

Getting Started with Programs



This tutorial is intended to get ScrumWorks Pro users up and running with Programs, a feature available as of ScrumWorks Pro 4. Programs are intended to provide large and complex development groups a way to coordinate development efforts toward common feature and date goals. While ScrumWorks Pro readily supports multiple teams working against a single Product Backlog, modern development organizations often need a way to coordinate distinct but related product backlogs. The Program feature was designed to support the following complex development scenarios:

- Coordinated release of a product suite (e.g., desktop "office" suite)
- Complex products that are made up of distinct components (e.g., a mobile phone)
- Core components contribute to multiple end-products (e.g., video games with core game engine)

This tutorial will make use of an example domain to illustrate these concepts. The fictitious scenario is a game development company that creates a suite of card games that ship on CD-ROM. In this scenario, several distinct games like Bridge, Cribbage, Rummy, and Solitaire are all being developed simultaneously for inclusion in the "2.0" release of this CD-ROM suite. The release date for version 2.0 has been set as October 31, 2009.

For this Card Games scenario, each "game" has its own Product Backlog which is achieved by creating separate "Products" in ScrumWorks Pro. Each game also has one or more Scrum Teams working on the game. For example, Solitaire has two Teams working against the Product Backlog. In addition, the organization has a group of developers working on the core "game engine" underlying each of the games by providing common components like randomization and animation support. Therefore, a "Game Engine" Product has been set up in ScrumWorks Pro.

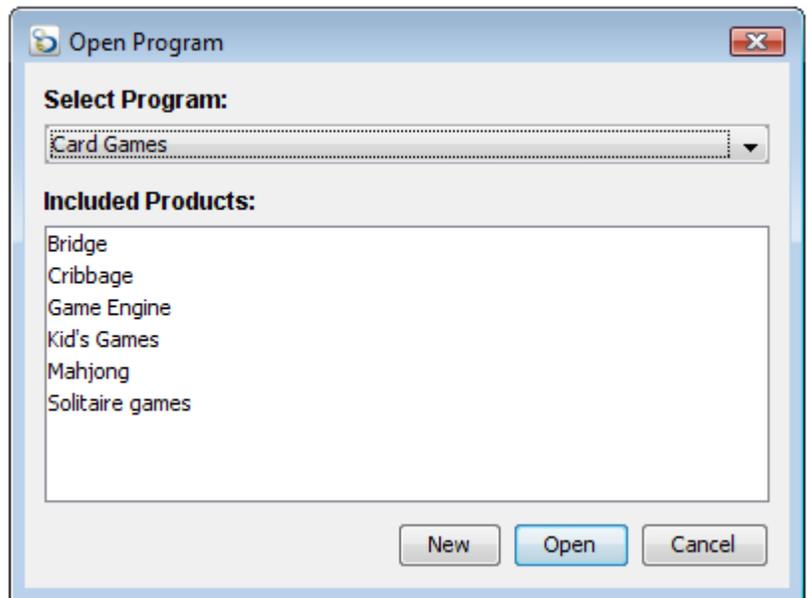


Figure 1: The Open Program Dialog

Creating Programs

The release planners in this example need a way to bind and coordinate these Products. They must be able to set date and feature-level goals that span all of these Products. To do this, they create a "Program" in ScrumWorks Pro. Programs represent a collection of related Products, so in this example each individual Product is included in the newly created "Card Games" Program.

Creating a Program can be accomplished by a user flagged as a Global Administrator by Select File > New Program.

Enter a Title for your Program, this title must be unique amongst all Programs within your installation of ScrumWorks Pro.

Optionally, you may fill in the Description field. Much like filling out the Description field in a PBI, here you can outline any information anyone working with this Program may need, goals or requirements for example.

Products may be included in the Program by selecting the appropriate checkbox. Products may be included in multiple Programs; a column to the right of each Product name lists the Programs in which a Product has already been included. When a Product is included in a Program, that Program's Releases, Epics, and Themes become available for use by the Product.

