

IBM Software

Innovate2012

The Premier Event for Software and System Innovation

Next  NOW!



IBM Mobile Enterprise Development Solution

Ayushman Jain

ayushman_jain@in.ibm.com

Why are we here?

10 Billion devices by 2020

300,000 apps developed in last 3 years

61% of CIOs put mobile as priority

76.9 billion App downloads by 2014

45% increased productivity with mobile apps

70% Customer interactions to originate from mobile by 2015

Cross-platform UX becoming critical



📱 How much do people use their mobile phones?



On average, Americans spend **2.7 hours** per day socializing on their mobile device



LOL



That's over **twice** the amount of time they spend **eating**, and over **1/3** of the time they spend **sleeping** each day



Zzzz... lol



and...

91%
of mobile internet access is to socialize...



...compared to **79%** on desktops

The IBM Seer App for Wimbledon



Top mobile development pain points for an app such as Seer

Developing for multiple mobile platforms

- Highly fragmented set of platforms, devices, languages, and tools **increases cost and complexity of development and test**
- Choosing not to support one or more platforms **reduces the reach of an application**



Delivering high quality apps that engage users and meet business objectives

- Poor quality can **negatively impact brand image**
- Bad ratings and comments can **cause other users to avoid trying an app**

Customer Ratings



Integrating with enterprise systems

- Recreating instead of leveraging existing business logic **increases maintenance costs and risk of inconsistent behavior**
- Lack of ready back-end services **slows front-end development and increases potential for last minute integration issues**



Meeting accelerated time to market requirements

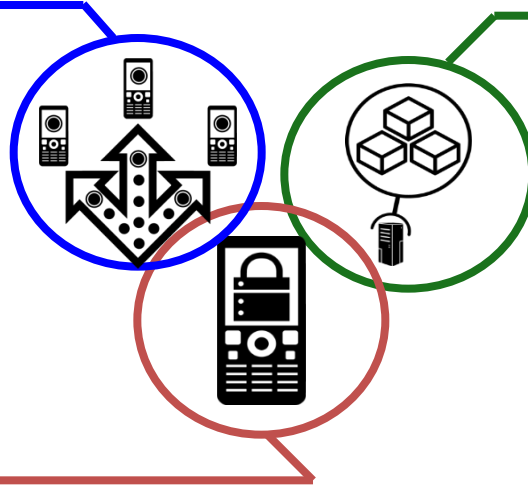
- Hand-off errors and delays between teams **slows progress and responsiveness to features and fixes**
- Misalignment of stakeholders **results in late rework and increased cycle times**



Imperatives in building mobile apps

Extend & Transform

Extend existing business capabilities to mobile devices
Transform the business by creating new opportunities



Build & Connect

Build mobile applications
Connect to, and **run** backend systems in support of mobile

Manage & Secure

Manage mobile devices and applications
Secure my mobile business

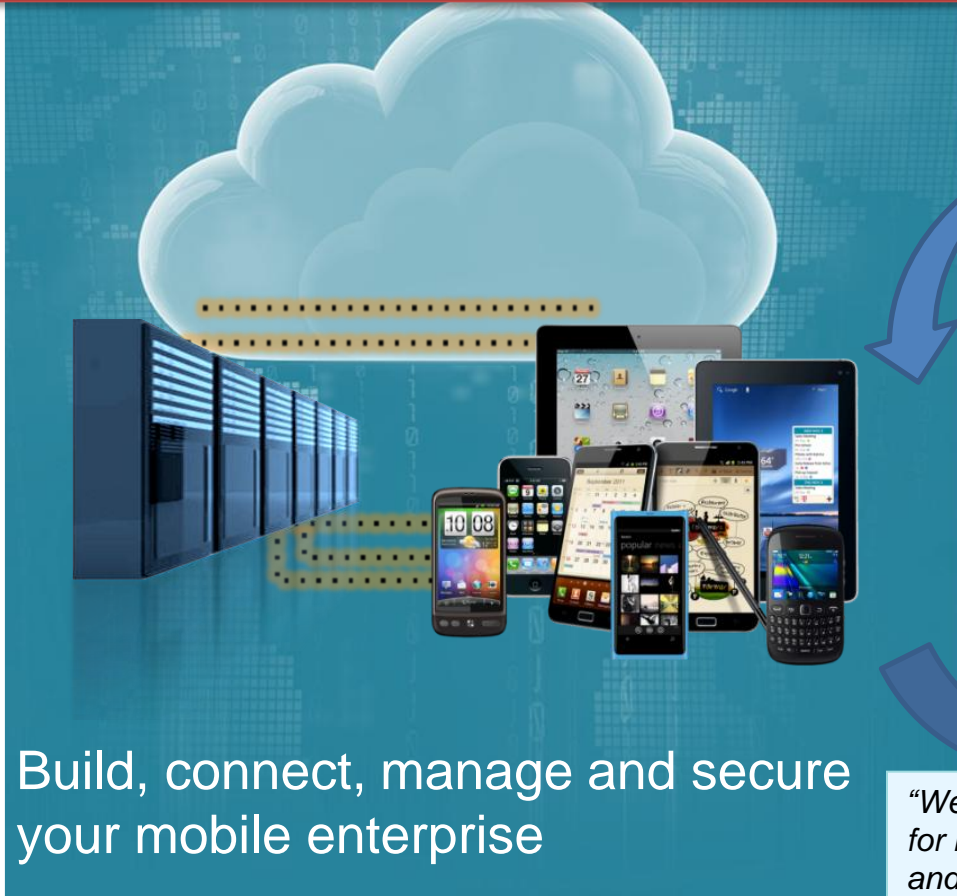
These challenges cannot be addressed using development tools alone!

Some kind of device neutral mobile run-time code is necessary.

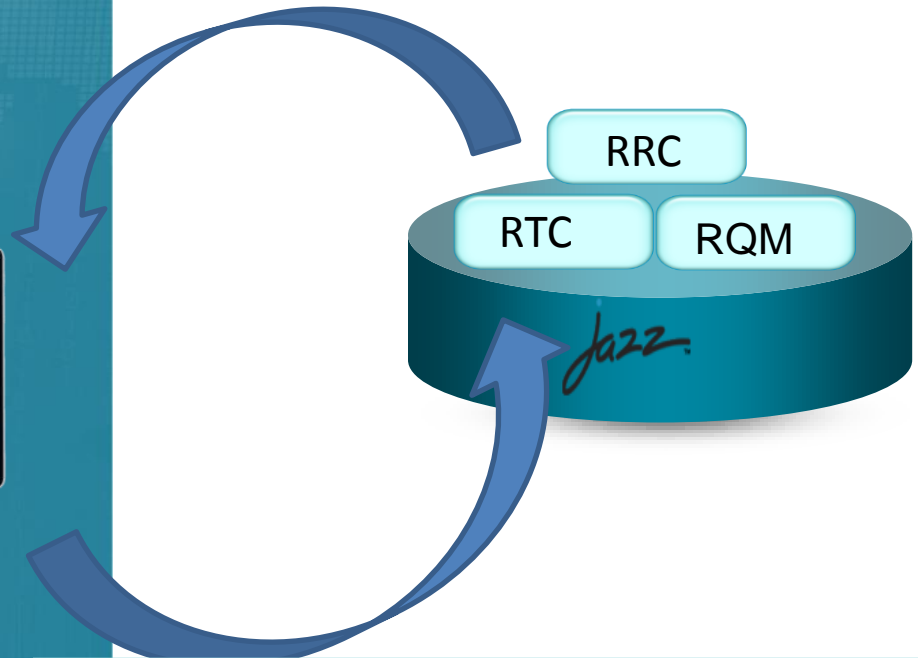
These device neutral mobile run-time products are known as **Mobile Application Platforms**. And they typically come with some associated code construction tooling.

Introducing: The IBM Mobile Enterprise Solution

Includes IBM Worklight V5.0, IBM WebSphere Cast Iron, IBM Endpoint Manager for Mobile Devices



Build, connect, manage and secure your mobile enterprise

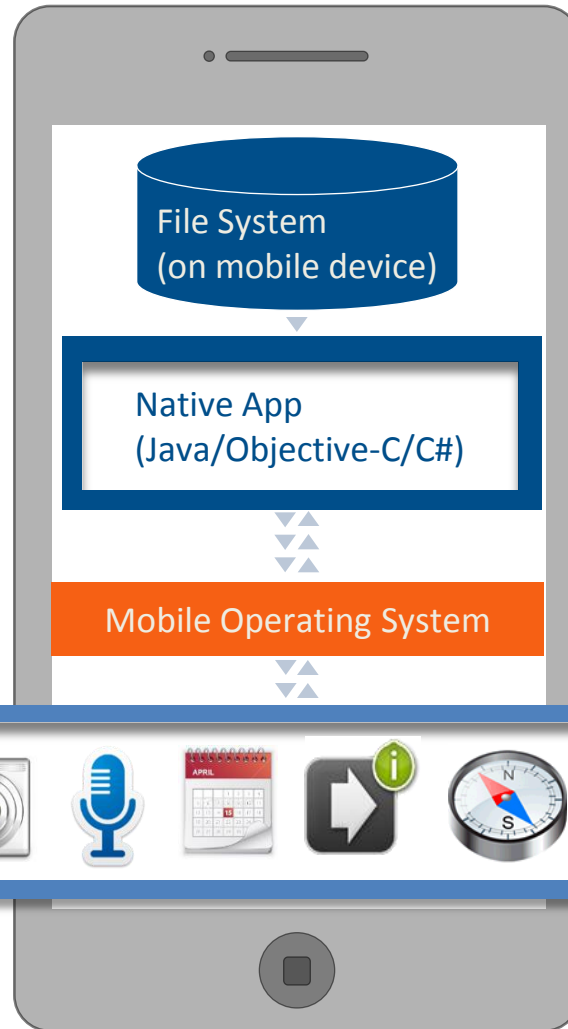


“We chose IBM Worklight because it was the best technology for Lotte to consolidate application development, enhancement and maintenance, while ensuring cost savings and timely delivery to our customers.”

— Kang, Chang Nam, IT Planning Director, Lotte Credit Card

Challenge 1: Developing for multiple mobile platforms

Downloadable (Native) Apps



High-quality user experience and full device access.

Platform-specific, requires unique expertise, expensive to develop and maintain.



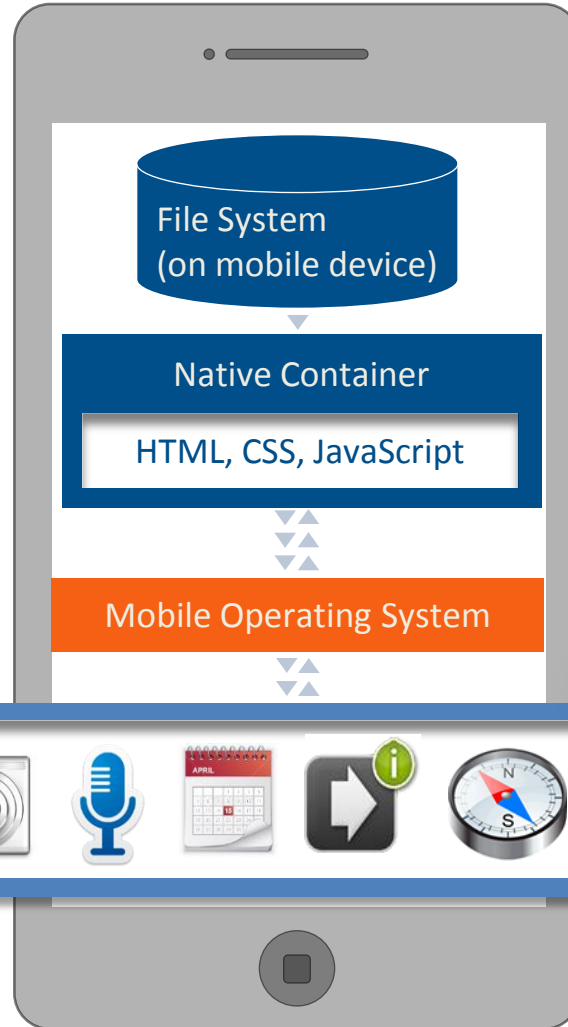
Web Apps



Written in HTML5
JavaScript and CSS3.
Quick and cheap to
develop.

Less powerful than
native and limited device
access.

Hybrid Apps



Combines best of both worlds:

Primarily written in HTML5, CSS, JS while allowing full access to device capabilities.



What is Phonegap (or Apache Cordova)?



- Consider – a web-based movie ticket booking application that lets you invite friends from the phone contact book. Problems:
 - How can you query the platform's contact book from within your web app?
 - How can you do so in a platform agnostic way?
 - How can you mix web page to book the ticket and a native page to show contact book?
- Phonegap shows the way - provides javascript APIs which abstract the platform's native services.
- Develop an app using HTML5, Dojo, javascript once. Package it for various platforms using Phonegap.
- Use any UI toolkit – Dojox.mobile, Sencha touch, Titanium, etc.
- Integrate native elements with the web-based elements.

