



JavaOneSM

Pro JavaFX Platform Building Enterprise Applications with JavaFX

Stephen Chin

GXS

steve@widgetfx.org

tweet: @steveonjava

Jim Weaver

JavaFXpert.com

jim.weaver@javafxpert.com

tweet: @javafxpert

What do you want to discuss?

Here's our [flexible] plan:

- > Examine Rally's Stratus application being developed in JavaFX 1.3.1 by the presenters
- > JavaFX 2.0 Code Examples in Java and Scala
- > Hit the high points of the newly published JavaFX roadmap at <http://javafx.com/roadmap>
- > Announce the RIA Exemplar Challenge winner
- > Questions encouraged at anytime!

Case Study: Rally's Stratus Application

- > Being developed in JavaFX using an Agile process and the Rally Agile tool
- > Accesses REST-based Rally API
- > Gives users an overall view of work in process in various Kanban stages

The greater the success experienced in R&D, the more disruption that it creates for the organization as a whole.

Feature selling becomes impossible (Sales Enablement)



Launch Cycle Time > Dev Cycle Time

...in the weeds ...

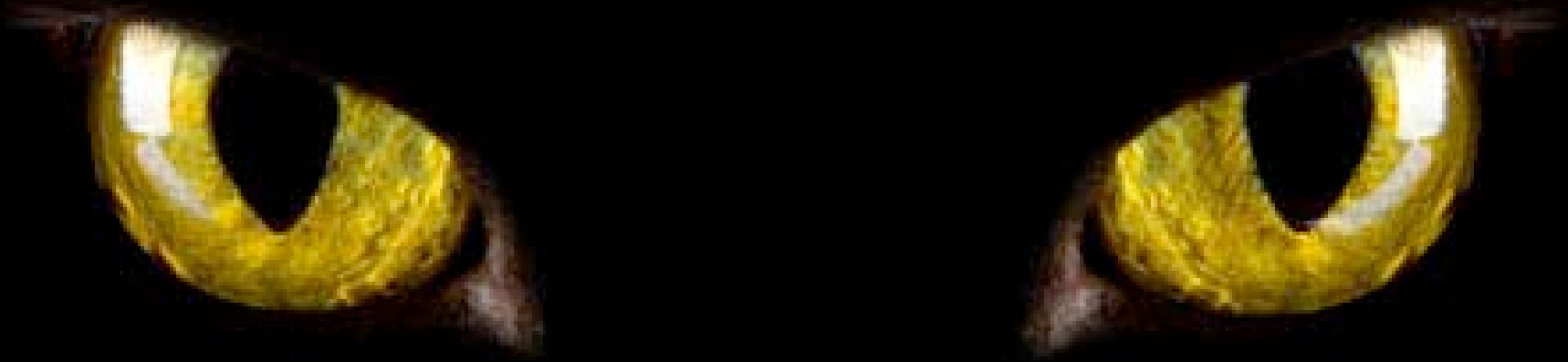


Melting Change Managers: 50 changes once a month to 900 changes constantly

Supported Release proliferation

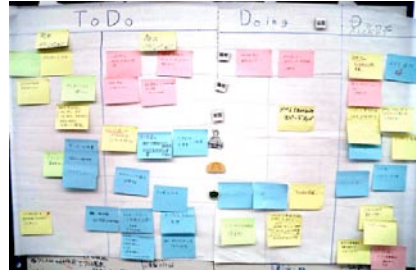
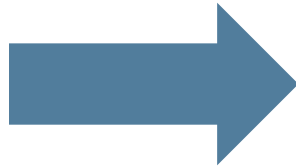


Innovator's Dilemma

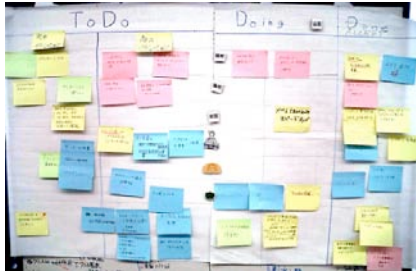
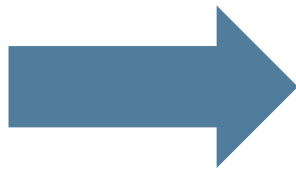


“Eat your spinach or the Scrum will get you.”

However, by applying/extending those same lean/kanban principles more broadly in the organization, these risks can be avoided, and organizational value increased.