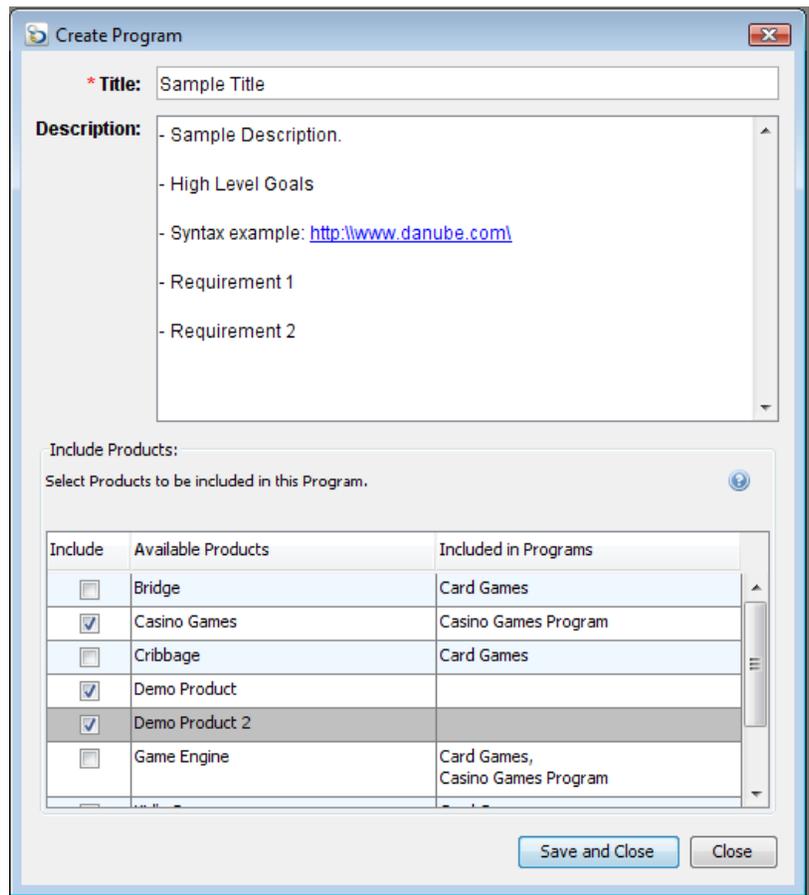


Figure 2: The Create Program Window

Creating Program Themes

Now that a Program has been established, the next step is to create a common organization mechanism between all Products involved in our Program. The concept of Themes has been extended to Programs in ScrumWorks Pro 4. Themes are essentially keyword "tags" that can be applied to backlog items to provide flexible organization capabilities.

In our example, we need a common set of Themes that exist in all Products. This is where Program Themes come into play.

Program Themes are defined at the Program level and are then available for use by Products involved in the Program. This provides a way to group work items in the various Product Backlogs that share some commonality.

To add, edit, or delete a Program Theme or Themes, you must have first opened your Program. Then you can select the Themes menu > Edit Program Theme. Figure 3 illustrates this.

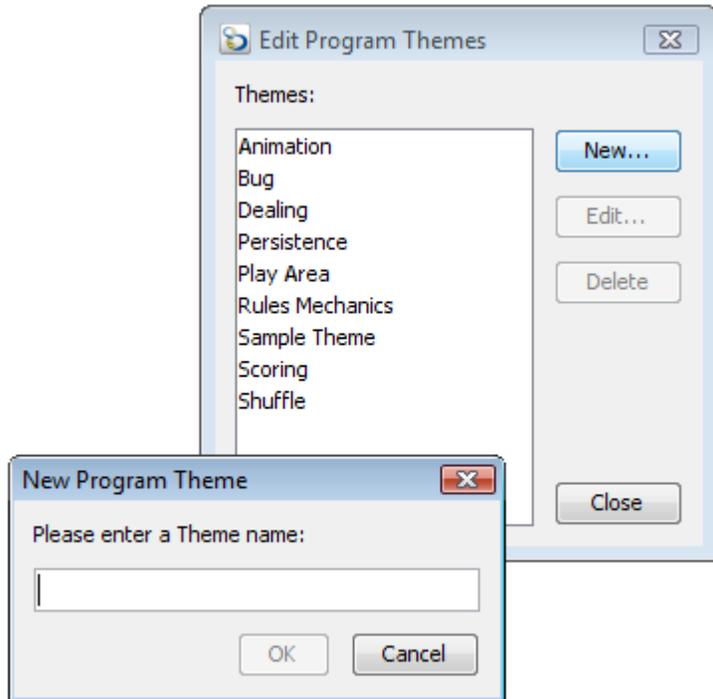


Figure 3: Adding a Program Theme

Program Releases

With Program Themes intact, the next step is to define milestone goals for the Program. Remember that the release planners want a production release of the Card Games suite on October 31, 2009. Program Releases are used to accomplish this in ScrumWorks Pro.

Program Releases are similar to a regular Product Release but are distinct in that they span all Products involved in the Program. Program Releases may have start and end dates designated, signifying the boundaries of the release cycle. In this case, the release planners enter October 31, 2009 in the end date, signifying a date oriented milestone that spans across the Program.

Once established, Program Releases appear in all Products associated with the Program. Program Releases are black rows in the Product Backlog to distinguish them from regular Product Releases. Product Owners can now use the newly created Program Release just as any other release: add backlog items, use it in reports, etc.

The newly created release also appears in the Program's Release Planner view. The release can be edited and monitored from this view as discussed below.

To create a Program Release, select File > New Program release. This will open the editor as seen in Figure 5.