Supported features

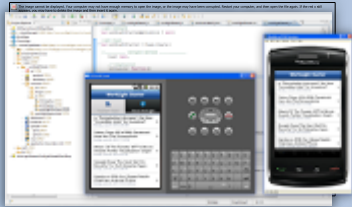
	iOS	iOS	Android	OS 4.6-4.7	OS 5.x	OS 6.0+	WebOS	WP7	Symbian	bada
	iPhone / iPhone 3G	iPhone 3GS and newer	Android	OS 4.6-4.7	OS 5.x	OS 6.0+	WebOS	WP7	Symbian	Bada
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
COMPASS	✗	✓	✓	✗	✗	✗	✗	✓	✗	✓
CONTACTS	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓
FILE	✓	✓	✓	✗	✓	✓	✗	✓	✗	✗
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗
NETWORK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (ALERT)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Solution: IBM Worklight

- Develop rich mobile apps that work across all major platforms using **standard technologies**, such as HTML5
- Supports **mobile web and hybrid approaches** – enables common code base across platforms, but with the ability to access native functions
- Connect apps to enterprise back-end systems and cloud services through **mobile-optimized middleware**
- Manage applications and versioning from one **centralized admin console**

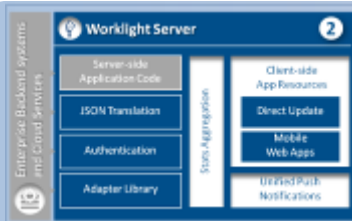


Worklight mobile platform overview



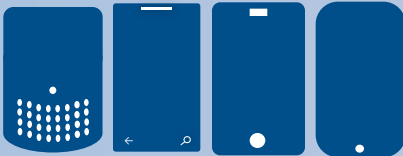
Worklight Studio

The most complete, extensible development environment with maximum code reuse and per-device optimization



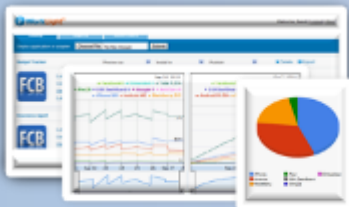
Worklight Server

Mobile middleware offering unified push notifications, version management, security and integration



Worklight Runtime Components

Extensive libraries and client APIs that expose and interface with native device functionality and the Worklight Server

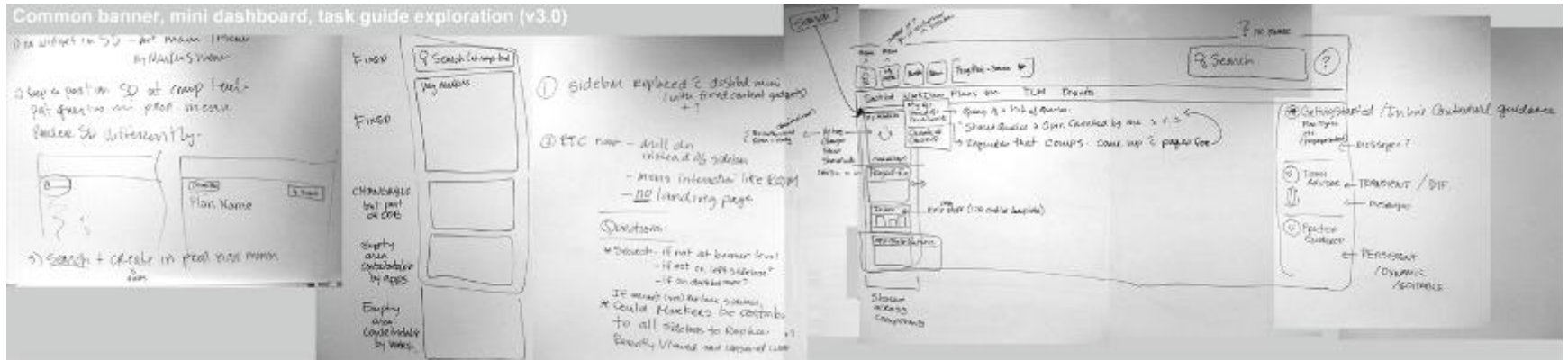


Worklight Console

A web-based console for real-time analytics and control of your mobile apps and infrastructure

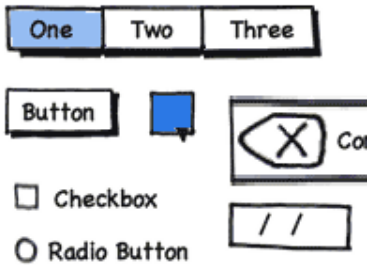
Application visualization

Is this how it's done?



- First class user experience is critical in modern applications
 - Particularly true with mobile applications
- Teams achieve first class user experience by iterating through a “UI Lifecycle”
 - Spans from
 - **Low fidelity**, easily changeable **sketches**
 - to **high fidelity wireframes**
 - to **UI implementation**
 - Different teams do it different ways.

The UI lifecycle

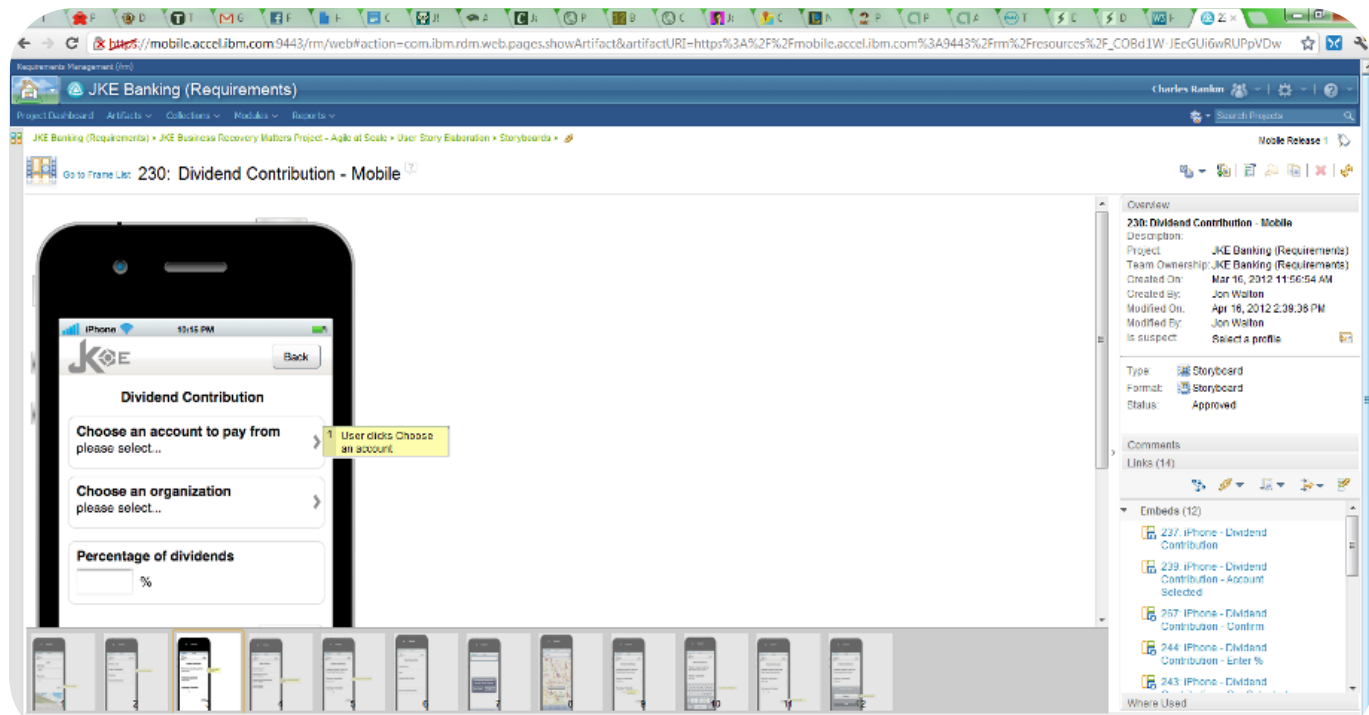


UI Sketching

The UI lifecycle



UI Sketching



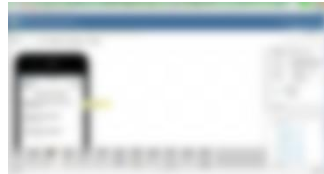
UI Specification

Tool shown: Rational Requirements Composer
 Capture requirements into a rough UI sketch and create a storyboard

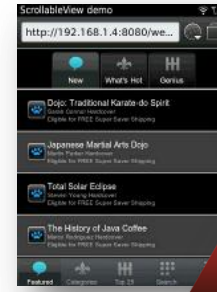
The UI lifecycle



UI Sketching



UI Specification



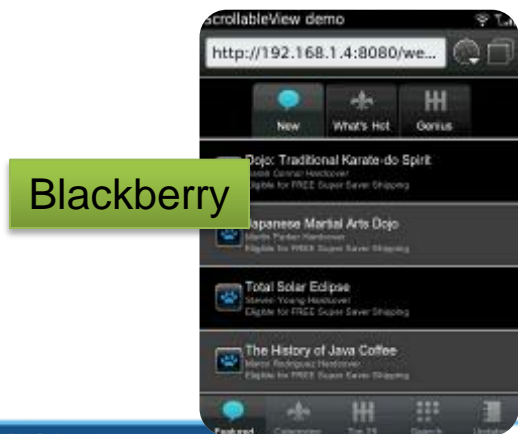
appcelerator



Choose a UI toolkit

Dojo Mobile

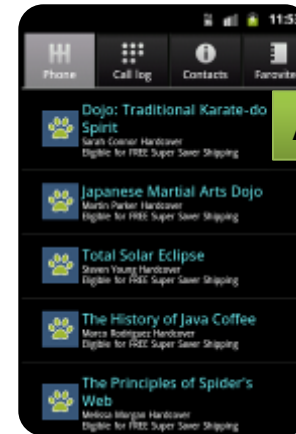
- A Dojo-based JavaScript widget set for creating mobile web applications
 - ▶ Available since Dojo 1.5 in open source as experimental
 - ▶ GA in Dojo 1.6
 - ▶ IBM support via IBM Web2.0, IBM Worklight and Mobile Feature Pack for WebSphere 1.1
- Provides lightweight UI widgets for mobile scenarios
 - ▶ Native device access or coding is not in the scope of Dojo Mobile (use PhoneGap for this)
- Allows developing device-specific or device-neutral look & feel applications
 - ▶ iPhone, Android, Blackberry or create your own custom themes
- Server technology agnostic
- Reuse application code across devices with a simple style sheet change!