Source: InfoQ



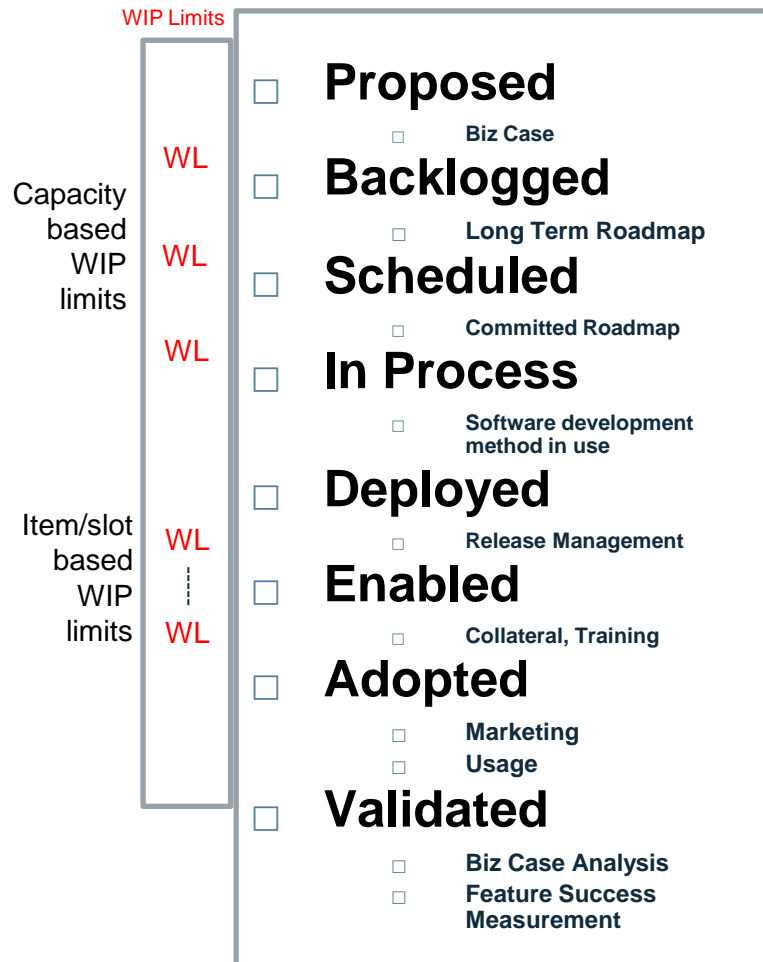
Source: InfoQ

Three Atypical, but Critical Practices in the Product Portfolio Kanban:

- Stakeholder Based Investment Themes and Business Case Management (organizational value)
- Upstream and Downstream WIP Limits
- Dynamic Allocations

We manage each business case through a Kanban which extends upstream and downstream from traditional development