Here you can name the Release. Remember that this release is going to be seen in every product attached to this Program.

Next you can fill out the description field. This is a good place to put high-level goals, or any information anyone working with the release may need to know. Again this will be seen across all constituent products. At this point, you can populate the date range for this release. These dates will be used in reporting to track progress.

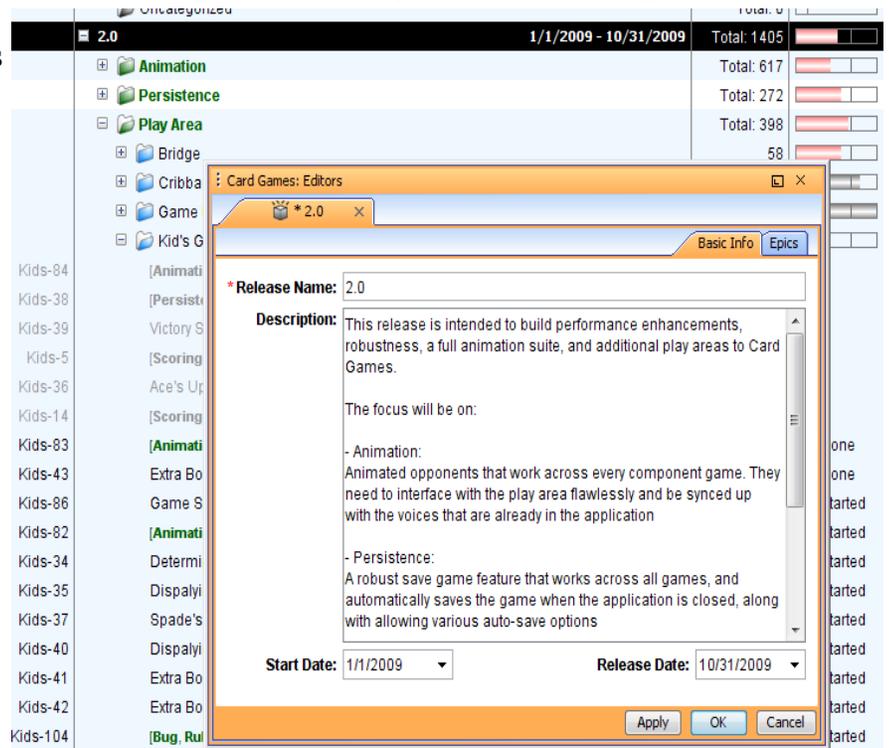


Figure 5: Program Release and Editor

Program Epics

With a date milestone established at the Program level, and all constituent Products aware of the milestone, the next step for the example Program is to establish sub-goals for the 2.0 release milestone. The release planners have four main areas they wish to target for development during the 2.0 milestone: Animation, Persistence, Play Area, and Shuffling. They now need some way to formalize these sub-goals and define the scope of each for purposes of the milestone.

In ScrumWorks Pro, this is accomplished using "Epics." Epics are formal goals that can be set up for each release in the Program. Epics are therefore created from the Release editor. Furthermore, Epics are based on Program Themes. In this way, a formal goal can be set at the Program level using Epics. Because Epics are based on Themes, backlog items in each Product that already use these themes are automatically associated with the newly created Epic. In this scenario, for example, any backlog item with the Theme "Animation" will be grouped into a newly created Epic "Animation."

In the Program Release Planner, the Epics associated with any Release appear as green folders subordinate to the Release in question. Any Products that intend to work on that Epic appear as Blue folders subordinate to the Epics of the Release.

Key	Backlog Items by Release/Epic	Backlog Effort	Status
2.0 1/1/2009 - 10/31/2009		Total: 1405	
+ Animation		Total: 617	
+ Persistence		Total: 272	
+ Play Area		Total: 398	
+ Bridge		58	
+ Cribbage		112	
+ Game Engine		105	
+ Kid's Games		51	
Kids-84	[Animation] 4 decks	8	Done
Kids-38	[Persistence] Play table background ...	2	Done
Kids-39	Victory Scroe set to 150	4	Done
Kids-5	[Scoring] Ace is Always Low	2	Done
Kids-36	Ace's Up knock Only on Gin	1	Done
Kids-14	[Scoring] Knocking a Run	2	Done
Kids-83	[Animation] 3 decks	8	Not Done
Kids-13	Extra Boxes - Aces	8	Not Done