Blackberry



iPhone



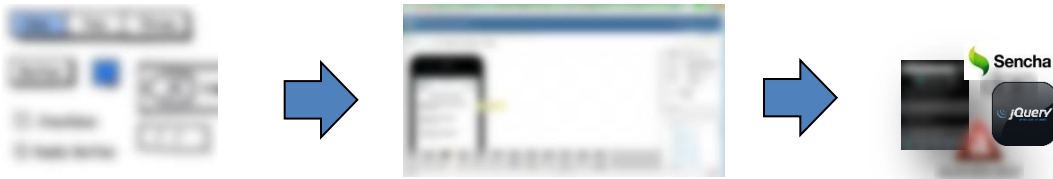
Android

Dojo 1.7 Themes: iPhone, iPad, Android 2.x, Blackberry



Manual or Automatic device detection to apply themes

The UI lifecycle



UI Specification

Choose a UI toolkit

UI Sketch

Design the UI

Tool shown: *Maqetta*

Part of Dojo Foundation

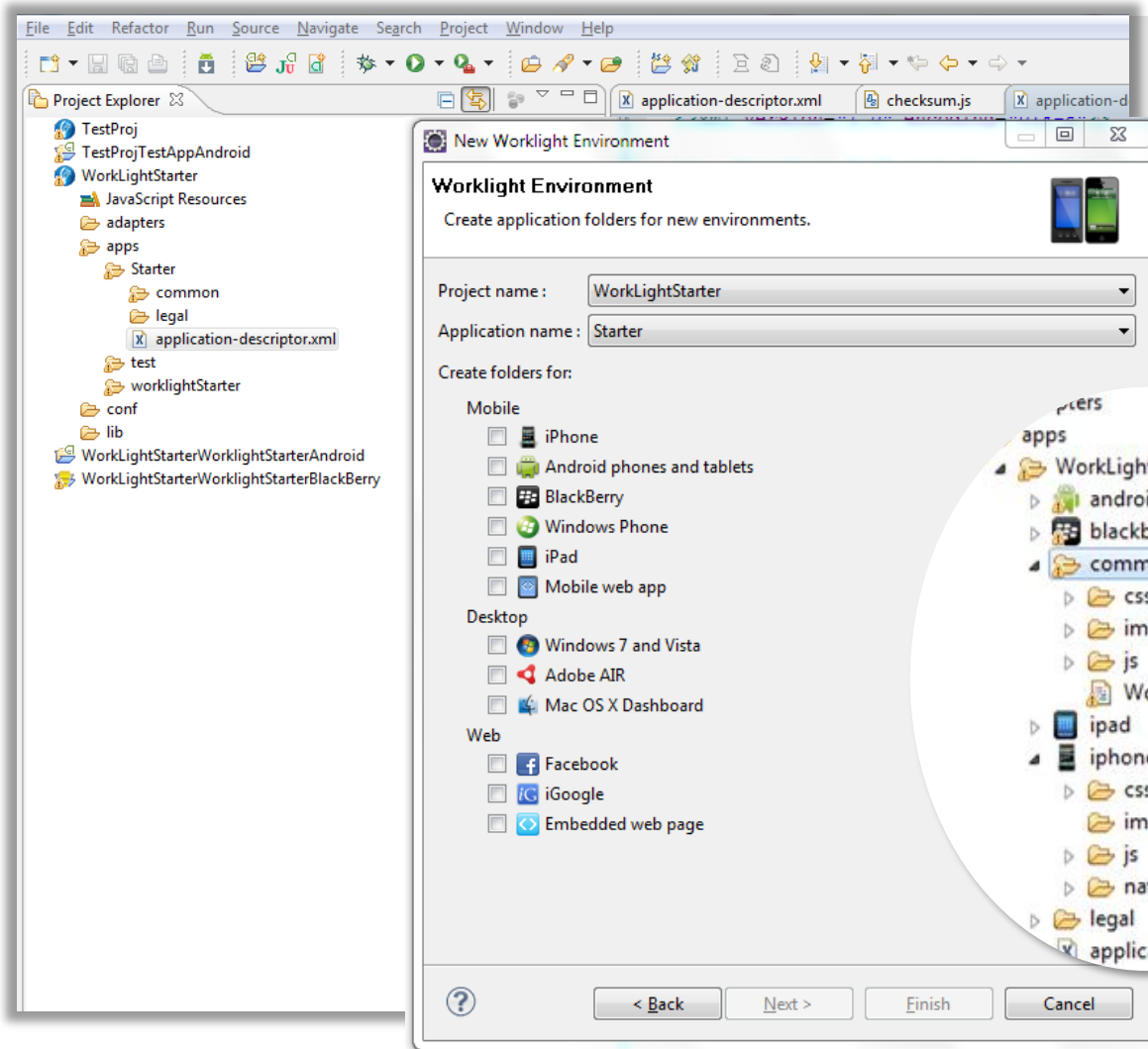
Community can run Maqetta for free (as is basis)

Can either register or "give it a try" as temporary guest

Integrates with:



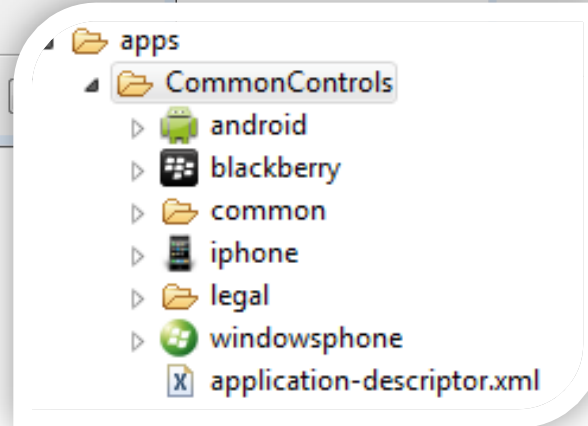
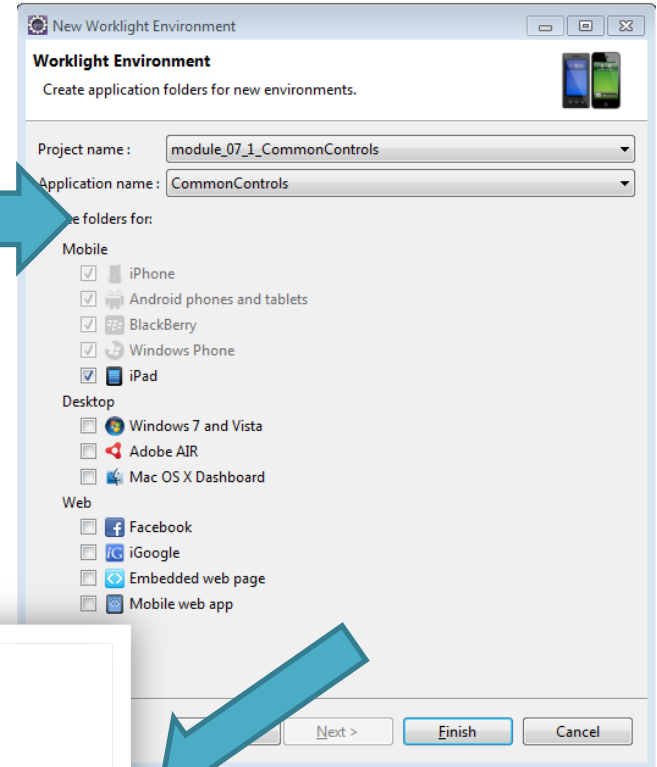
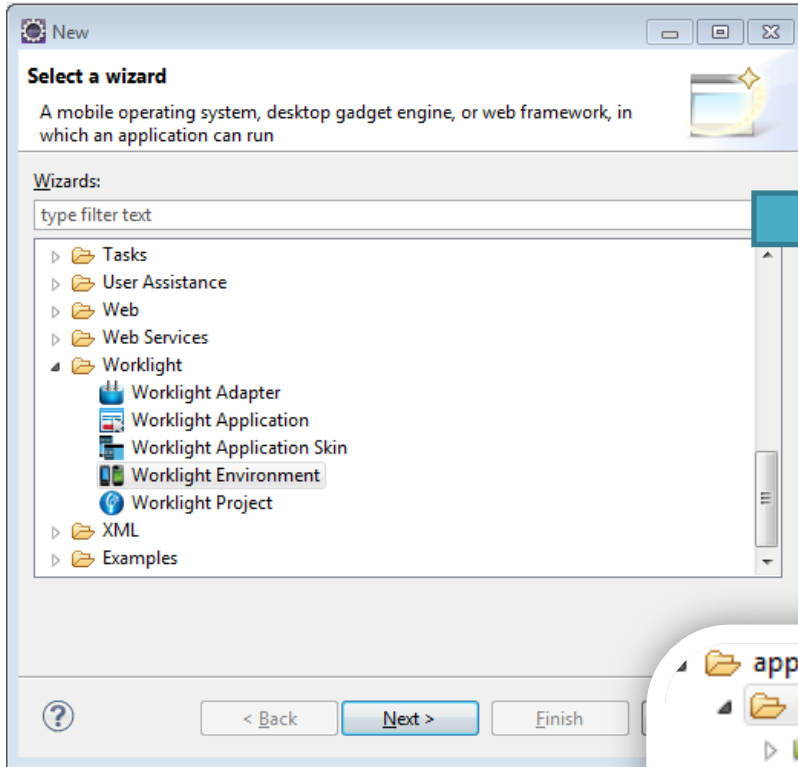
Common web codebase facilitates reuse



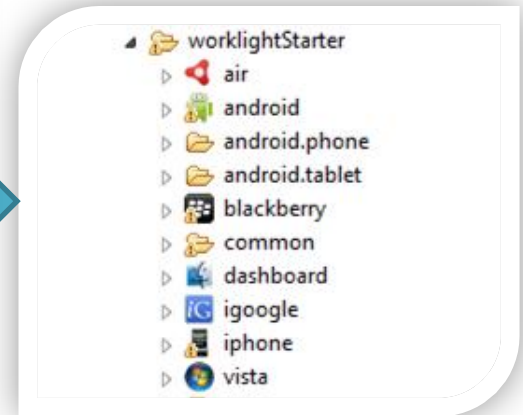
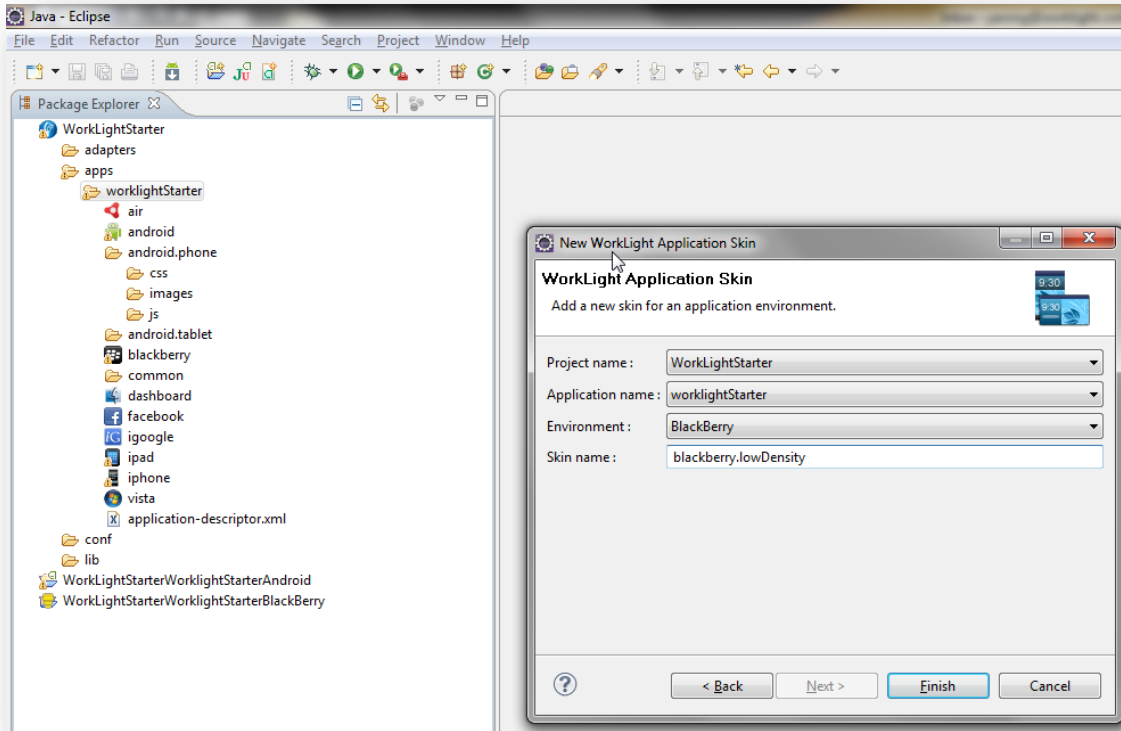
Worklight projects contain:

- Applications (mobile UI)
- Adapters (stateless services)

Add Environment



Add skins



Source editing tools

	HTML	JavaScript	Dojo	CSS	JSON
Code Assist	X	X	X	X	X
Validation	X	X	X	X	X
Outline	X	X	X	X	X

The screenshot shows an IDE with a code editor and two completion windows. The code editor contains HTML and JavaScript code. The JavaScript completion window shows methods like `getElementById`, `getElementsByName`, `getElementsByTagName`, and `getElementsByTagNameNS`. The CSS completion window shows properties like `border-bottom-right-radius`, which is expanded to show supported browsers: Chrome 40, Firefox 30, and Safari 5.0.