Kanban Stages

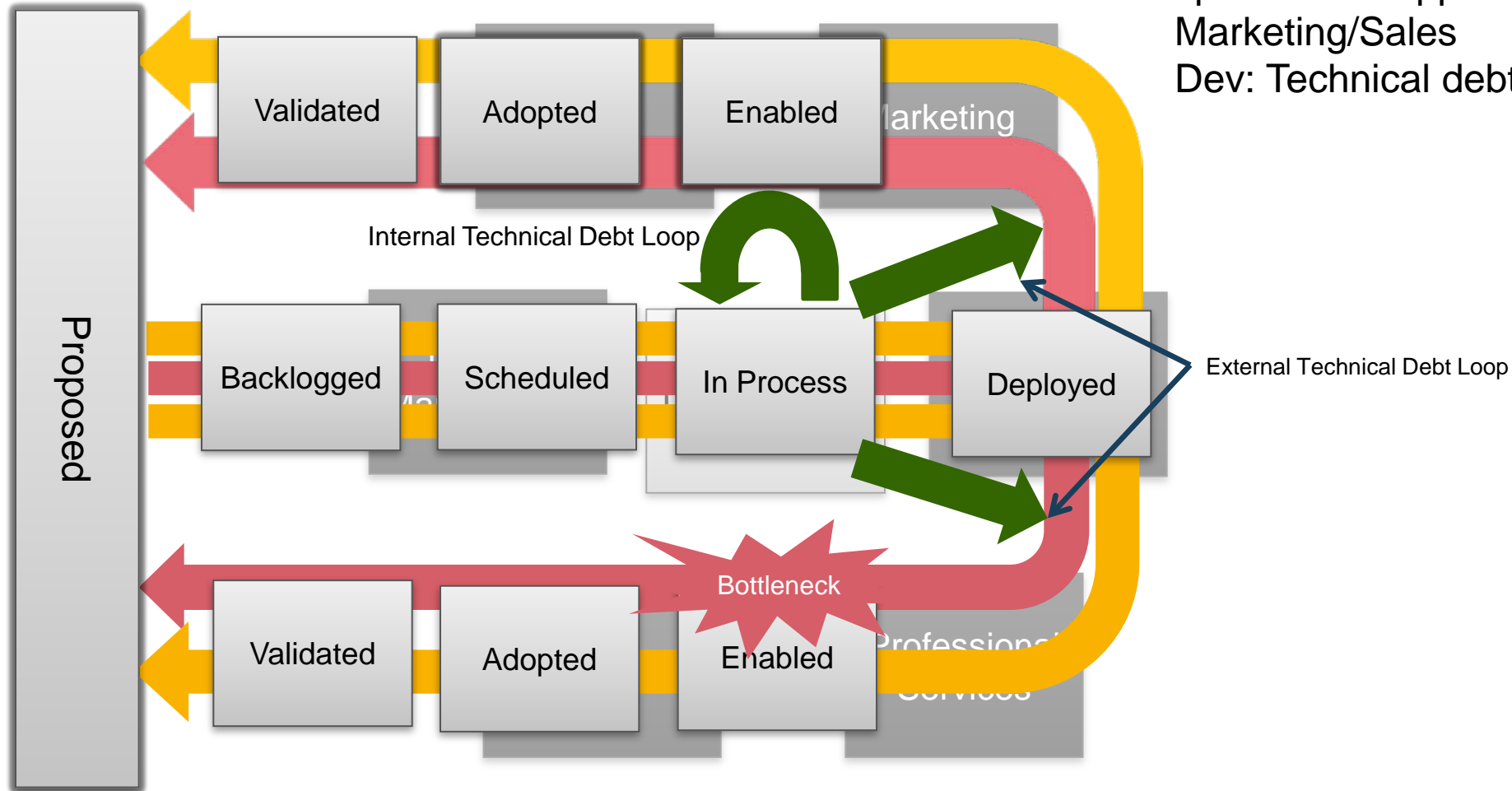


Benefits

- Alleviate the “Agile Death Ray” Effect and Achieve Flow
- Expose and Route Around Political Roadblocks and Priority Alignment
- Focus Organizational Value Return

The Three Loops of Software Governance

Operations/Support
Marketing/Sales
Dev: Technical debt



Stratus Kanban Page

Stratus PREVIEW

Integrations Online Store

Kanban Roadmap Plan Analyze

Investment Allocation: <All> Release: <All> Team: <All> Configure Refresh

Propose			Backlog			Schedule			Develop			Deploy			Enable			Adopt			Validate			Archive								
Feature	Est		Feature	Est		Feature	Est		Feature	Est		Feature	Est		Feature	Est		Feature	Est		Feature	Est		Feature	Est		Feature	Est				
Recent Purchases View	2		Persistent Shopping Cart	2		Combine Orders	3		Purchase Your Items	2		API for Basic Shipping	2																			
			API for Multi-Site Shipping	2		Order Modification	3																									

Count	Total	Limit	Count	Total	Limit	Count	Total	Limit	Count	Total	Limit	Count	Total	Limit	Count	Total	Limit	Count	Total	Limit	Count	Total	Limit	Count	Total	Limit	Count	Total	Limit			
1	2	None	2	4	None	2	6	None	1	2	None	1	2	None	0	0	None	0	0	None	0	0	None	0	0	None	0	0	None	0	0	None

©2003-2010 Rally Software Development Corp. Version 0.9

Stratus Roadmap Page

Stratus PREVIEW

Integrations Online Store

Kanban Roadmap Plan Analyze

Investment Allocation: <All> Team: <All> Configure Refresh

Backlog

	Feature	Est
1 Epic: Shopping	Persistent Shopping Cart	2
2 Shipping API	API for Multi-Site Shipping	2

2010 Q3

	Feature	Est

2010 Q4

	Feature	Est
1 Epic: Order Management	Combine Orders	3
2 Epic: Order Management	Order Modification	3