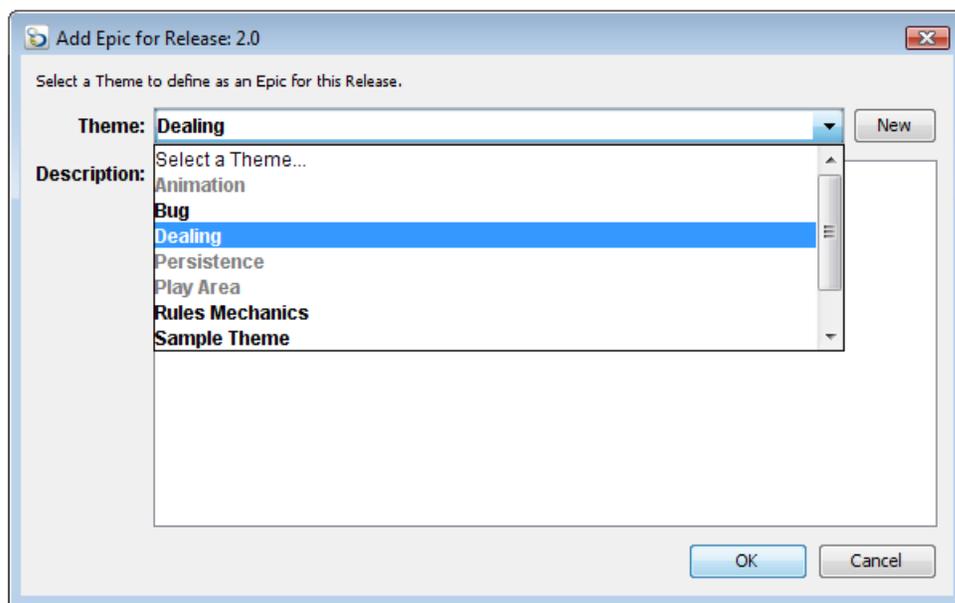
Figure 6: Program Release Planner

Epic Name	Description
Animation	Done: All animations are fully tested, and completely integrated with all play areas. This includes all randomization options for each
Persistence	Done: - All save and auto save functionality is in place
Play Area	Done: - Western play areas are complete and integrated properly with
Shuffle	Done: - Both the Fisher-Yates and Sattolo's algorithm are in place and

Figure 7: Epics Tab

To create Epics, right-click on the release you wish to add the Epics to, then select Edit. This will open up the Release editor. Here you will want to select the Epic tab. See Figure 7.

This will display any Epics currently associated with this Program Release. In Figure 7, we already have four Epics with descriptions. To add another, simply select the Add button.



When adding an Epic, we can select any pre-existing Program Theme we previously created, or choose to create a new one using the New option. This is illustrated in figure 8.

At this point, you can fill out the Description field with the goals, and any information that you may need to associate with this Epic.

It's important to understand that Epics are limited in scope. Naturally, this sample Program could

Figure 8: Add Epic Window work on the concept of "Animation" nearly infinitely. But because Epics are tied to a particular Release, they are bound in scope to the end date set for that Release.

Epics can also be confined by providing a description in which the "Done" criteria is made clear. The Epic is therefore bound by dates and some written expectation for what "Done" means for purposes of the release in question.

In our example, "Animation" is established as an Epic for purposes of Release 2.0. Any backlog item in any Product that is themed with "Animation" and, at the same time, has been placed inside the "2.0" release backlog will automatically be associated with the "Animation" Epic. In this way, the concept of "Animation" has continuity and can be revisited in subsequent Releases. Say a 3.0 Release also elects to focus on "Animation," the scope of that Epic will be different from the 2.0 Release "Animation" Epic.

Moving backlog items with the "Animation" Theme from Release 2.0 to Release 3.0 will therefore move those items from the 2.0 Epic to the 3.0 Epic. In this way, concepts can be revisited repeatedly, as enhancements are introduced milestone by milestone.