WYSIWYG UI construction

The screenshot displays a Rich Page Editor interface for constructing a mobile UI. The main design area shows a mobile application with a 'Fruits' section containing 'Apples' and 'Oranges' items. The code editor shows the underlying HTML and Dojo widget markup, including a callback function and widget configurations. A configuration table at the bottom allows for modifying the items in the Round Rectangle list.

```

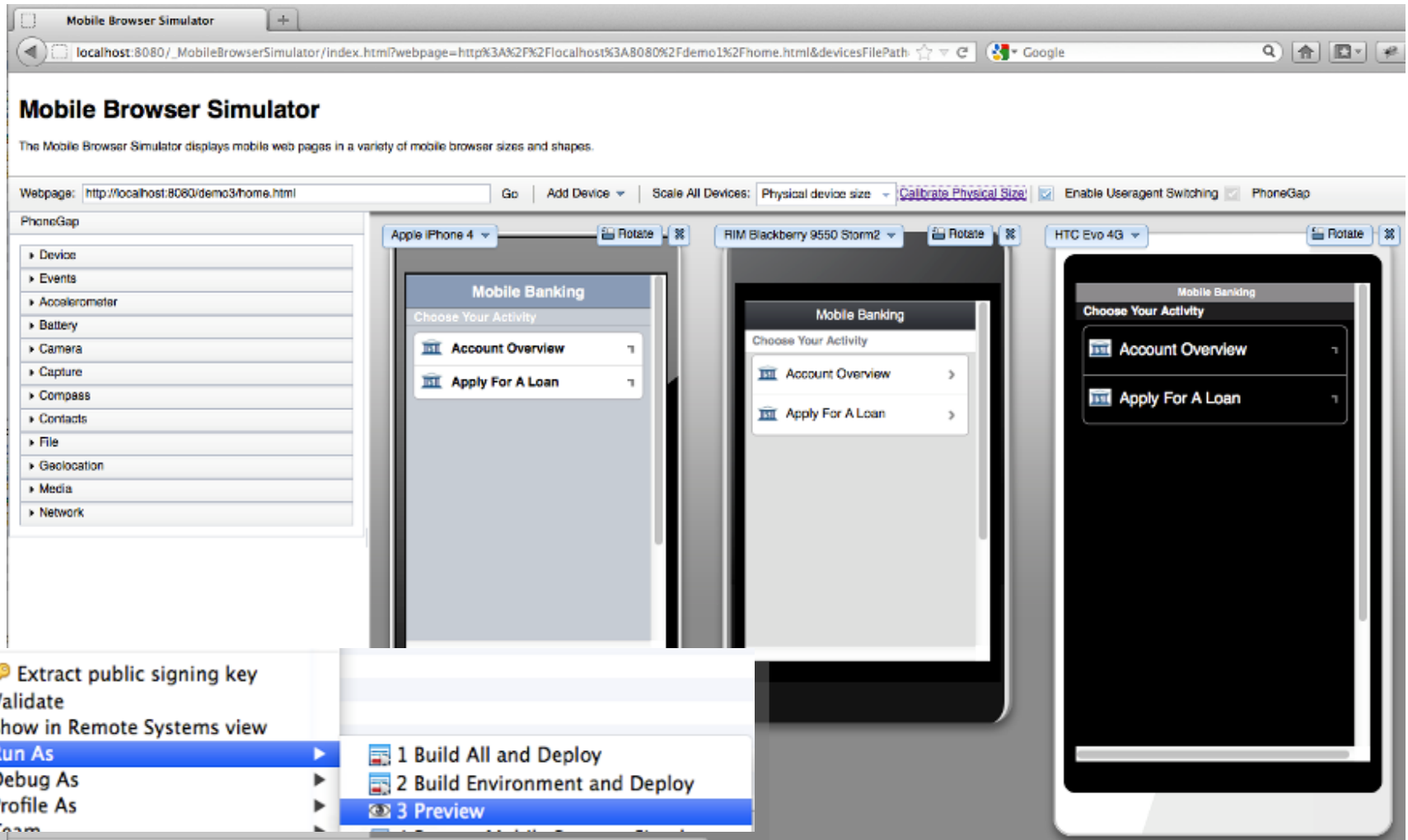
    "dojox/mobile/ListItem", "dojox/mobile/EdgeToEdgeCategory" ],
    // Callback function, invoked on dependencies evaluation results
    function(dojox) {
        dojox.ready(function() {

            });
        });
    </script>
    </head>
    <body>
    <div data-dojox-type="dojox.mobile.View" id="view0"
        data-dojox-props="selected:true">
    <h1 data-dojox-type="dojox.mobile.Heading"
        data-dojox-props="label:'Demo'"></h1>
    <div data-dojox-type="dojox.mobile.EdgeToEdgeCategory">Fruits</div>
    <div data-dojox-type="dojox.mobile.RoundRectList">
    <div data-dojox-type="dojox.mobile.ListItem"
        data-dojox-props="label:'Apples',icon:'apples.png',moveTo:'apples'>
    <div data-dojox-type="dojox.mobile.ListItem"
        data-dojox-props="label:'Oranges',icon:'oranges.png',moveTo:'oranges'>
    </div>
    </div>
    <div data-dojox-type="dojox.mobile.View" id="apples"
        data-dojox-props="selected:false">
    <h1 data-dojox-type="dojox.mobile.Heading"
        data-dojox-props="label:'Apples',back:'Home',moveTo:'view0'"></h1>
    </div>
    <div data-dojox-type="dojox.mobile.View" id="oranges"
        data-dojox-props="selected:false">
    <h1 data-dojox-type="dojox.mobile.Heading"
        data-dojox-props="label:'Oranges',back:'Home',moveTo:'view0'"></h1>
    </div>
    </body>
  
```

Page	Use the table below to modify the items in this Round Rectangle list						
html	Icon	Label	Right Text	Variable Height	Move To	Transition	
body	apples.png	Apples		Fixed	apples	fade	Add Remove Up Down
View	oranges.png	Oranges		Fixed	oranges	flip	
Mobile List							
List Items							

Construct Mobile UI with Rich Page Editor

IBM Worklight Studio preview feature



Application distribution (for development and test)



The screenshot shows the IBM Developer Application Store interface. The top navigation bar includes "IBM Developer Application Store", "Applications", "Devices", and "Welcome demo1". The main content area is titled "Application Management" and lists available applications:

- Dojo Showcase** (Android): Installation: unrestricted, Administration: restricted, version 20120109, 02/04/12.
- Managed Mysurance** (Android): Installation: unrestricted, Administration: restricted, version 20120109, 02/04/12.
- Notifications Sample** (Android): Installation: unrestricted, Administration: restricted, version 20120109, 02/04/12.
- TabletApp** (iOS): Installation: unrestricted, Administration: restricted, version 1.0, 02/04/12.

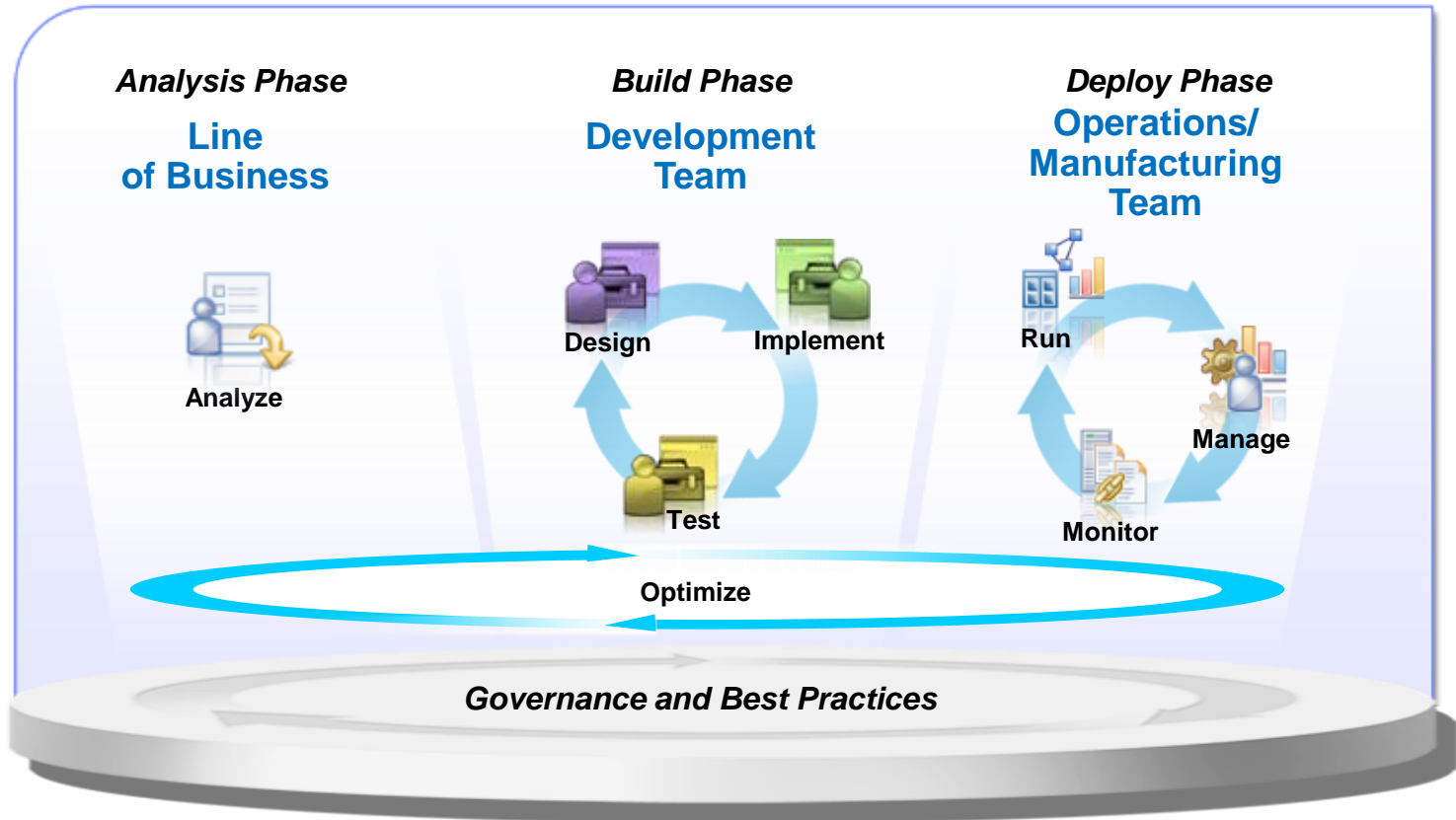
The mobile device screen displays a "Listed Applications" list with three items: Mysurance, PhoneGapShowcase, and DojoShowcase.

A cross platform private mobile application store similar to public app stores but focused on the needs of an organization or a team

Key capabilities:

- delivers distribution and management of mobile applications within a company / teams
- easy distribution of iOS and Android apps within a team
- provides versioning and updates
- centralizes rating and feedback information
- controls who can modify or install an application
- easy to install and simple to run

Mobile development is more than just coding ...



Coding is just one component of the mobile application development lifecycle

Challenge 2: Design and functional quality are both critical to success

- Customers demand good user experience
- Customer facing (B2C) applications
 - ‘Face of the Business’
 - Engage customer personally and drive loyalty
 - User experience is key to brand perception
- Employee facing (B2E) applications
 - Increase worker productivity
 - Speed decision making and action
 - Poor user experience compromises investment



Customer Ratings



Lifecycle techniques to help optimize user experience

- **Improve collaboration** and communications between line-of-business stakeholders and development team
 - Ensure that development understands and will deliver an app that meets business objectives
 - Engage non-technical stakeholders
 - Improve collaboration to find a cheaper yet satisfying solution
- Put more focus on the **application design** and conceptualization phase
 - Iterate during the phase of the lifecycle when it is cheapest to make changes, not when the code is done and changes require drastic re-architecture
- **Use UI sketching, storyboarding, and business process diagrams**
 - Improves communications – “a picture is worth a thousand words”
 - Avoid over-elaboration with sketches; focus on high-level, big-picture issues
 - Ensure proper application flow and interactions with business process diagrams

Storyboards depict the flow of the application

- ✓ Collaborate as a team using comments and annotations
- ✓ Get feedback from non-technical stakeholders

Formal reviews drive agreement and prevent re-work

- ✓ Reviews & approvals ensures artifacts are reviewed and/or approved by key team members and captures compliance requirements.

Requirements



JKE Banking (Requirements) > Approval

Participant's Review: In progress 0% completed | Your role: Approver

Overall Review: Draft → In progress 65% completed | Pause Review

Due: Apr 18, 2011

Instructions to reviewers:
List of requirements for our second sprint. All comments have been resolved. Please review and approve requirements as written.

<input type="checkbox"/>	Participant	Type of Participant	Review results
<input type="checkbox"/>	Bob	Approver	Done - 5 Approved
<input type="checkbox"/>	Deb	Approver	Done - 5 Approved
<input type="checkbox"/>	Marco	Approver	60% - 2 Approved, 1 Disapproved
<input type="checkbox"/>	Tanuj	Approver	

Work Items



Task 41 * ?

Summary: * Implement - Frequency of dividend transfer

Overview | Links | Approvals | History

Approvals (1) - 1 pending

New: Approval

Type	Name	State	Due
Review	Check of algorithm to prevent transfer on weekend	Pending	Apr
	AI	Pending	
	Curtis	Pending	
	Tanuj	Approved	

Add Approvers...

Test Artifacts



18 : Dividend Allocation by Percentage

Test Case Overview | Snapshots | History

Originator: Tanuj | Owner: Tanuj | State: Under Review

Description: select a list of potential donations and enter percentages for each

Formal Review ?

List the people who will be reviewers and approvers of this content and define your review process.

