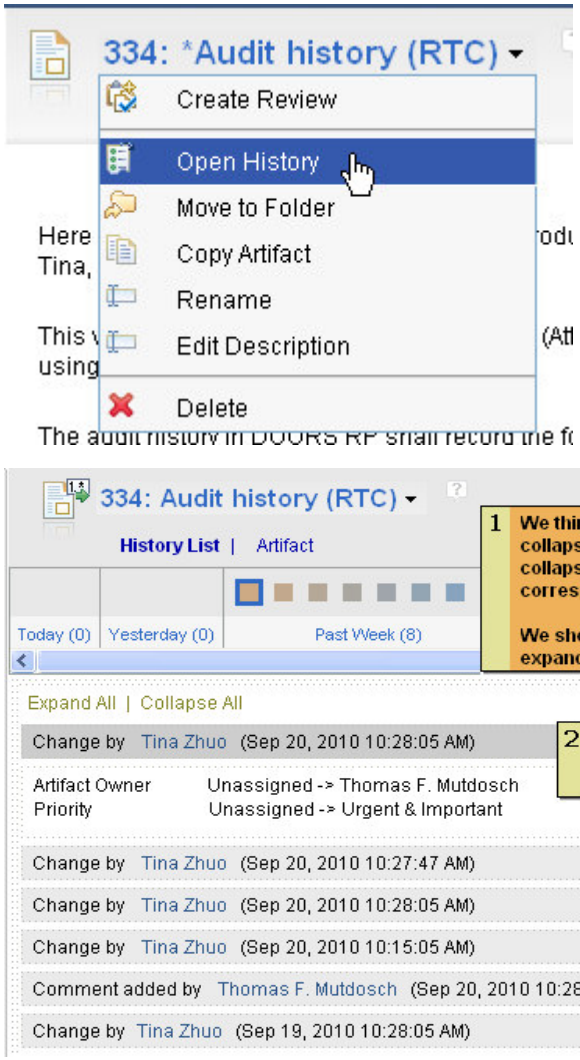
Proposing the user interface and user interaction

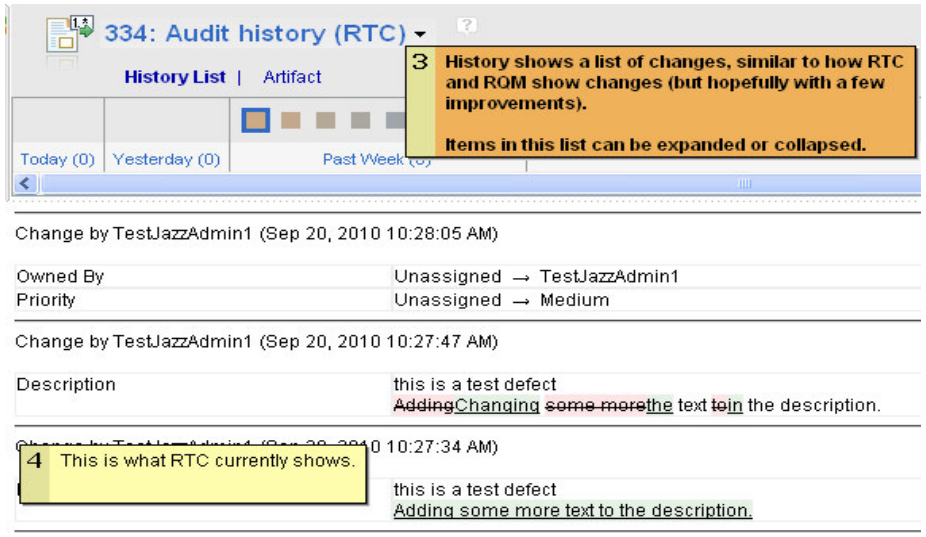
- User experience team narrows down and “presents” proposed UI designs
  - Usually just storyboards at this point
- More detailed, but still fairly low fidelity and focused on primary user tasks
- Business team confirms that it satisfies requirement
- Development team confirms that they can implement it, updates estimates if necessary
- Will often require further modifications to UI designs
- Can use to get feedback from stakeholders
- Other teams (e.g., test, user assistance, etc.) can begin to see how the requirement will be manifested
- Note that teams can see this work in progress as well