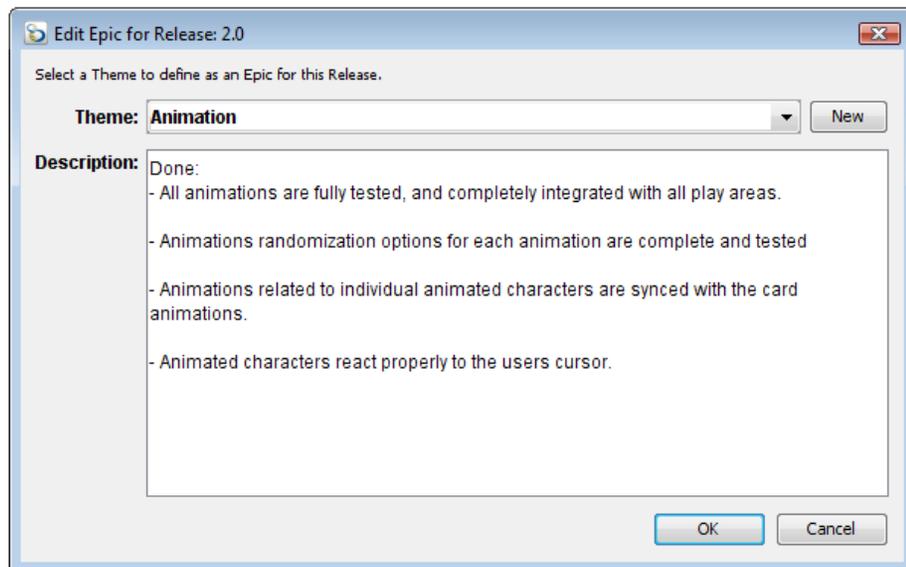


Figure 9: Epic Description

Adding Program Epics to Products

Now that we have Program Epics defined within the Program Window, individual Products can elect to contribute toward some or all Program Epics. This is done by editing the Program Release from *within a Product Window* and adding the selected Program Epics to the Release from the context of that Product. Within a larger Program, individual Products may wish to contribute only to a subset of the Epics defined for the Program. This extra step is required to ensure that Products only contribute to relevant Epics.

In our example, let's say that the Solitaire Product needs to contribute to the "Shuffle" Program Epic but the Cribbage Product does not. In this case, the Solitaire Product would add the "Shuffle" Program Epic to the Program Release from within the Solitaire Product, but the Cribbage Product would not do so and instead elect to contribute toward other Epics that are more relevant to Cribbage.

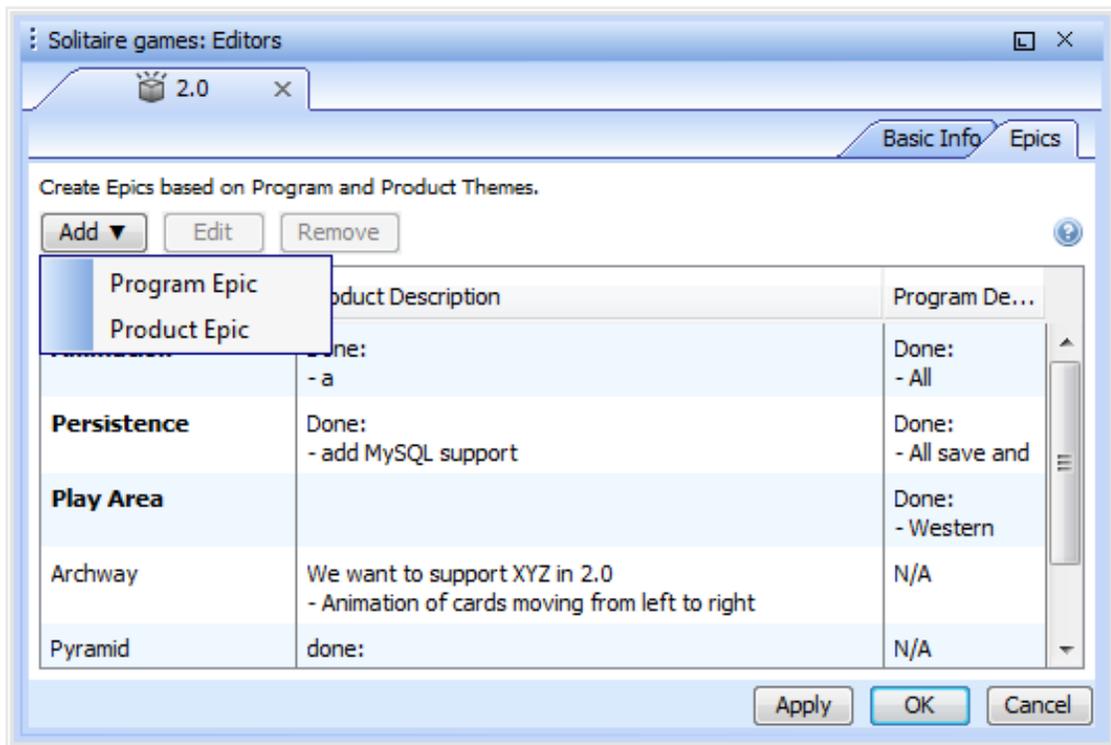


Figure 10: Product Epics

Contribute to a Program Epic from Within a Product

A Product can contribute to a Program Release's Epics by editing the Program Release in question and adding the relevant Program Epics. The following steps illustrate this process:

1. Double-click a Release row in either the Product Backlog frame or the Release Planner frame.
2. In the resulting Edit Release frame, select the Epics tab.
3. Select the "Add" button and a drop-down option will appear offering either "Program Epic" or "Product Epic". Since in this case the goal is to contribute to a Program Epic, select "Program Epic".
4. Optionally enter a Product level description. This field can be used to describe the specific Product's involvement in the Program Epic, and any "definition of done" that may apply.
5. Click "OK" in the Add Epic dialog, and then click "OK" or "Apply" in the Release editor dialog.