View: All

Show All | Items per page | Previous | 1 - 4 of 4 | Next

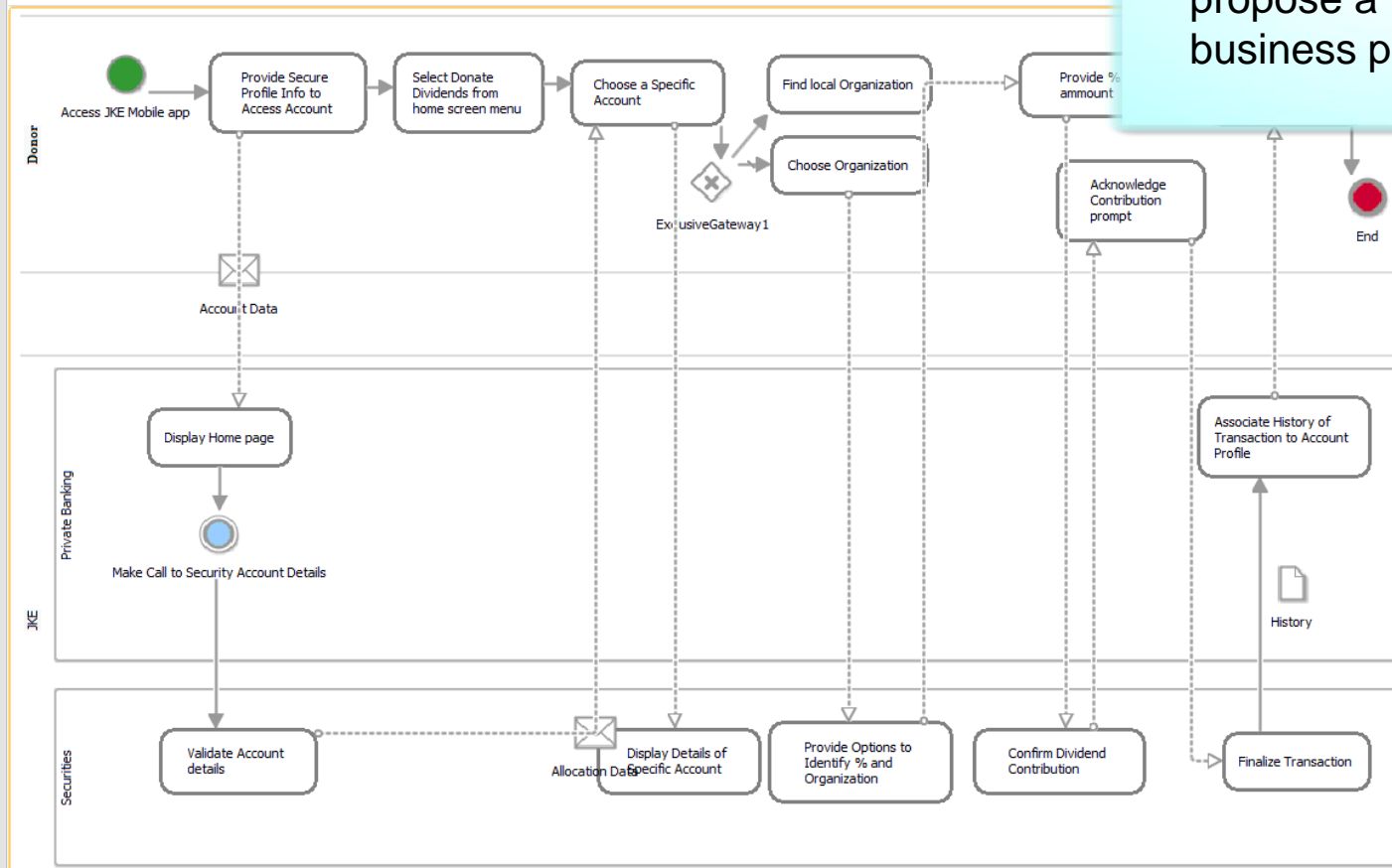
<input type="checkbox"/>	Review Type	Name	Status	Comments
<input type="checkbox"/>	Approver	Bob	Pending	
<input type="checkbox"/>	Reviewer	Sally	Pending	

Diagrams communicate interactions and flows

- ✓ Provide visual representation of flow and interactions
- ✓ Capture the current state and propose a future state with business process diagrams

JKE Banking (Requirements) > JKE Business Recovery Matters Project > Processes >

370: Allocating Dividends to a Cause via JKE Mobile



Link To (1)

320: Donor must be registered user to access account details (Provide Secure Profile Info to Access Account)

Link From (1)

141: Support dividend processing via mobile devices

Satisfied By (2)

- 363: Mobile – Allocate dividends by Percentage
- 365: Mobile – Donor Chooses Local Organization

Illustrated By (5)

- 362: Dividend Contribution – Mobile
- 350: iPhone - Home (Display Home page)
- 352: iPhone - Select Account (Choose a Specific Account)
- 356: iPhone – Dividend Contribution – Enter % (Provide % amount)
- 360: iPhone – Find charity on map (Find local Organization)

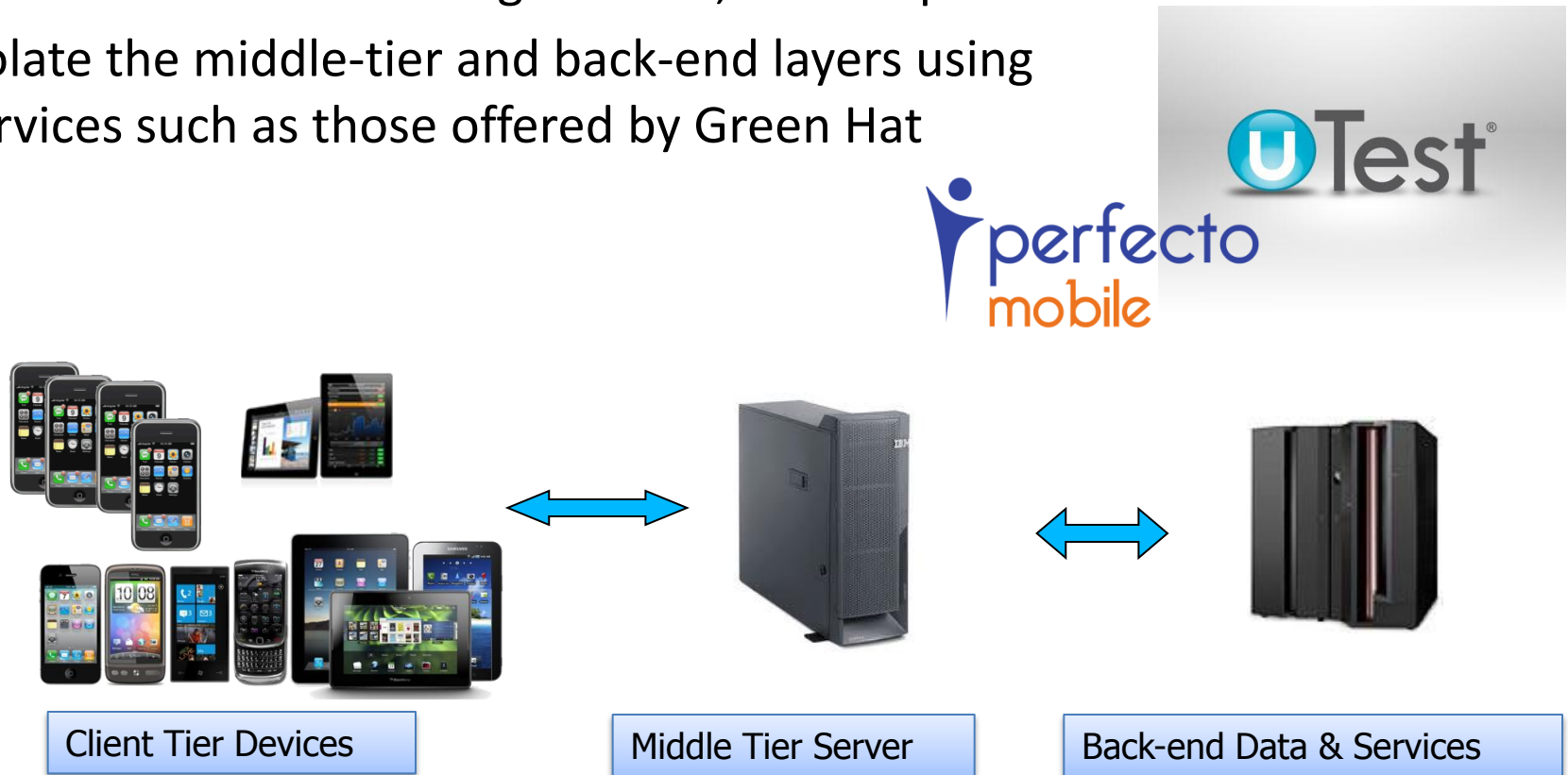
Mobile Testing Challenges

According to a study highlighted on TechCrunch, the average shelf life of an iPhone app is roughly 30 days. For free apps, less than 20% of users return to an app even one day after downloading it. And by day 30, less than 5% of users are still using it.

- Testing on emulators does not guarantee functioning on real device.
- Device environment diversity – should I buy a mobile device farm for my testing team?
- Automating testing across devices using scripts – programming skills?
- Middle tier and backend set up or simulation
- Real life testing through BETA release not possible with short time to market and low popularity unless you're a big brand.

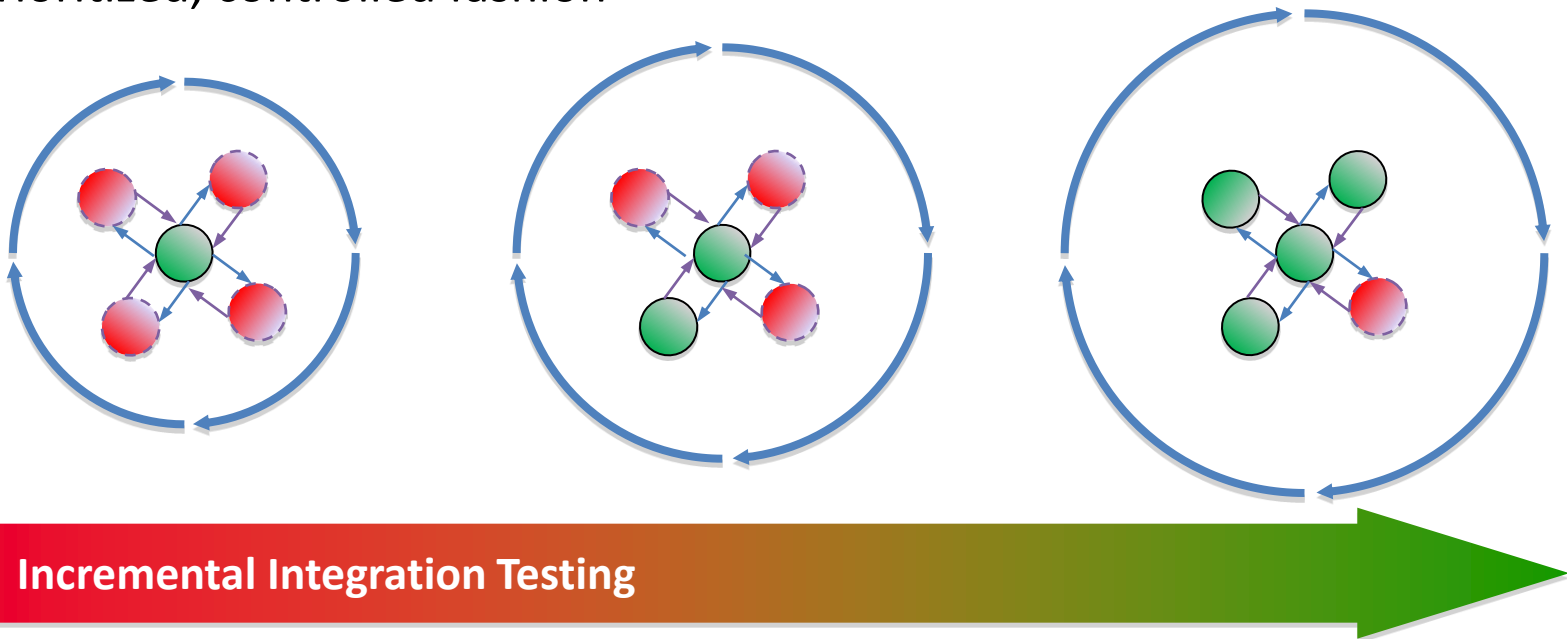
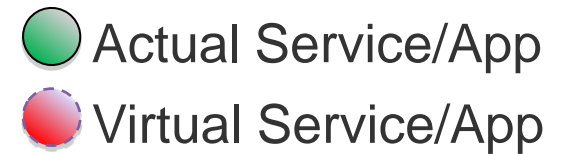
Setting up a mobile testing infrastructure

- Request test runs on device cloud providers, such as Perfecto Mobile.
- Use crowdsourced testing services, such as μ Test.
- Isolate the middle-tier and back-end layers using services such as those offered by Green Hat