Example: Filtering by attributes



Example: audit history



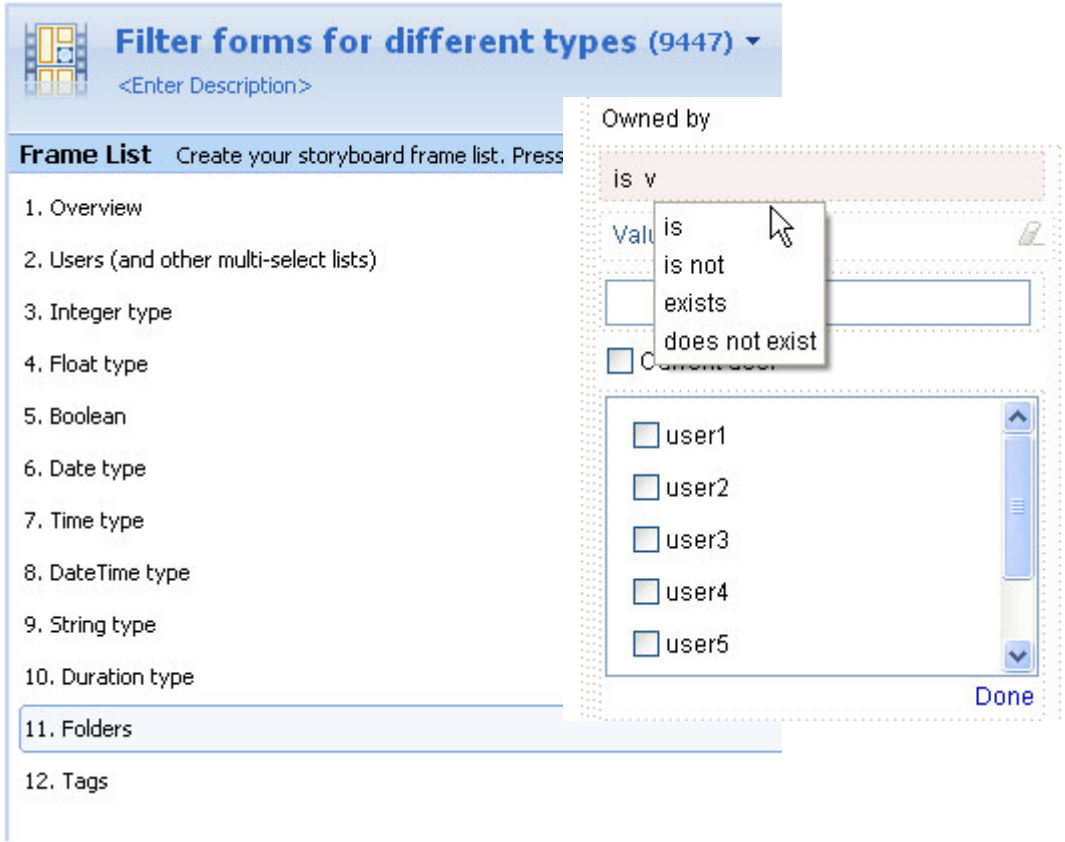


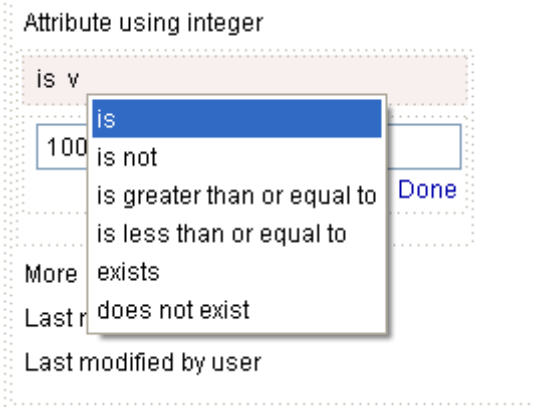
Specifying the user interface design

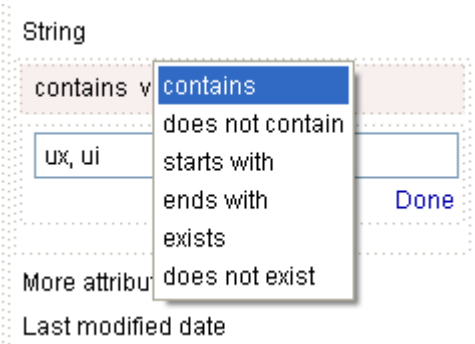
- Usually need to specify the design in more detail for implementation
- Include storyboards for secondary or alternative tasks and flows
- Add detailed descriptions to user interface elements and components
  - Add notes to sketches
  - Create a composite rich text document with textual descriptions and UI sketches or storyboards embedded in the document
- Helpful in identifying gaps, inconsistencies, problems, etc.
- Higher-fidelity prototypes may be created outside of Rational Requirements Composer
  - E.g., visual design team uses graphical editing tools
  - These external resources can be added to Requirements Composer repository (and linked to sketching artifacts), or can even be pasted into storyboards.
- Primarily intended for development team
  - Also useful for any team needing more details on interactions (e.g., test, user assistance)