The selected Epic will appear in the Product level Release Planner. In the Program Window, a blue folder icon will appear under the Program Epic in question, signifying that the Product has elected to contribute to the Program Epic.

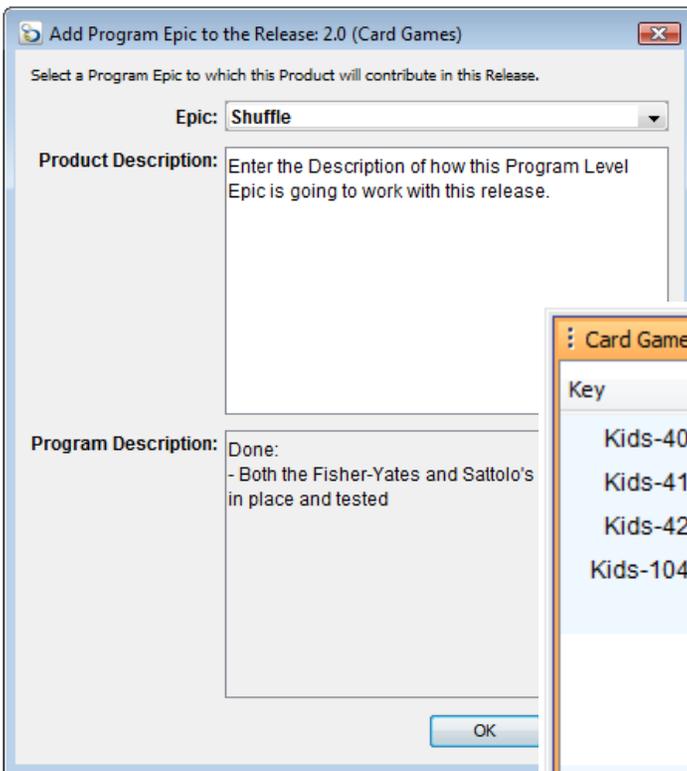


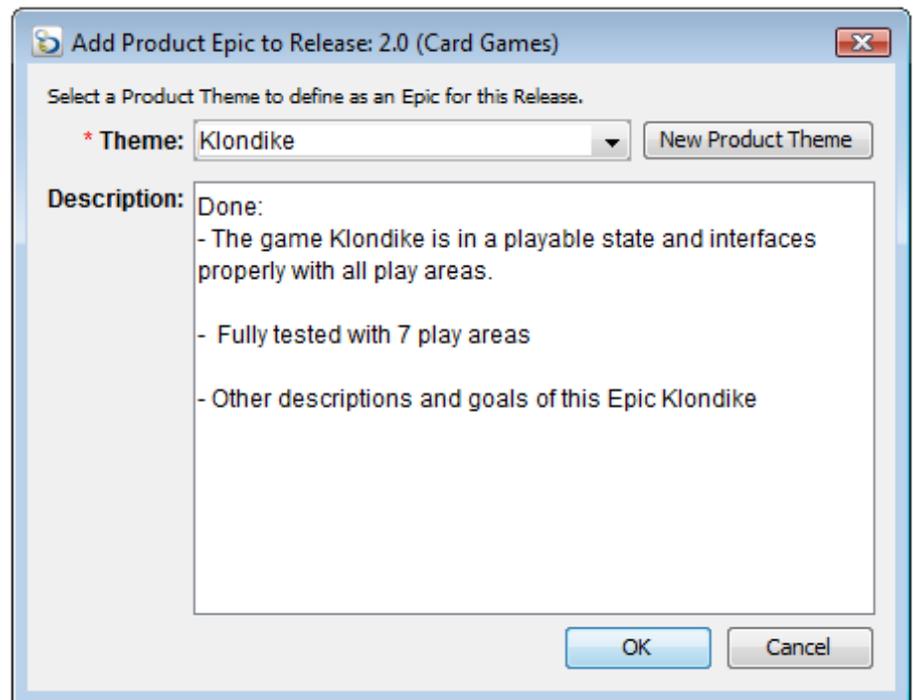
Figure 11: Add Program Epic

Key	Backlog Items by Release/Epic	Backlog Effort	Status
Kids-40	Dispalying Extra Boxes	2	Not Started
Kids-41	Extra Box Point Values	1	Not Started
Kids-42	Extra Boxes - Spade	4	Not Started
Kids-104	[Bug, Rules Mechanics]	1	Not Started
	+ Solitaire games	54	
	[-] Shuffle	Total: 119	
	+ Game Engine	119	
	+ Solitaire games	0	
	[-] Uncategorized	Total: 158	
	+ Cribbage	Total: 8	
	+ Game Engine	Total: 33	
	+ Kid's Games	Total: 10	
	+ Solitaire games	Total: 107	

Figure 12: Release Planner

Adding Product Epics to Program Releases

There may be cases in which a Product contributing to a Program Release and Epics may also wish to add a Product level Epic to a Program Release. Using our example domain, the Solitaire Product has work to do related "Klondike" and "Row Stacks", concepts that are particular to the Solitaire game and irrelevant to other games. This work needs to be done as part of the 2.0 Program Release, so the Solitaire game can add Product level Epics to the Program Release, thereby creating localized sub-goals specific to the Solitaire Product.