2011 Q1

	Feature	Est

2011 Q2

	Feature	Est

Count	Total	Limit
2	4	None

Count	Total	Limit
0	0	None

Count	Total	Limit
2	6	None

Count	Total	Limit
0	0	None

Count	Total	Limit
0	0	None

©2003-2010 Rally Software Development Corp. Version 0.9

Stratus Plan Page

Integrations
Stratus PREVIEW

Online Store
Kanban Roadmap **Plan** Analyze

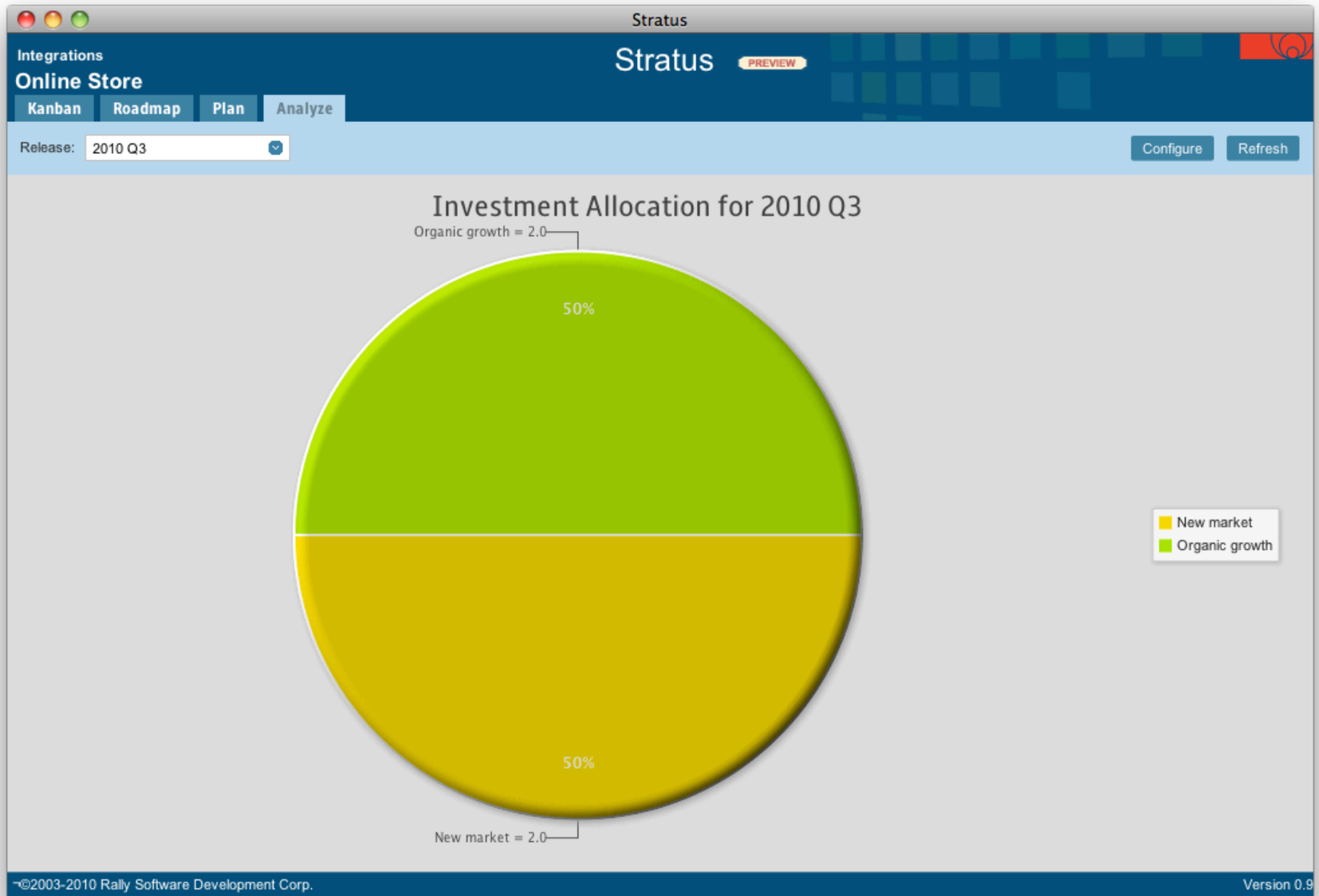
Investment Allocation: <All>
Release: 2010 Q3
Team: <None>
Configure Refresh