Example: detailed designs for attribute filter forms

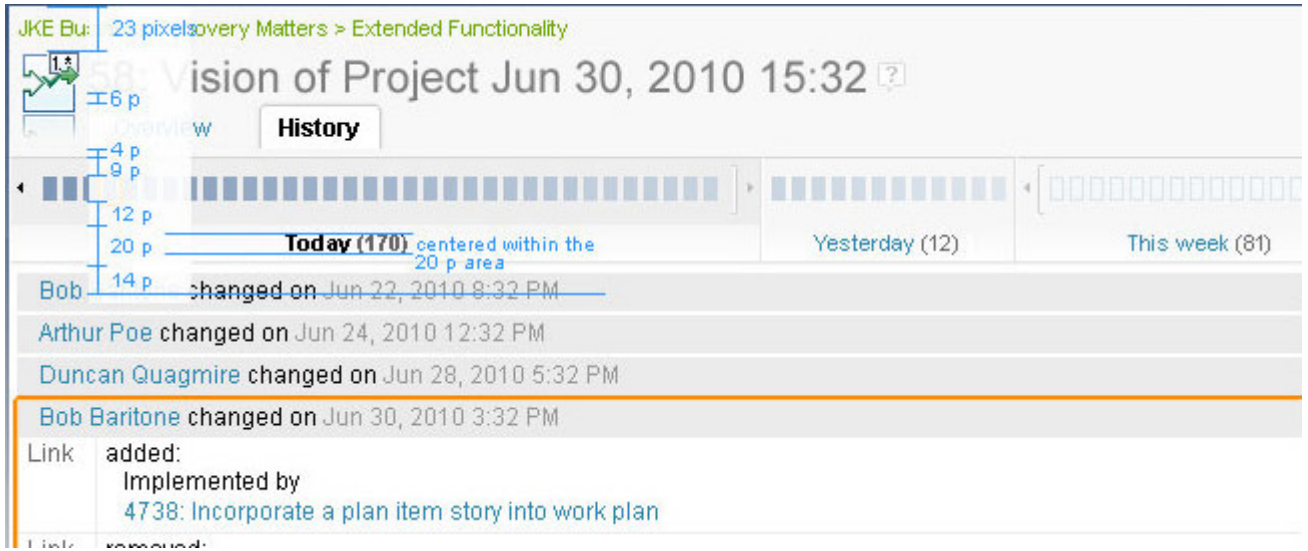
Add more detail to storyboards as needed (depending on audience and purpose)

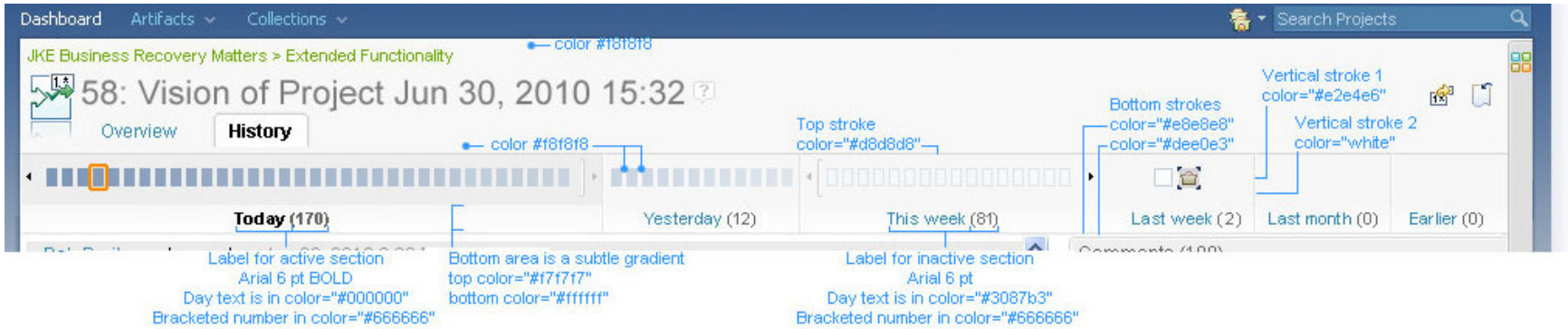






Example: detailed visual specifications





Example: detailed UI specifications document

01. Review Editor (Edit) - UI specs (3655) ▾ Composer2.0 | UX Designs | Feature: Review and Approval

<Enter Description>

visual designs. Detailed visual blueprints, where necessary, will be described separately.