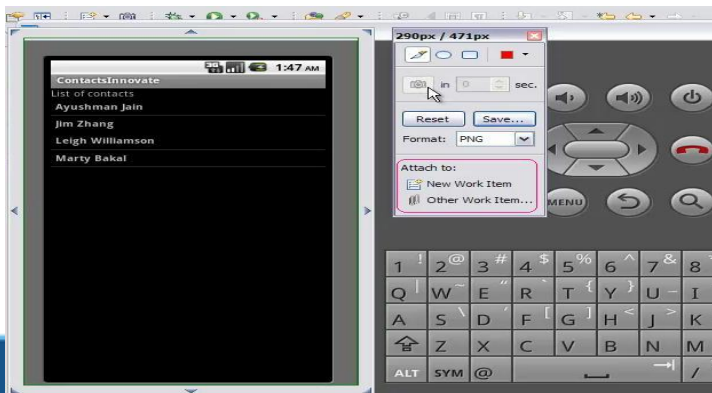
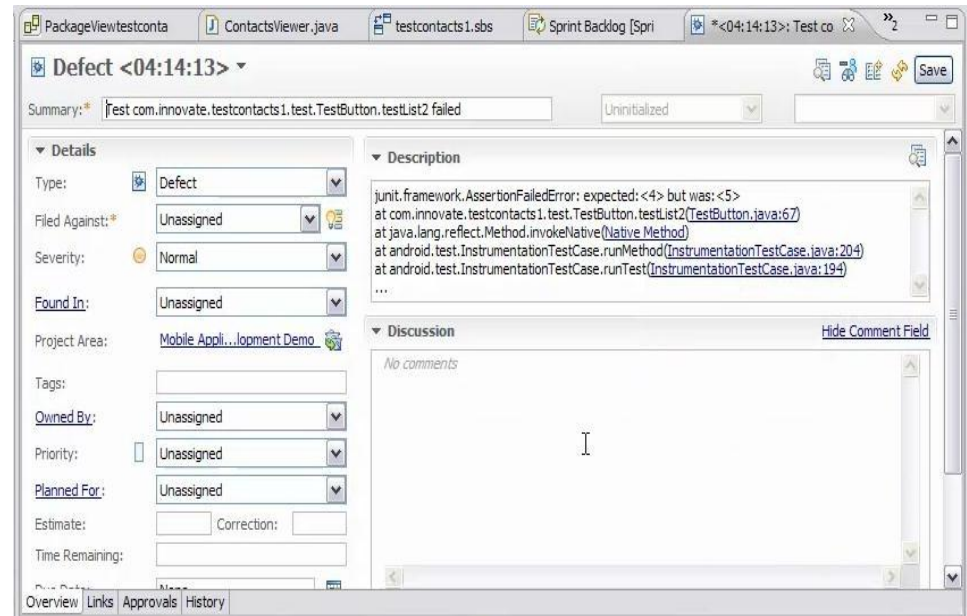
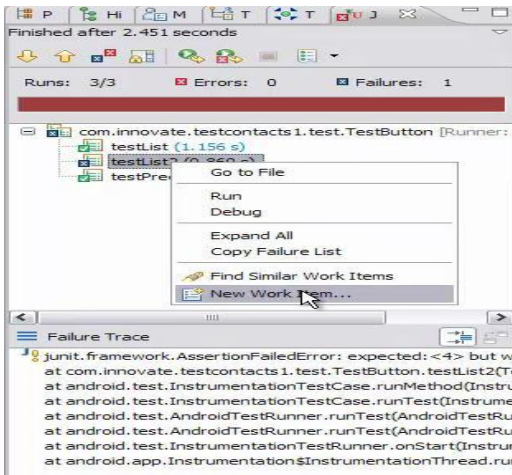
Green Hat Virtualization Technology

- Test Virtualization is an enabler for continuous Integration Testing
- Services, applications, systems are introduced into the continuous integration cycle in a prioritized, controlled fashion



Traditional mobile testing

- Testing on emulators or browser-based simulators.
- Unit testing using Xcode for iOS and Eclipse/Eclipse-based tools for Android.



Tool shown: Rational Team Concert

Functional Testing

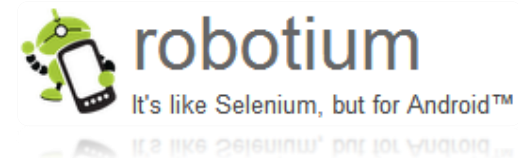
- Test basic functionality – UI appearance, procedural behaviour, placement of UI elements
- Should be done early on in the release cycle.
- Can be done manually, but automated scripts are ideal.
- Use tools that can help create test scripts or test cases in a black-box fashion.

```

solo.sendKeys(Solo.MENU);
solo.clickOnText("More");
solo.clickOnText("Preferences");
solo.clickOnText("Edit File Extensions");
Assert.assertTrue(solo.searchText("rtf"));

solo.clickOnText("txt");
solo.clearEditText(2);
solo.enterText(2, "robotium");
solo.clickOnButton("Save");
solo.goBack();
solo.clickOnText("Edit File Extensions");
Assert.assertTrue(solo.searchText("application/robotium"));

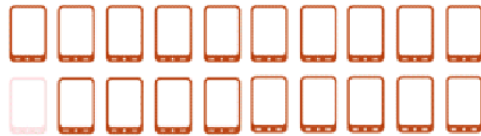
```



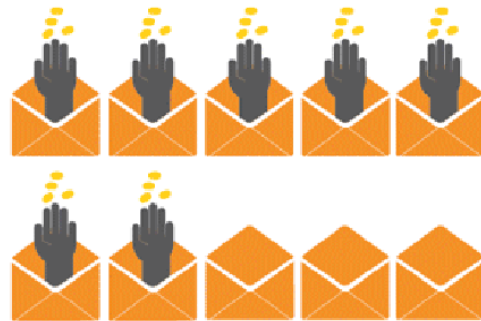
Security Testing

Real threats, right now

Loss and theft



Spam



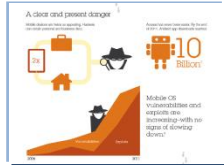
Malware

In 2011, mobile device users saw a **155%** increase in malware across all platforms.⁵

Google Android malware grew **400%** from June 2010 to January 2011.⁵

Wifi

Wifi hotspot are set to increase **350% by 2015**⁵ providing more opportunities for "man-in-the-middle" attacks.



Security Testing

- Static analysis of mobile app.
 - Confidentiality
 - Authentication
 - Availability
 - Integrity
 - Authorization
 - Non-Repudiation
- Manual penetration testing
 - Cross-Site Scripting
 - Denial of Service
 - SQL Injection
 - Buffer Overflow
- Search for vulnerabilities, especially in the web code and XSS, SQL and other common operations.

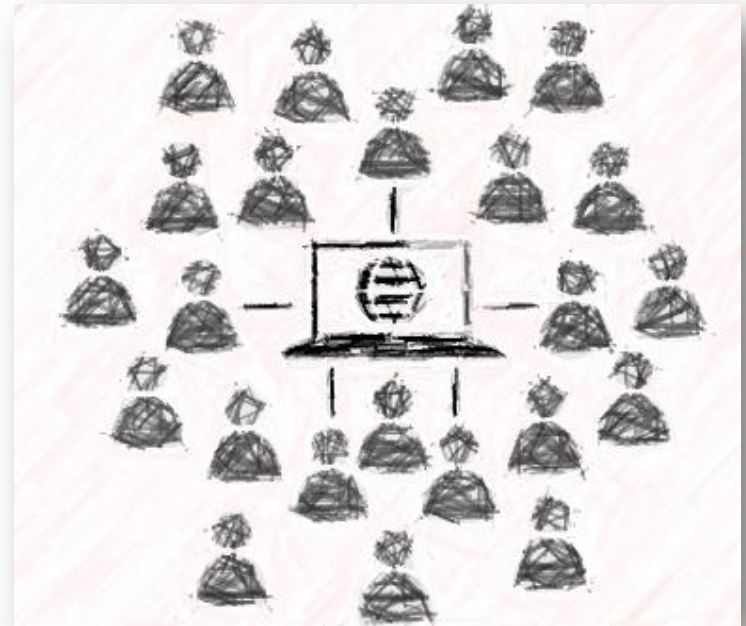
IBM Worklight and IBM Appscan offer unique features to plug security holes

Usability Testing

- User experience is an important part of mobile applications.
- A good user experience on iPhone doesn't guarantee one on Android.
- Different screen sizes and resolutions need to be tested.
- For mobile web applications, behaviour on different mobile browsers need to be tested.

Load testing

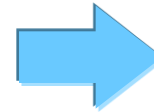
- LIVE: Get people from across the world to simultaneously test your app.
- SIMULATION: Use tools to create simultaneous sessions and stress test your app.
- Can be combined with the other forms of testing.
- Cannot be done in-house.
- BETA release can invite live load testing from early adopters.
- Crowdsourcing



What does a mobile tester need?

- **Test management**

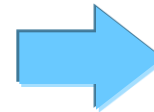
- Planning, tracking, etc.
- Manual testing capabilities
- Integration into broader collaborative lifecycle



Rational Quality Manager (RQM)

- **Automation**

- Device agnostic test cases
- Multi-target test cases
- Interaction with device-specific capabilities
- Virtualization of middle-tier and back-end systems



Third-Party automated mobile testing integrated with RQM. Green Hat Virtualization.

- **Access to a diverse set of devices**

- Platforms
- Manufacturers
- Form factors
- Carriers



Third-party mobile device-cloud services

Test environment optimization

- ✓ Generate an optimized set of device characteristic combinations (screen size, memory, CPU, etc.)
- ✓ Reduce test device possibilities from thousands to dozens

Generate Test Case Execution Records:

Step1 > Step2 > Step3

Generate Test Case Execution Records: CVR Test

Overview
Specify the attributes for the generated Test Case Execution Record(s).

Originator: Charles Rankin
 Test Plan: Unassigned
 Iteration: Unassigned
 Owner: Unassigned

Environment ?

Reuse Existing Test Environments | **Generate Test Environments**

Select one or several attributes from the columns below. The number of Test (environments) will be determined by the level of coverage that you choose, and the Advanced Properties (attribute inclusions, Exclusions, and Weightings).

Browsers	Test Adapter	Database
<input type="checkbox"/> Firefox 3.0 <input type="checkbox"/> Firefox 4 <input type="checkbox"/> Firefox 5 <input checked="" type="checkbox"/> Internet Explorer <input type="checkbox"/> Internet Explorer 6 <input type="checkbox"/> Internet Explorer 7	<input type="checkbox"/> ALL <input type="checkbox"/> HP LoadRunner <input type="checkbox"/> HP Quick Test Professional <input type="checkbox"/> JUnit Selenium <input type="checkbox"/> Rational AppScan Tester Edition <input type="checkbox"/> Rational Functional Tester	<input type="checkbox"/> Oracle 9i <input checked="" type="checkbox"/> SQL Server <input type="checkbox"/> SQL Server 2000 <input type="checkbox"/> SQL Server 2005 <input type="checkbox"/> SQL Server 2008
Application Server	Management Agent	Operating System
<input type="checkbox"/> Tomcat 7.0 <input checked="" type="checkbox"/> WAS <input type="checkbox"/> WAS 6.0 <input type="checkbox"/> WAS 6.1 <input type="checkbox"/> WAS 7.0	<input type="checkbox"/> ALL <input type="checkbox"/> Build Forge Agent <input type="checkbox"/> STAF/STAX <input type="checkbox"/> Tivoli Application Dependency Discovery Manager <input type="checkbox"/> Tivoli Provisioning Manager	<input type="checkbox"/> VMWare ESX Server 2.5 <input type="checkbox"/> VMWare ESX Server 3.0 <input type="checkbox"/> VMWare ESX Server 3.5 <input type="checkbox"/> Vista <input checked="" type="checkbox"/> Windows <input type="checkbox"/> Windows 2000 Advanced Server
CPU		
<input type="checkbox"/> ALL <input type="checkbox"/> AMD64 <input type="checkbox"/> Alpha <input type="checkbox"/> EM64T <input type="checkbox"/> IA64 <input type="checkbox"/> PA-RISC		

[Advanced Properties](#)

Generation Properties ?

Coverage:

- Minimal - One-way interaction
- Medium - Pair-wise interaction
- Large - Three-wise interaction
- All - All permutations

Medium - Pair-wise interaction

Test integrated fully into the process

- ✓ No wasted effort – development is qualitative and aligned to the agreed upon requirements at the right time
- ✓ Quality Assured – test teams know exactly what requirements and functionality have and haven't been tested
- ✓ Whole team buy-in improves team trust, efficiency and focus

Developers understand the business needs and test effort

JKE Banking

1: JKE Banking Release 1

Name State
1: JKE Banking Release 1 Approved

Description
Defines testing for the Release 1 of JKE Banking. Significant capabilities being added in Release 1 include the Money that Matters initiative whereby account holders can elect to redirect dividends from their investments to causes needing assistance.

Details
Originator: tammy Updated: 9/21/10



JKE Banking Release 1
Test Plan Overview | Snapshots | History

Originator: Tammy Owner: Tammy Action: Select Action State: Approved

Description: Defines testing for the Release 1 of JKE Banking. Significant capabilities being added in Release 1 include the Money that Matters initiative whereby account holders can elect to redirect dividends from their investments to causes needing assistance.