#	ID	Feature	Team	Est
1	S6	Shipping API	API for Basic Shipping	2

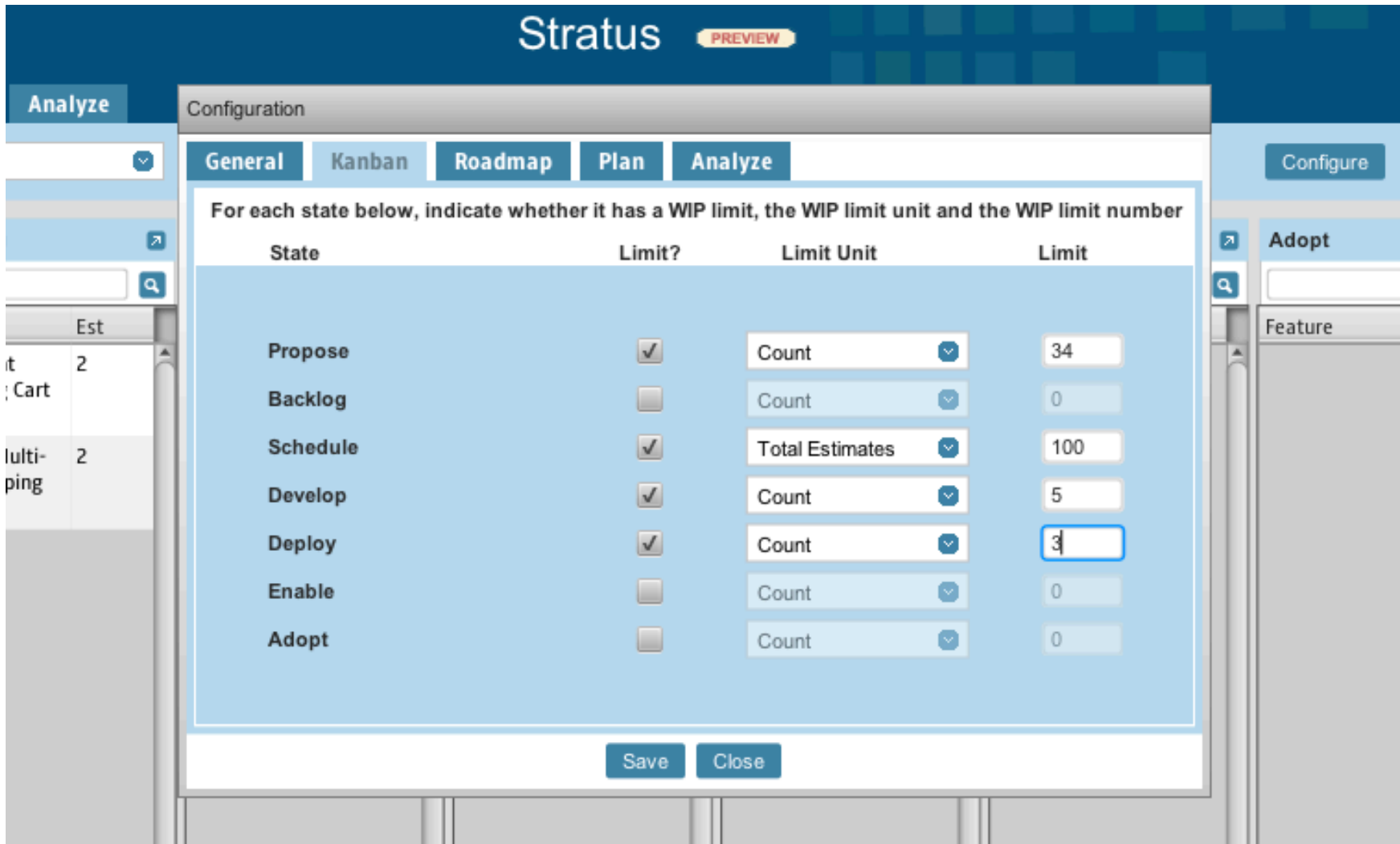
Team	Total	Limit	Compliance	
API Team	0.0	<input style="width: 40px;" type="text" value="10"/>	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0%	Own Selected Feature
Analytics Team	2.0	<input style="width: 40px;" type="text" value="10"/>	<div style="width: 20%; height: 10px; background-color: #ffc107;"></div> 20%	Own Selected Feature
Fulfillment Team	0.0	<input style="width: 40px;" type="text" value="10"/>	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0%	Own Selected Feature
Payment Team	0.0	<input style="width: 40px;" type="text" value="10"/>	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0%	Own Selected Feature
Reseller Portal Team	0.0	<input style="width: 40px;" type="text" value="10"/>	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0%	Own Selected Feature
Shopping Team	0.0	<input style="width: 40px;" type="text" value="10"/>	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0%	Own Selected Feature

©2003-2010 Rally Software Development Corp.
Version 0.9

Stratus Analyze Page



Stratus Configuration Dialog



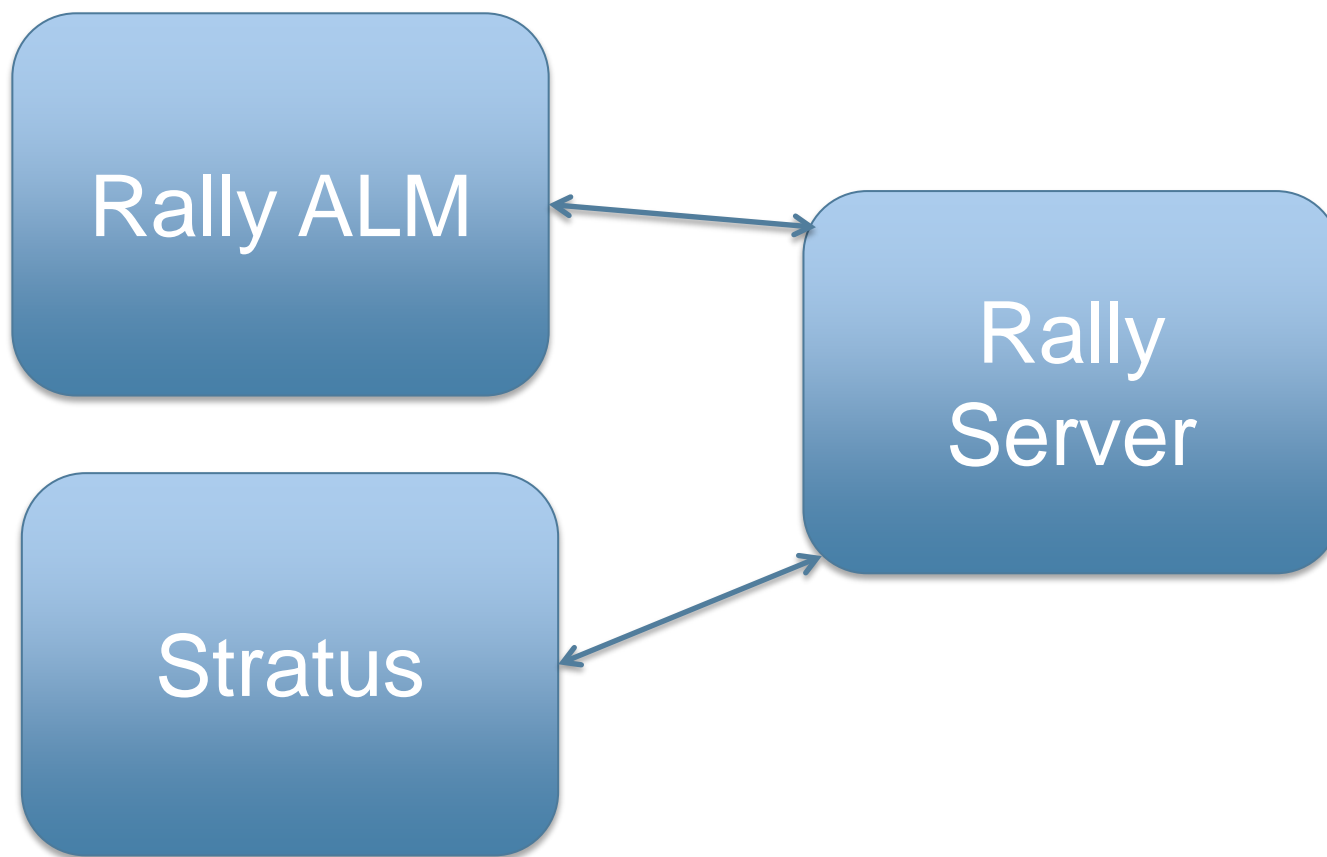
The screenshot shows the 'Stratus' application interface with a 'PREVIEW' badge. The 'Analyze' tab is active, displaying a 'Configuration' dialog box. The dialog has tabs for 'General', 'Kanban', 'Roadmap', 'Plan', and 'Analyze'. The 'Analyze' tab contains a table for configuring WIP limits for various states. The table has columns for 'State', 'Limit?', 'Limit Unit', and 'Limit'. The 'Limit?' column contains checkboxes, and the 'Limit' column contains input fields. The 'Limit' field for the 'Deploy' state is currently set to 3.