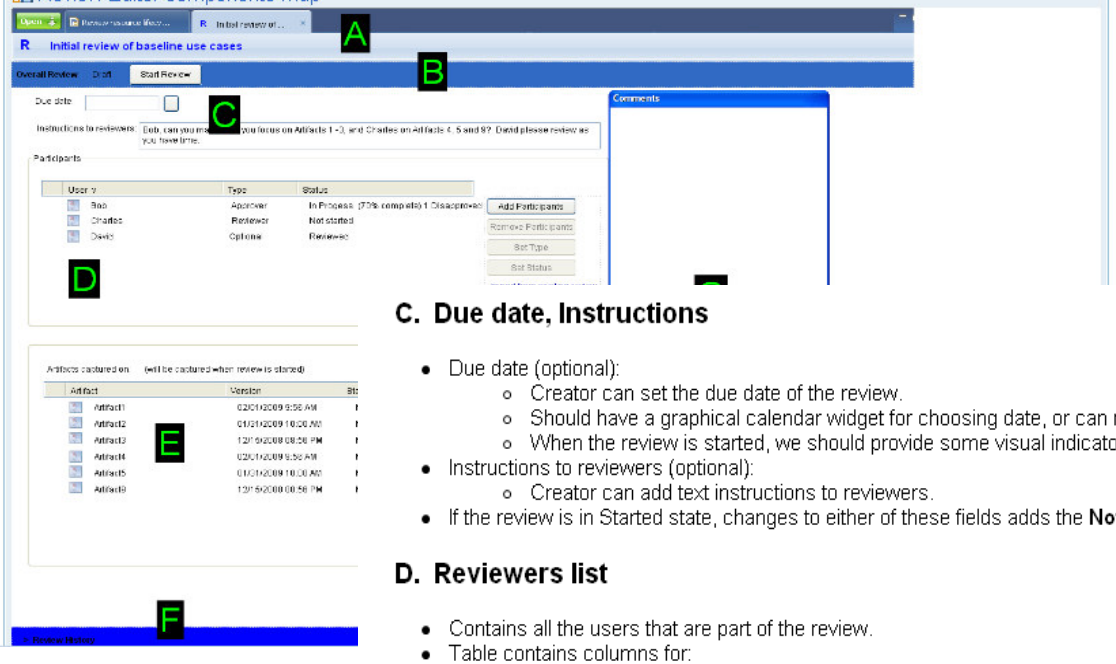
1. Review Editor - "Review State" section

This section of the review is for editing the review as well as showing the current state of the review. It is a different view of the same review as shown in the "Your Review" section (Section 2). It will primarily be used by the creator of the review.

All users can see this section, although only creators of the review or users with admin rights can edit it (for all others, the action buttons should be hidden). When the review is in draft state, this will be the top section (after the review is started, "Your Review" will be the top section.

The following document describes what a user can see and do based on their role and the state of the review: Views and actions for reviewers and non-reviewers

Review Editor components map



C. Due date, Instructions

- Due date (optional):
  - Creator can set the due date of the review.
  - Should have a graphical calendar widget for choosing date, or can manually type in date.
  - When the review is started, we should provide some visual indicator that it is past the due date.
- Instructions to reviewers (optional):
  - Creator can add text instructions to reviewers.
- If the review is in Started state, changes to either of these fields adds the **Notify Participants of Changes** button to the Status bar (section 1B).

D. Reviewers list

- Contains all the users that are part of the review.
- Table contains columns for:

Helping frame and resolve issues during development

- Issues and problems will arise during development
- Being able to visualize the problem as well as potential solutions can increase clarity and reduce the time to resolve the problems.
- Often used within development teams themselves, as well as discussions between development and user experience.

Illustrate and capture UI patterns and guidelines

- Use to illustrate “UI patterns” or “UI guidelines” for your product
  - Mock up examples, or capture and annotate existing UI
- Using common parts and sketches as building blocks for all UI mockups helps achieve consistency
- Sharing UI artifacts within team and across related product teams can also help achieve consistency, common patterns and synergy
- Can use to store screen captures of existing UI (to show patterns, identify inconsistencies, etc.)