The following steps illustrate this process:

1. Double-click a Release row in either the Product Backlog frame or the Release Planner frame.
2. In the resulting Edit Release frame, select the Epics tab.
3. Select the "Add" button and a drop-down option will appear offering either "Program Epic" or "Product Epic". Since in this case the goal is to add a local, Product Epic, choose "Product Epic".
4. Optionally enter a description.
5. Click "OK" in the Add Epic dialog, and then click "OK" or "Apply" in the Release editor dialog.

Figure 13: Add Product Epic

Tracking Program Release and Epic Progress

Once Program Releases and Epics are established and the individual Products are working toward these goals, the program planner will need some way to track progress through to completion. The Release Planner view has built-in progress roll-ups to provide at-a-glance summaries of progress. The right-most column provides progress indicator bars that show the percentage of work items done under that particular level. There is an overall release progress indicator, a progress indicator per Epic, and also per Product as it contributes to individual Epics. A vertical line between the start and end dates indicates "Today" and provides a guide for determining whether a particular Release, Epic, or Product is behind schedule. The bar is silver if the row in question is ahead of schedule, and red if the row is behind schedule.

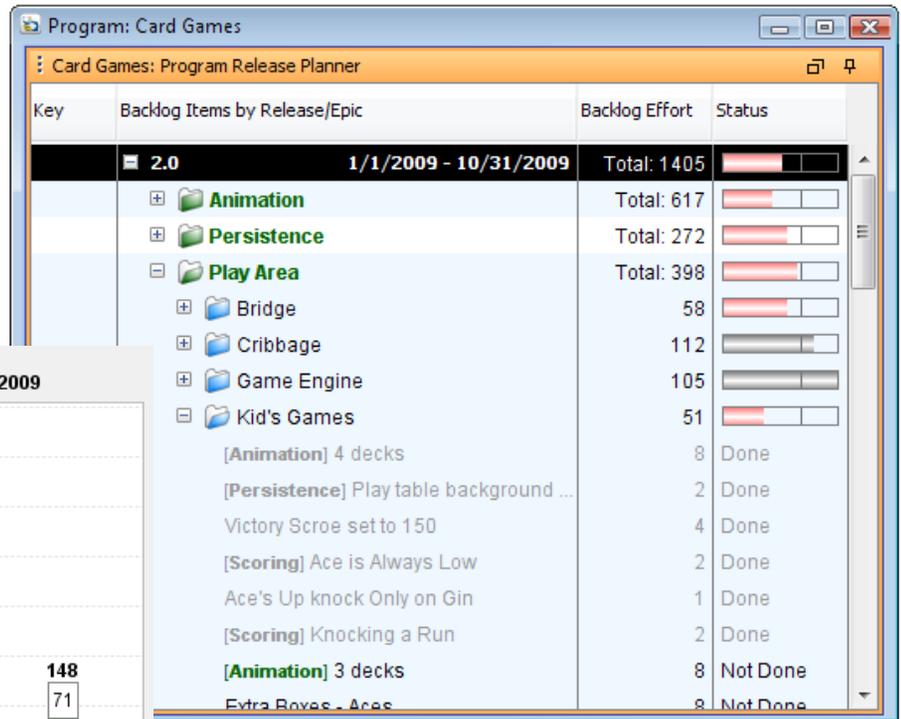


Figure 12: The Release Planner

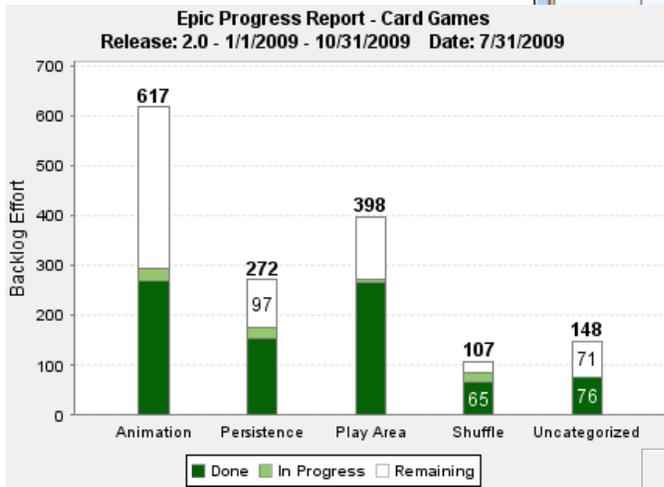


Figure 13: Epic Progress Report

The Epic Progress Report provides a summary of progress across all Epics in one graphical report. Please see the documentation for more on this report.