Development Plan Links
Linked development plans

- Release 1.0 Backlog [Release 1.0]
Owner: JKE Banking
Iteration: Release 1.0 (9/6/10 - 10/17/10)
42/78 pts
100%



Testers understand Sprint Plans and business expectations

49: Release 1 Planning

Showing 18 Artifacts

ID	Name	Artifact Type	Last Modified By	Last Modified Date
5	Allocate dividends by amount and frequency	User Story Elaboration	cp	Sep 20, 2010 8:39:57 PM
8	Dividend allocated by percentage			
11	Customers can nominate an organization for the program			
12	Donor Dividend Allocation Criteria			
13	Donors Depos Money in a Pooled Assistance Fund			

Release 1.0 Backlog [Release 1.0]

Owner: JKE Banking
Iteration: Release 1.0 (9/6/10 - 10/17/10)
42/78 pts
100%

Requirements owners clarify the business needs in a collection



Challenge 3: Integrating with existing systems

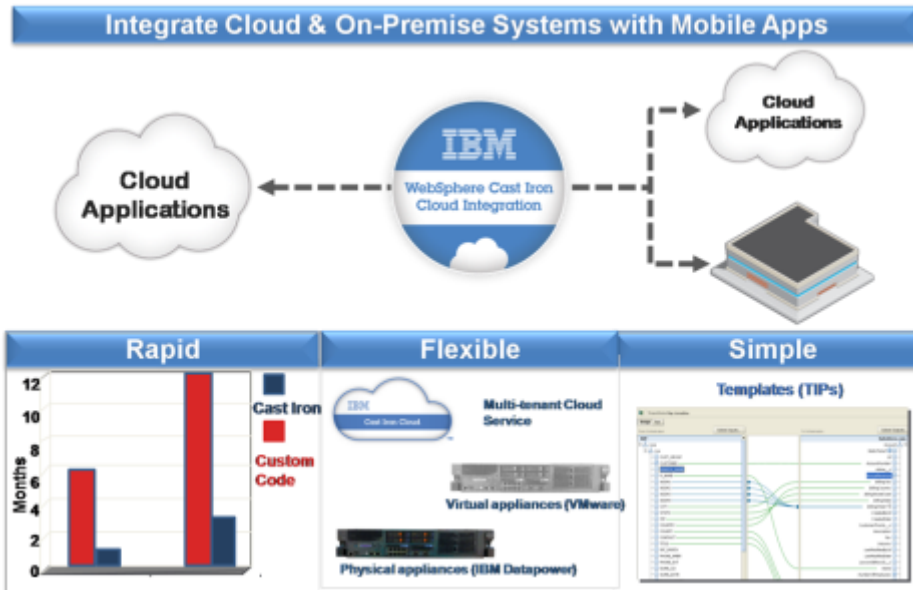


- Mobile applications need to connect to enterprise back-end data and services
- Existing programs and services may need to be modified for the mobile app
- Multiple skills will be involved, responsible for different parts of the mobile application

Rapid, simple & flexible connectivity for mobile apps

WebSphere Cast Iron Cloud integration

Simple and flexible integration for all connectivity projects, allowing you to rapidly integrate SaaS and back-end systems with mobile apps



Client Challenge

Simplified and cost effective mobile integration to back-end systems and cloud

Key Capabilities

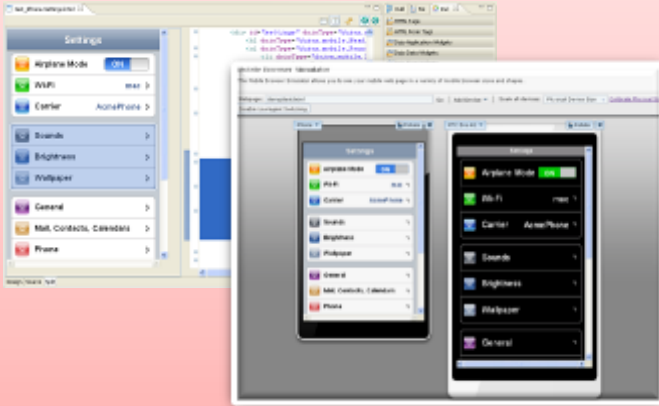
- Native connectors and template integration processes to connect mobile apps to backend & cloud systems
- Bidirectional connectivity and business logic
- Centralized monitoring
- Simple and flexible, user-friendly, wizard-based, “configuration, not coding” architecture

IBM Integrated Development Environments with IBM

Worklight

Extend existing back-end services and data to mobile apps

Integrated multi-platform development environments



Construct, debug, and test mobile user interfaces



Refactor and extend existing logic on enterprise platforms (System z, Power) as mobile-consumable services

Challenge 4: Meeting tight time-to-market requirements

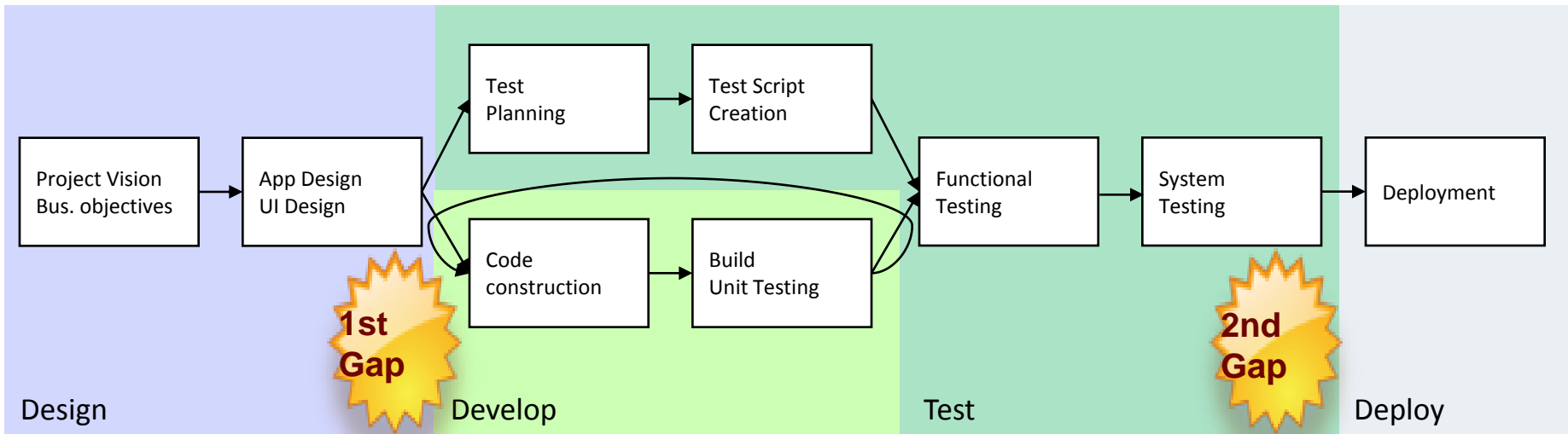
Mobile is pushing traditional delivery approaches to the breaking point



	Mobile Apps	Desktop Apps
Time-to-market	Weeks to Months	Months to Years
Frequency of updates	Once every several weeks	12-18 month cycles

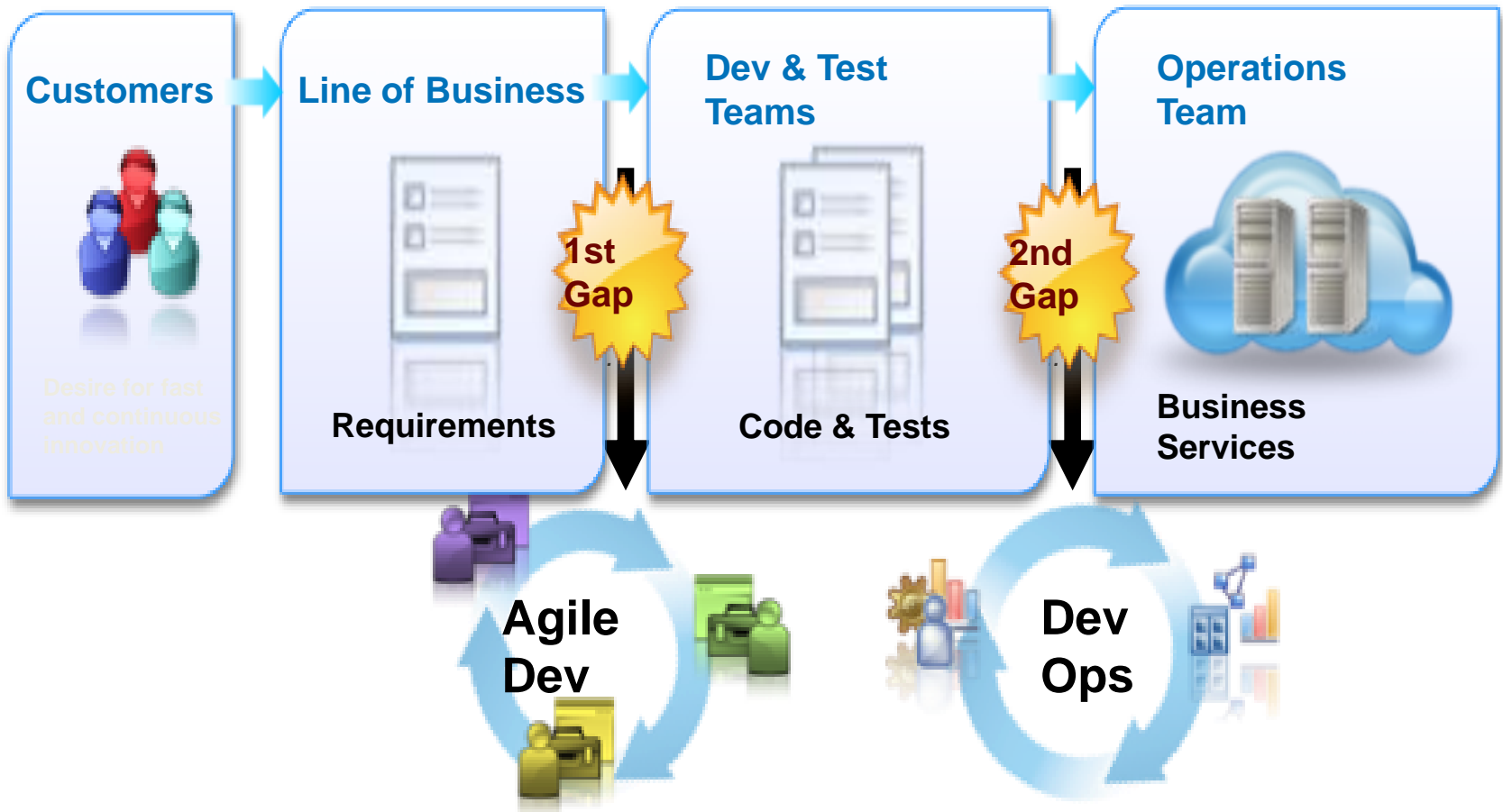


Factors affecting project velocity



- There are 2 key factors affecting project velocity:
- **Gap 1:** amongst Line of Business, Development and Test teams (ALM)
 - Late rework due to misalignment of stakeholders
 - Slow progress due to hand-off errors and delays between team roles
- **Gap 2:** between Development/Test and Operations Team (DevOps)
 - Slow cycle/iteration times due to DevOps challenges

Addressing application lifecycle management gaps



Hybrid development requires multiple tools and skillsets



Objective-C

Native Code for iOS



Java

Native Code for Android



HTML/JS

Application Code (Shared)

IMF + Rational CLM allows teams of specialists to collaborate

Development of Native Portions



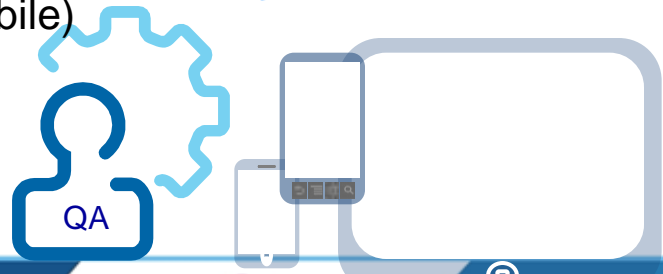
Development of Web Portions



Worklight Builder

Rational Team Concert

Testing of Official Build (Local or PerfectoMobile)

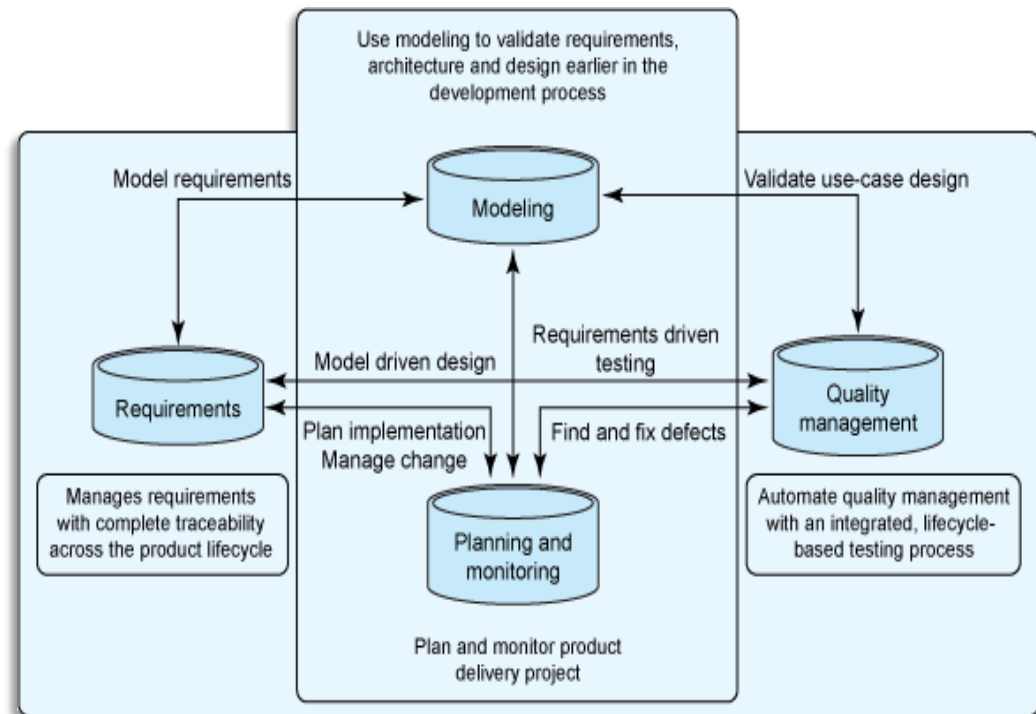


Back to the future!