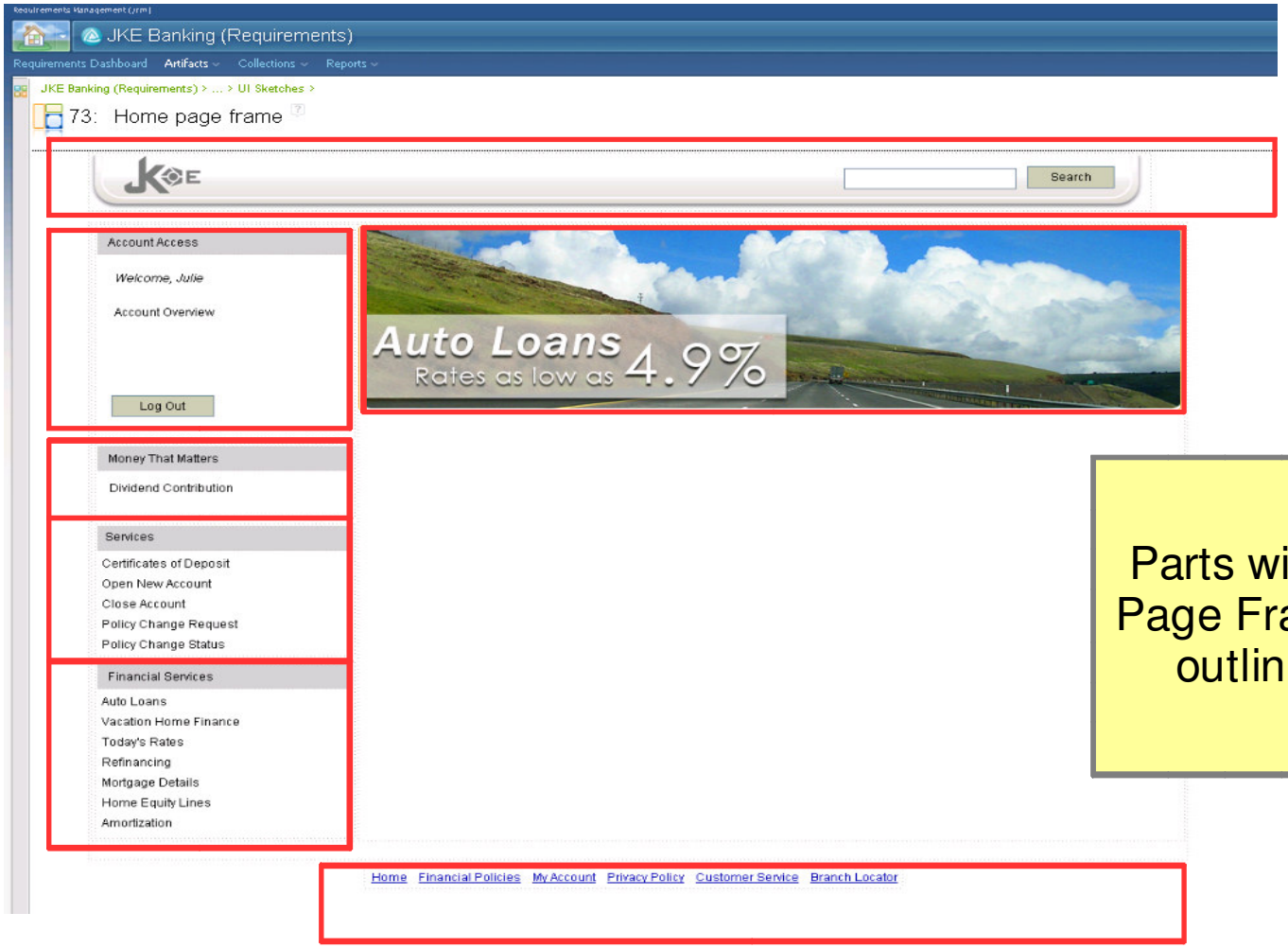
Tips and Techniques for Using Sketching Tools

Tips and techniques for using sketching tools

Make use of re-use (before you get too far)

- When you find yourself creating something that already exists
- Dialogs
- Common components, widgets, UI elements, etc.
- But don't let it interfere with your design thinking – you can extract parts and sketches later.
- Not just parts – sketches can inherit sketches
  - Good for designing alternatives
  - Can use as “templates” (different from actual artifact templates)
  - Storyboards are also good for variations on similar designs
  - Think twice before copying a sketch and breaking inheritance