The Program Release Burndown Chart provides an indication for the overall release forecast by overlaying the release burndown charts of each contributing Product. Please see the documentation for more on this report.

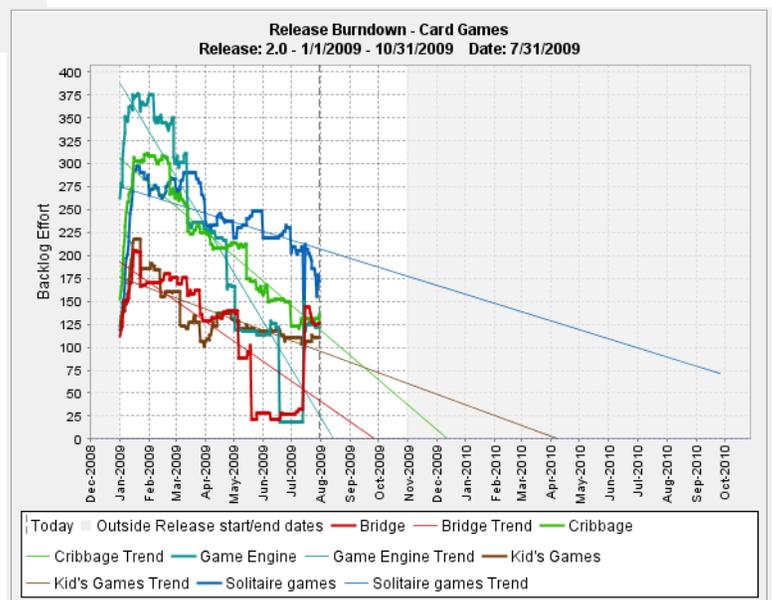


Figure 14: Program Release Burndown

Shared Components

To effectively model the complex development environments, ScrumWorks Pro supports the notion of a "shared component" or "core module." Earlier in this tutorial, the Game Engine product was introduced as providing foundational support to all games in the Card Game suite. But what if this Games Engine's core components were also used by another product suite entirely, say a "Casino Games" suite?

Key	Uncommitted Backlog Items/Tasks	Backlog Effort
Casino release 1.0 (Casino Games Program)		Total: 31
Eng-...	<input type="checkbox"/> [Animation, Animatio...] Computer Player Joy animation	3
Eng-...	<input type="checkbox"/> [Animation, Animatio...] Computer Player Emotion - Hatred	8
Eng-...	<input type="checkbox"/> [Animation, Animatio...] Computer Player Emotion - Happiness	9
Eng-...	<input type="checkbox"/> [Animation, Animation] Computer Player Emotion - Dissapointment	8
Eng-...	<input type="checkbox"/> [Voice, Animation, D...] Edward's Useless Comments	3
2.0 (Card Games)		Total: 93
Eng-...	<input type="checkbox"/> [AI, Dealing, Shuffle] Set AI Level - Edward - Easy	8
Eng-...	<input type="checkbox"/> [Voice, Dealing] Edward Annoyed	2
Eng-...	<input type="checkbox"/> [Voice, Bug, Dealing...] Voice - Abraham's Introduction	1
Eng-...	<input type="checkbox"/> [Animation] Computer Player Special Animation Frequency Setting	4
Eng-...	<input type="checkbox"/> k3b recognizes CD-RW only as a CD-Rom	8
Eng-...	<input type="checkbox"/> Exported localization file content should be sorted	1
Eng-...	<input type="checkbox"/> qtcurve 32-bit is missing on an x86-64 installation, but needed for Firefox, RealPlayer	8
Eng-...	<input type="checkbox"/> [Voice, Dealing, Shu...] Voice - Annoyed Noise Triggers	8
Eng-...	<input type="checkbox"/> [Animation] Computer player Ear animations	8
Eng-...	<input type="checkbox"/> [Animation] Computer player Eyebrow animations	1
Eng-...	<input type="checkbox"/> [Animation] Special Animation - Holding a 6 shooter	8
Eng-...	<input type="checkbox"/> [Animation] Special Animation - Holding a Ray Gun	2
Eng-...	<input type="checkbox"/> [Animation] Special Animation - Temper Tantrum	8

Figure 15: Game Engine Product

In ScrumWorks Pro, this would be accomplished by adding the Game Engine product to a second Program (Casino Games). Releases, Themes, and Epics from the Casino Games product would appear for use in the Game Engine product backlog. In this way, the Game Engine team(s) could work on backlog items related to both Card Games and Casino Games Programs.

This concludes the ScrumWorks Pro Programs tutorial. Please contact support with any questions or clarifications.