Model Driven Development of Mobile Applications

- Early design and architecture of application, consumable by developers.
- Visualize the system with semantic information.
- Connect software development to requirements, testing and planning.
- Quickly create a visual representation of an app to communicate its value to multiple stakeholders.



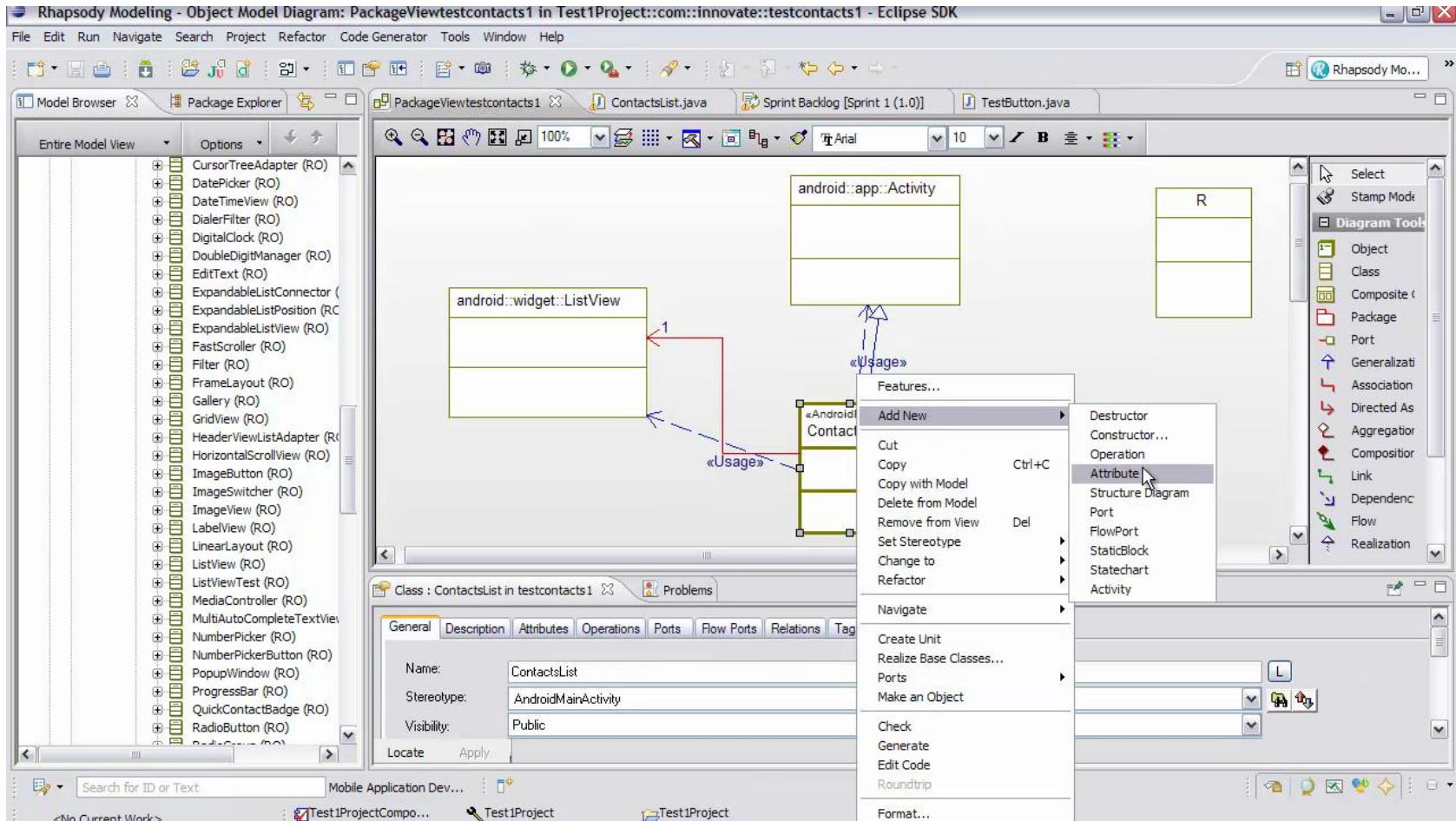
IBM Rational Rhapsody enables MDD of Android applications

Visualizing an android app as UML diagrams

The screenshot displays the Rhapsody Modeling environment with the following components:

- Package Explorer:** Shows the project structure for 'NotesList', including packages like 'com.example.android.notespad' and 'com.google.provider', and classes like 'NotePadProvider', 'TitleEditor', 'NotesLiveFolder', and 'NoteEditor'.
- Model Browser:** Provides a hierarchical view of the model, showing components, packages, and classes.
- UML Class Diagram:**
 - android.content.ContentProvider:** Base class with methods like `ContentProvider()`, `ContentProvider(context Context)`, `getContext() Context`, and `getUriForPath(String path)`.
 - android.app.Activity:** Base class with constants like `RESULT_CANCELED`, `RESULT_OK`, `RESULT_FIRST_USER`, and `FOCUSED_STATE_SET`.
 - android.net.Uri:** Base class with constants like `EMPTY` and `CREATOR`, and methods like `isOpaque()`, `isRelative()`, `isAbsolute()`, `getScheme()`, `getSchemeSpecificPart()`, and `getEncodedSchemeSpecificPart()`.
 - Derived Classes:**
 - `«AndroidContentProvider» NotePadProvider` (Generalization)
 - `«AndroidActivity» TitleEditor` (Generalization)
 - `«AndroidActivity» NotesLiveFolder` (Generalization)
 - `«AndroidActivity» NoteEditor` (Generalization)
 - `«AndroidManifestActivity» NOTESLIST` (Generalization)
 - Usage Relationships:**
 - `«Usage»` from `NotesLiveFolder` to `android.net.Uri`.
 - `«Usage»` from `NotesLiveFolder` to `android.app.Activity`.
 - `«Usage»` from `NoteEditor` to `android.net.Uri`.
 - `«Usage»` from `NoteEditor` to `android.app.Activity`.
- Properties Window:** Shows details for the selected class, including Name, Stereotype, and Default Package.

Round-trip code generation



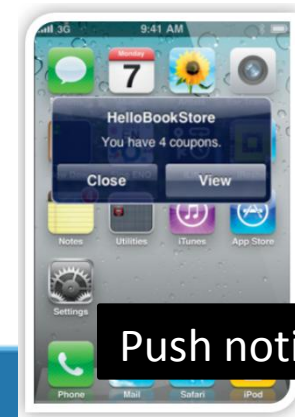
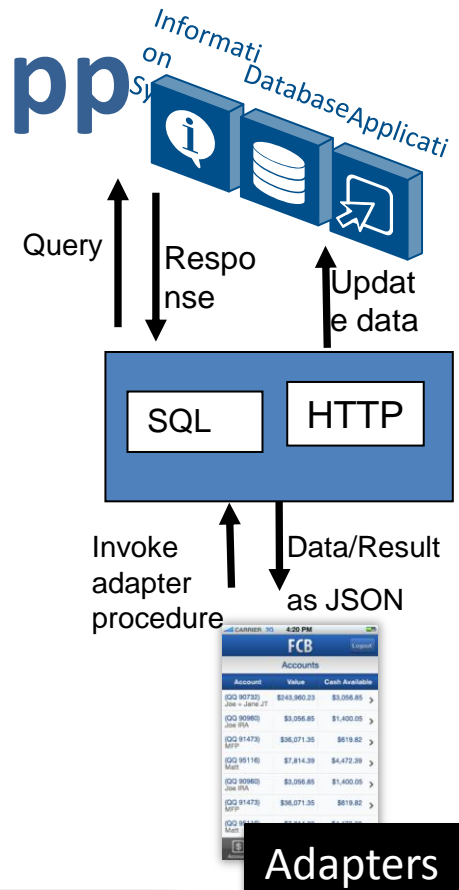
Debugging at the model level

The image displays a multi-pane view of an Eclipse IDE used for developing and debugging an Android application. The application is a stopwatch.

- Left Pane (Sequence Diagram):** Shows the interaction between the environment (ENV), the Stopwatch class, and the StopwatchActivity class. Key messages include 'Create()', 'reset()', 'show()', 'Create()', 'onCreate(savedInstanceState = null)', 'addEvent()', 'pressed(event = 0)', and 'evStartStop()'.
- Center Pane (State Machine Diagram):** Illustrates the state transitions of the stopwatch. States include 'Active', 'Stopped', 'Started', and 'Running'. Transitions are triggered by events like 'evReset', 'evStartStop', and 'evLap'. The 'Started' state contains a sub-diagram for timing logic with 'Tic' and 'Toc' states and a 50ms interval.
- Right Pane (Application Preview):** Shows the actual Android application running on a yellow stopwatch device. The screen displays 'Stopwatch' at the top, 'Start/Stop', 'Lap', and 'Reset' buttons, and a digital display showing '00:00'.

Monetize your app

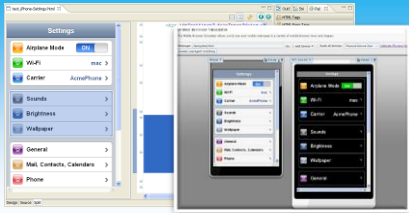
- Use SDKs of ad networks such as Google Admob, InMobi, etc.
- Leverage tools such as IBM Worklight to incorporate ads into the common code of your app and deploy to all platforms.
- For in-house advertising or ad mashups, use Worklight adapters.
- Use push notifications to send advertisements, coupons, etc. in a controlled fashion.



Push notifications

Aligning teams across the development lifecycle

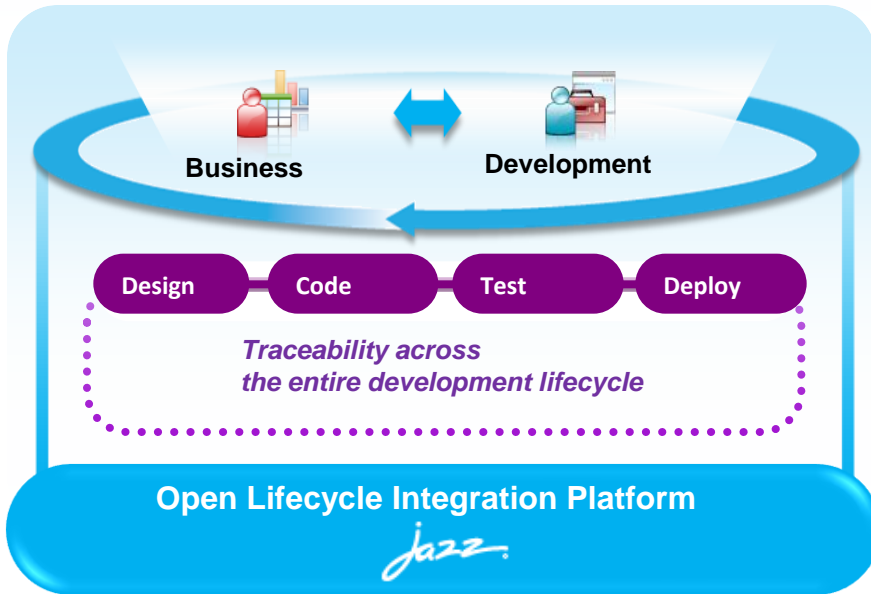
Collaborative Lifecycle Management, Rational IDEs, and Worklight



Construct, debug, and test mobile UIs



Refactor and extend existing logic as mobile-consumable services



End-to-end Lifecycle Management for Mobile Application Development

Client Challenge

Mobile apps are typically multi-tiered and require collaboration between multiple teams, including teams responsible for design, development, test, and deployment

Key Capabilities

- Common, integrated tool set across all phases of development and components of the mobile solution
- Integration with Worklight Studio to ensure developers have access to plans, tasks, builds, and code from within their development IDE
- Traceability across the entire mobile application development lifecycle – all teams are aware of changes (for example, a changing requirement)

QUESTIONS

<http://www-01.ibm.com/software/solutions/mobile-enterprise/>

<http://www.ibm.com/software/rational/mobile/>

Twitter @ayushman_jain



© Copyright IBM Corporation 2012. 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.