Example: re-use in JKE Banking sample project

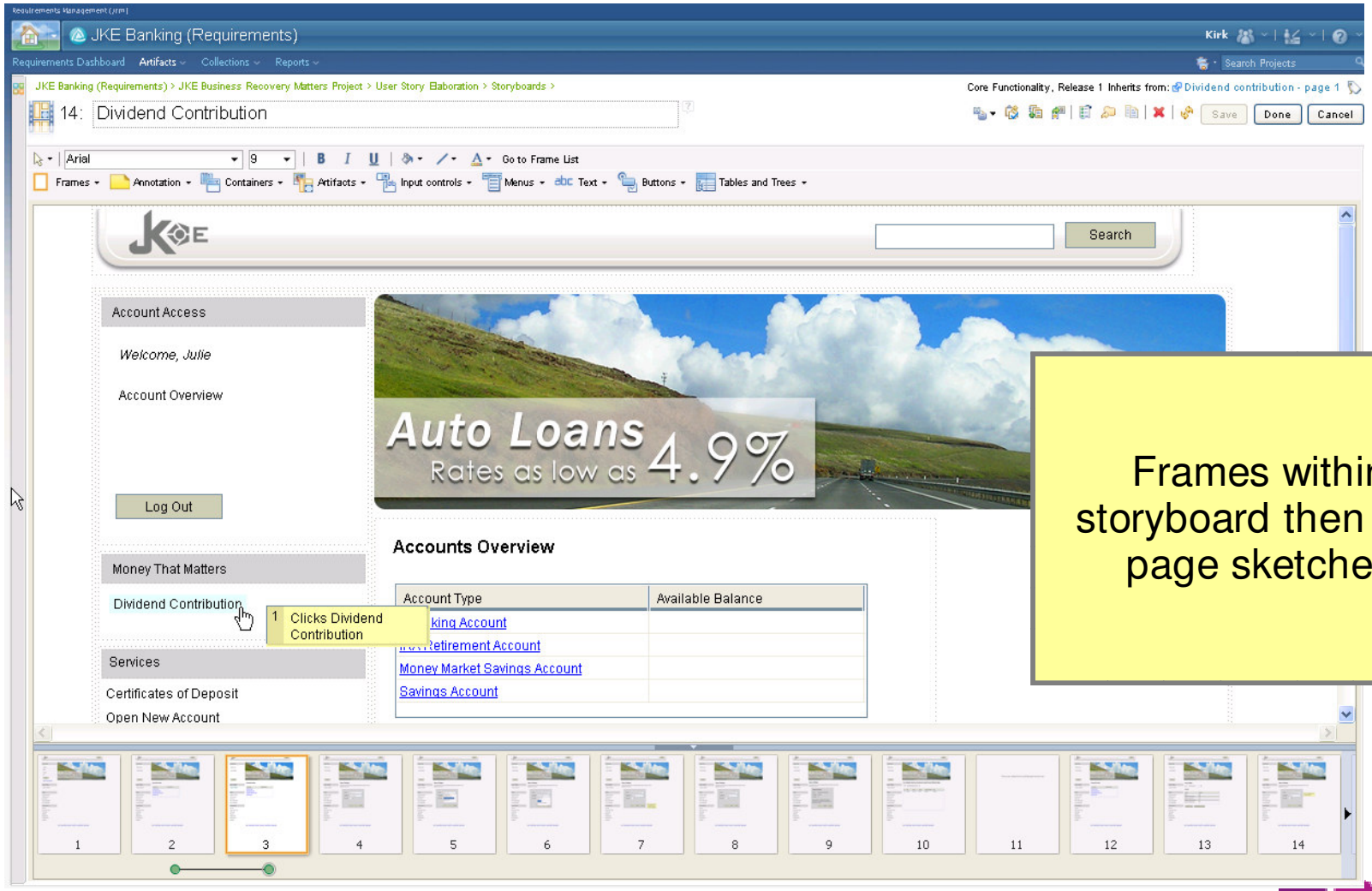


Parts within “Home Page Frame” sketch outlined in red

Example: re-use in JKE Banking sample project

"Accounts Overview (Home Page) sketch uses "Home Page Frame" sketch, and adds additional parts.

Example: re-use in JKE Banking sample project



Tips and techniques for using sketching tools

When exploring, create many design alternatives

- Ease of creating sketches should enable this
- Don't get caught up in details, especially early on; just make it “good enough to get the idea across”

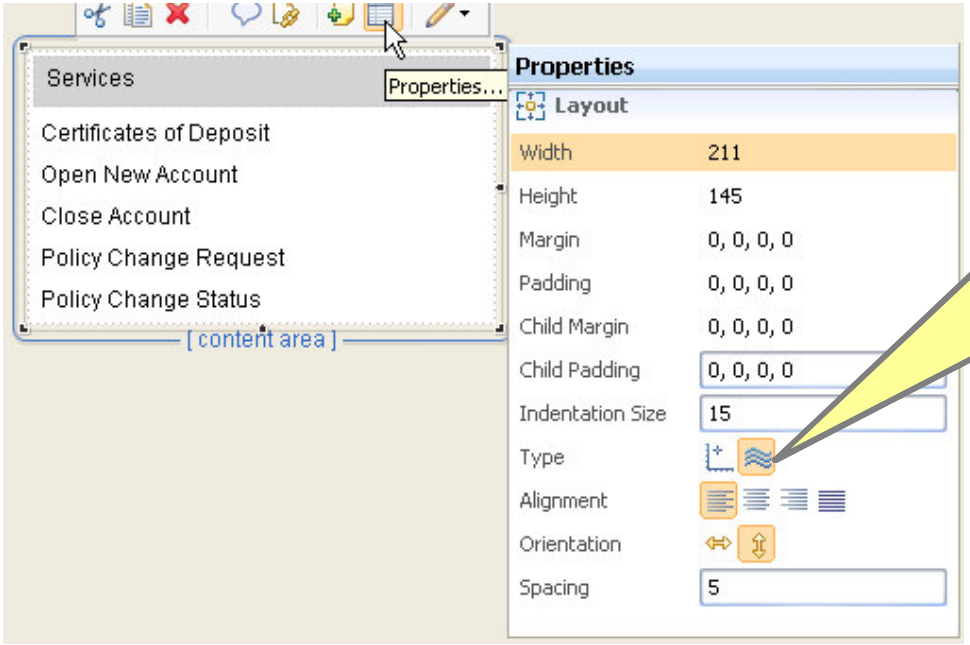
Think twice before deleting sketches/designs