State	Limit?	Limit Unit	Limit
Propose	<input checked="" type="checkbox"/>	Count	34
Backlog	<input type="checkbox"/>	Count	0
Schedule	<input checked="" type="checkbox"/>	Total Estimates	100
Develop	<input checked="" type="checkbox"/>	Count	5
Deploy	<input checked="" type="checkbox"/>	Count	3
Enable	<input type="checkbox"/>	Count	0
Adopt	<input type="checkbox"/>	Count	0

JavaFX CSS Styling: Config Dialog with No Style Sheet



Stratus Communicates with Server via HTTP/JSON



Some JavaFX Tech used in Stratus

- > REST Interface / Parsing
 - RedFX
 - Reflection/Recursion/Multitasking
- > CSS Styling
- > JFXtras
 - XTableView
 - XForm
 - XPane
- > Local Storage

XTableView

> Insanely Scalable

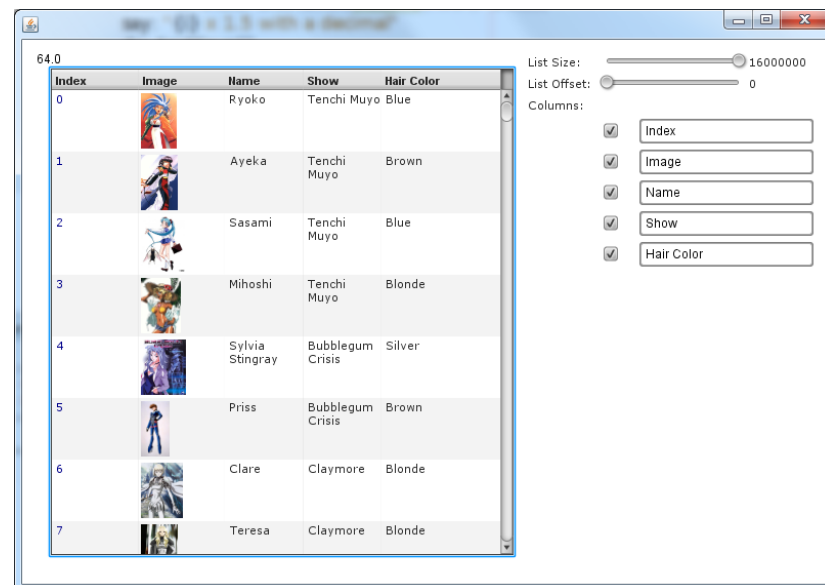
- Up to 16 million rows

> Extreme Performance

- Pools rendered nodes
- Caches images
- Optimized scene graph

> Features:

- Drag-and-Drop Column Reordering
- Dynamic Updating from Model
- Automatically Populates Column Headers
- Fully Styleable via CSS