- Good to always keep design alternatives around
- Be sure to keep them separate from official designs though

Tips and techniques for using sketching tools

Use Flow Panels for automatic layout

- Especially when you will be adding/removing things frequently from a container in a storyboard
- Can use nested flow panels
  - Set padding and margins to avoid too much white space

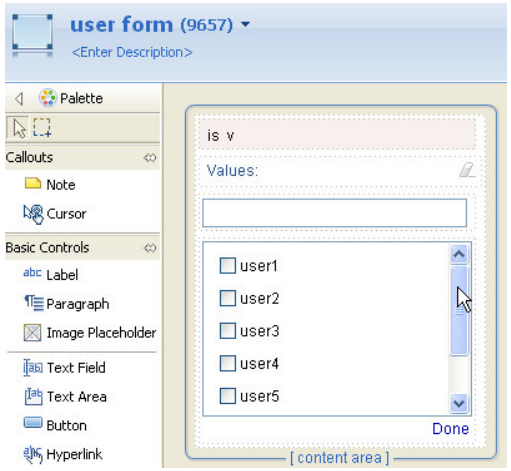


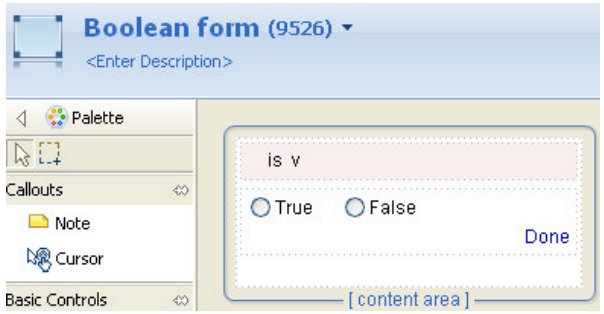
From Properties, click 2nd icon to set Type to "Flow". Can also set alignment, orientationand spacing.

Tips and techniques for using sketching tools

- Use parts to create your own “palette” of controls
- Improves efficiency in creating new sketches
- Improves consistency across UI designs and implementations

Examples





Tips and techniques for using sketching tools

"Share" sketching designs early

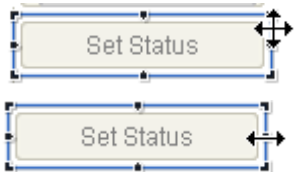
- Make it clear that it's in progress, but make sketching work visible early
- Make use of commenting and reviews

Add external visual artifacts to your repository

- If you create visual artifacts (e.g., high-fidelity UI design prototypes), add them to your Requirements Composer repository.
- If they are multiple images representing a work flow, paste the images into storyboard frames
  - Easier to navigate the flow than looking through a series of images
  - Can take advantage of commenting, reviews, linking, etc.
  - Can access via web

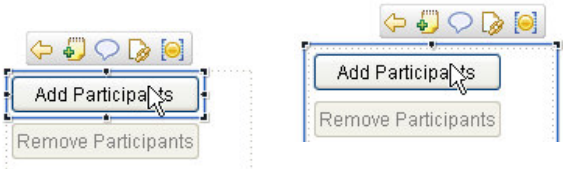
Tips and techniques for using sketching tools

Some lesser-known, but useful, techniques



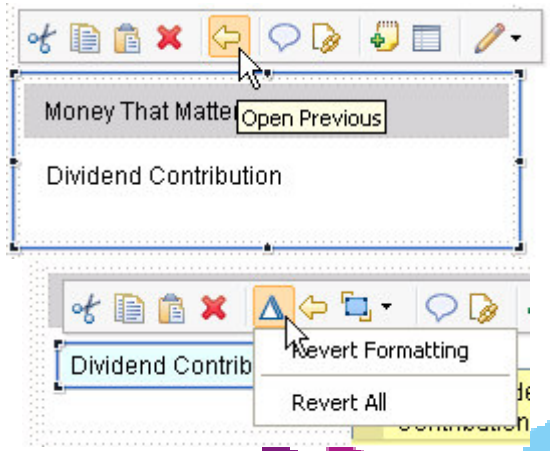
Arrow keys will move focus between UI elements

- To use arrow keys to "nudge" UI elements, press "." (period) first. Press Enter when done (or Esc to cancel move).
- To use arrow keys to resize UI elements, press "." multiple times until the cursor is on the side you want to move; then use arrow keys to resize UI element.



Alt + Up arrow to move focus to parent container, Alt + Down arrow to move focus within container

Open inherited sketches and parts using Previous button in contextual toolbar



Reset override values of inherited elements using Revert Changes button in contextual toolbar

Tips and techniques for using sketching tools

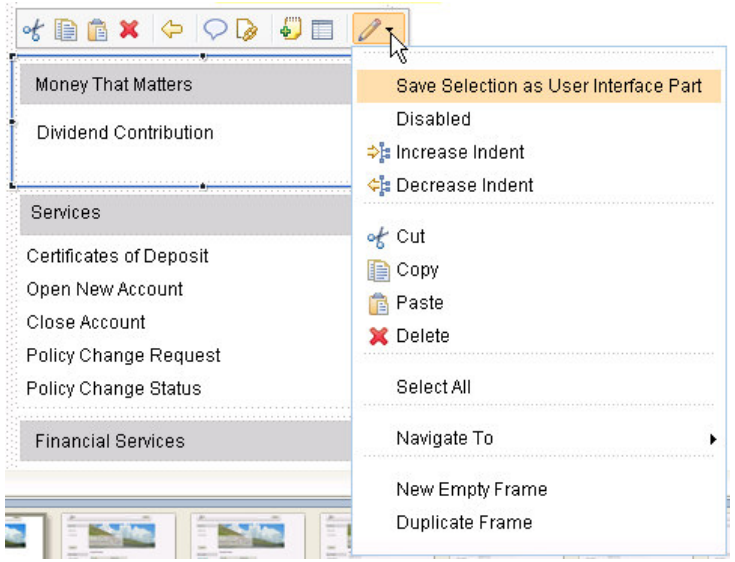
Some general changes in RRC Next that affect sketching editors

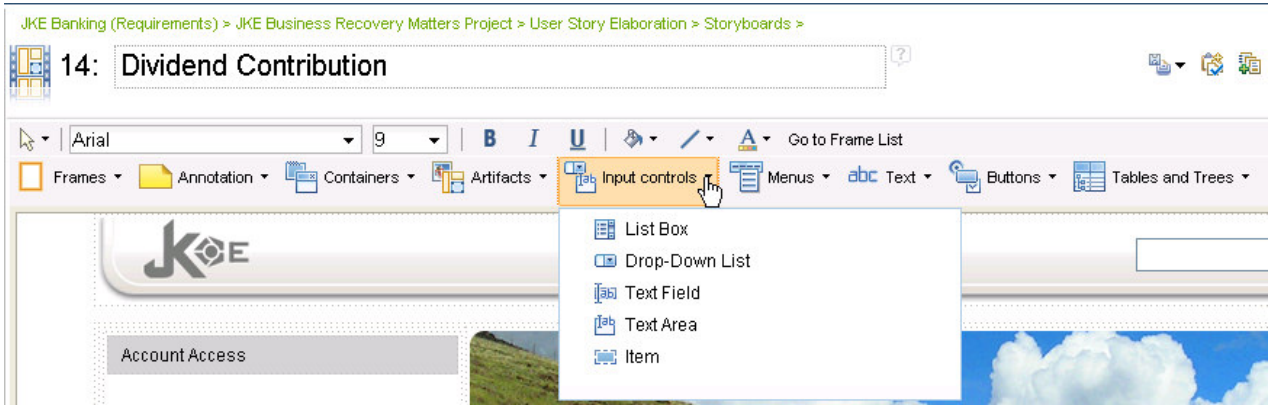
- Everything is web based.
- Web interaction vs tabbed IDE interaction
- Read mode vs edit mode
- Editor toolbar has moved to within editor
- Customizable type system

Tips and techniques for using sketching tools

Some changes to sketching editors in RRC Next.

- Palette
  - Palette on top instead of left side.
  - Grouping of elements in palette
  - Slightly different palette interaction
- Element interaction
  - No context menu
  - Expanded contextual toolbar
- Frame interaction slightly different





www.ibm/software/rational

www.ibm/software/rational