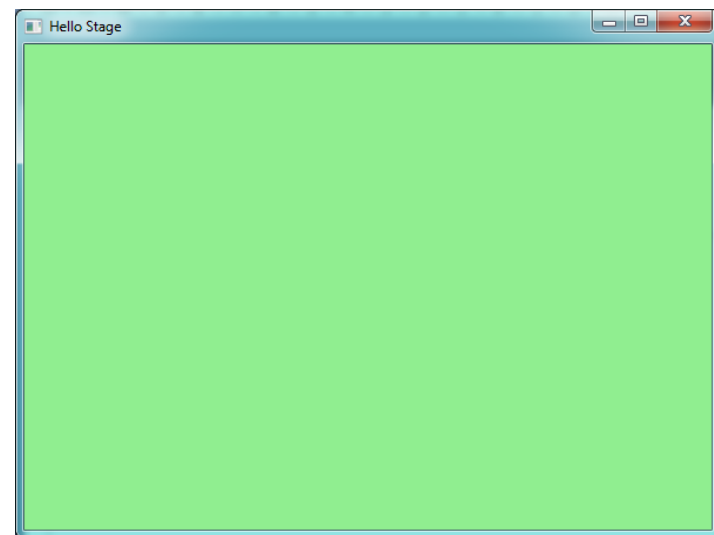


JavaFX in Java

- > JavaFX API follows JavaBeans approach
- > Similar in feel to other UI toolkits (Swing, etc)
- > Researching approaches to minimize boilerplate

Example Application

```
public class HelloStage implements Runnable {  
  
    public void run() {  
        Stage stage = new Stage();  
        stage.setTitle("Hello Stage");  
        stage.setWidth(600);  
        stage.setHeight(450);  
  
        Scene scene = new Scene();  
        scene.setFill(Color.LIGHTGREEN);  
  
        stage.setScene(scene);  
        stage.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        FX.start(new HelloStage());  
    }  
}
```



Why Scala?

- > Shares many language features with JavaFX Script that make GUI programming easier:
 - Static type checking – Catch your errors at compile time
 - Closures – Wrap behavior and pass it by reference
 - Declarative – Express the UI by describing what it should look like
- > Scala also supports DSLs!

Java vs. Scala DSL

```
public class HelloStage implements Runnable {  
  
    public void run() {  
        Stage stage = new Stage();  
        stage.setTitle("Hello Stage");  
        stage.setWidth(600);  
        stage.setHeight(450);  
        Scene scene = new Scene();  
        scene.setFill(Color.LIGHTGREEN);  
        Rectangle rect = new Rectangle();  
        rect.setX(25);  
        rect.setY(40);  
        rect.setWidth(100);  
        rect.setHeight(50);  
        rect.setFill(Color.RED);  
        stage.add(rect);  
        stage.setScene(scene);  
        stage.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        FX.start(new HelloStage());  
    }  
}
```

22 Lines

545 Characters

```
object HelloJavaFX extends JavaFXApplication {  
    def stage = new Stage {  
        title = "Hello Stage"  
        width = 600  
        height = 450  
        scene = new Scene {  
            fill = Color.LIGHTGREEN  
            content = List(new Rectangle {  
                x = 25  
                y = 40  
                width = 100  
                height = 50  
                fill = Color.RED  
            })  
        }  
    }  
}
```

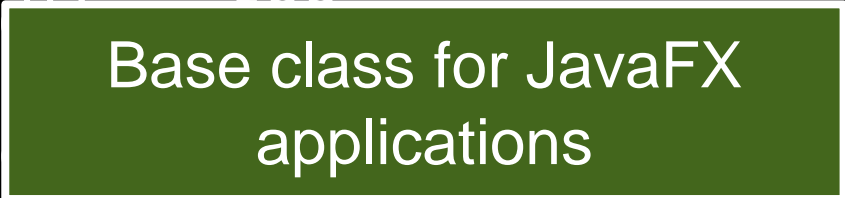
17 Lines

324 Characters

```
object HelloJavaFX extends JavaFXApplication {
  def stage = new Stage {
    title = "Hello Stage"
    width = 600
    height = 450
    scene = new Scene {
      fill = Color.LIGHTGREEN
      content = List(new Rectangle {
        x = 25
        y = 40
        width = 100
        height = 50
        fill = Color.RED
      })
    }
  }
}
```

```
object HelloJavaFX extends JavaFXApplication {  
  def stage = new Stage {  
    title = "Hello Stage"  
    width = 300  
    height = 200  
    scene = new Scene {  
      fill = Color.LIGHTGREEN  
      content = List(new Rectangle {  
        x = 25  
        y = 40  
        width = 100  
        height = 50  
        fill = Color.RED  
      })  
    }  
  }  
}
```

Base class for JavaFX applications



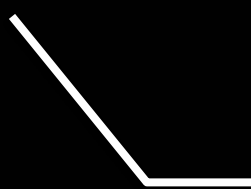
```
object HelloJavaFX extends JavaFXApplication {  
  def stage = new Stage {  
    title = "Hello Stage"  
    width = 600  
    height = 450  
    scene = new Scene {  
      fill = Color.LIGHTGREEN  
      content = List(new Rectangle {  
        x = 25  
        y = 40  
        width = 100  
        height = 50  
        fill = Color.RED  
      })  
    }  
  }  
}
```

Declarative Stage
definition

```
object HelloJavaFX extends JavaFXApplication {  
  def stage = new Stage {  
    title = "Hello Stage"  
    width = 600  
    height = 450  
    scene = new Scene {  
      fill = Color.LIGHTGREEN  
      content = List(new Rectangle {  
        x = 25  
        y = 40  
        width = 100  
        height = 50  
        fill = Color.RED  
      })  
    }  
  }  
}
```

Inline property
definitions

```
object HelloJavaFX extends JavaFXApplication {  
  def stage = new Stage {  
    title = "Hello Stage"  
    width = 600  
    height = 450  
    scene = new Scene {  
      fill = Color.LIGHTGREEN  
      content = List(new Rectangle {  
        x = 25  
        y = 40  
        width = 100  
        height = 50  
        fill = Color.RED  
      })  
    }  
  }  
}
```



List Construction
Syntax

JavaFX 2.0 Highlights: Java and Alternative JVM Languages

- > JavaFX has a new API face
- > All the JavaFX 2.0 APIs will be exposed via Java classes that will make it much easier to integrate Java server and client code.
- > This also opens up some huge possibilities for JVM language integration (e.g. with Ruby, Clojure, Groovy, and Scala)

JavaFX 2.0 Highlights: Open Source Controls

- > JavaFX controls will be open sourced moving forward.
- > This is a huge move in the right direction for the platform, and will make life for us third-party control developers much better.

JavaFX 2.0 Highlights: Multithreading Improvements

- > The move to Java APIs breaks down some of the barriers to multi-threaded programming that were present with JavaFX.
- > Presumably a similar model to Swing will exist where you can launch worker threads, but still have to do all UI operations on a main event thread.

JavaFX 2.0 Highlights: Texture Paint

- > Interesting to see this highlighted, but its use in JavaFX was pioneered by Jeff Friesen and included in JFXtras 0.7

JavaFX 2.0 Highlights: Grid Layout Container + CSS

- > Very good to see that the Grid Layout from JFXtras is continuing to be adopted and evolved.
- > The addition of making it accessible from CSS will make it an extremely powerful layout container suitable for multiple uses.

JavaFX 2.0 Highlights: HD Media

- > Media seems to be getting a big upgrade, which has been long overdue.
- > This is in addition to other promised improvements in full screen capabilities, media markers, animation synchronization, and low latency audio.

JavaFX 2.0 Highlights: HTML5 WebView

- > It is good to see that this is finally getting the attention it deserves.
- > JavaFX is great for dynamic application development, but is not well suited for content presentation.
- > The combination of JavaFX + HTML5 will greatly expand the range of applications that can be developed.

JavaFX 2.0 Highlights: Controls Galore!

- > TableView, SplitView, TabView, and Rich Text to name a few.
- > This is a necessity to build robust enterprise applications.

JavaFX 2.0 Highlights: File (and other) Dialogs

- > This may seem like a minor point, but is incredibly important for building real applications.

JavaFX 2.0 Highlights: HTML5 Support

- > Not to be confused with the WebView, there is also a plan for the successor to JavaFX 2.0 (2012 timeframe) to support an alternate HTML5 rendering pipeline.
- > Not many details are available about this yet, but it could be a huge technological breakthrough if pulled off successfully.
- > The practical applications of being able to deploy your JavaFX application to any HTML5 compliant device is enormous.

And the winner [of the JavaFXpert RIA Exemplar Challenge] is... Abhi Soni!

CARGO CONTAINERS

AIRCRAFT MODEL: BEEHCRAFT 1900C | MAIN DECK | NY CITY -> L.A

533 Tail 465 Sec I 435 Sec H 405 Sec G 375 Sec F 345 Sec E 315 Sec D 285 Sec C 255 Sec B 225 Sec A 0 Nose

Reset Containers Disable Animation View Weights About Beechcraft Cargolader Help

Parameter	Value
Empty Aircraft Weight	9005.0 lbs
Empty Aircraft Moment	2593400.0 lb-in
Empty Aircraft Center of Gravity	287.99554 in
Total Cargo Weight	1924.0 lbs
Total Cargo Moment	651960.0lb-in
Total Aircraft Weight	13902.0 lbs
Total Aircraft Moment	3297126.0 lb-in
Current Center of Gravity	291.8857 in
Fuel Volume	385.0 gls
Fuel Density @ 25 C	6.8 lb/gls
Fuel Weight	2633.0 lbs
Fuel Moment	385.0 lb-in

Weight Coverage

Ratio = 20310.0

53% 6% 10% 4% 9% 4% 7%

A = 0.0 B = 320.0 C = 218.0 D = 460.0 E = 2633.0

F = 0.0 H = 180.0 C = 536.0 F = 2633.0

Weight x 100

Center of Gravity (inches from datum)



JavaOneSM

Thank You

Stephen Chin
steve@widgetfx.org
tweet: @steveonjava

Jim Weaver
jim.weaver@javafxpert.com
tweet: @javafxpert