



IBM Software Development Technical Conference

Automating Software Release Management with IBM Rational ClearCase and IBM Rational ClearQuest

Kevin Lee – Technical Consultant, IBM Rational Software
kevin.lee@uk.ibm.com



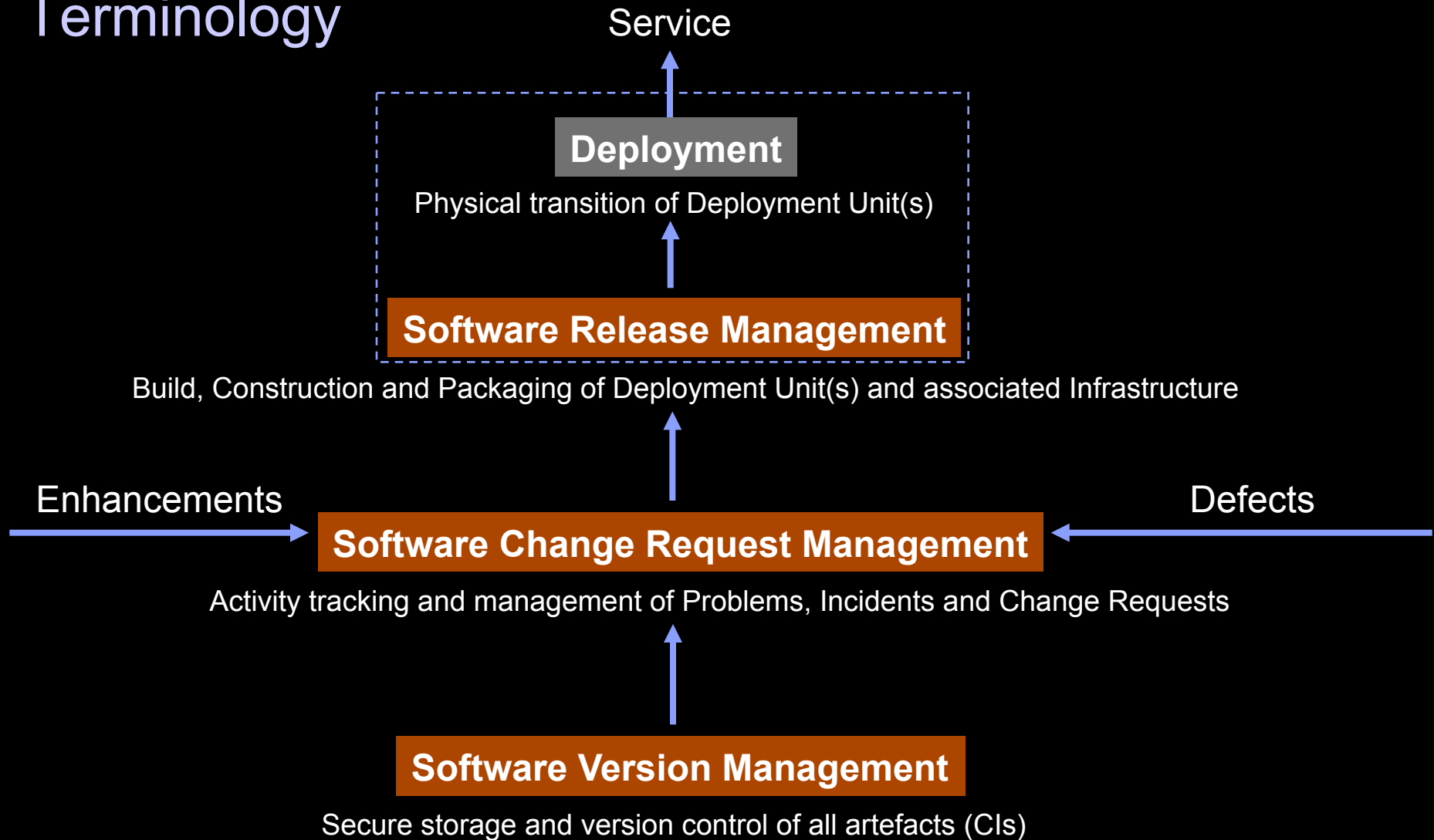
@business on demand.

Agenda

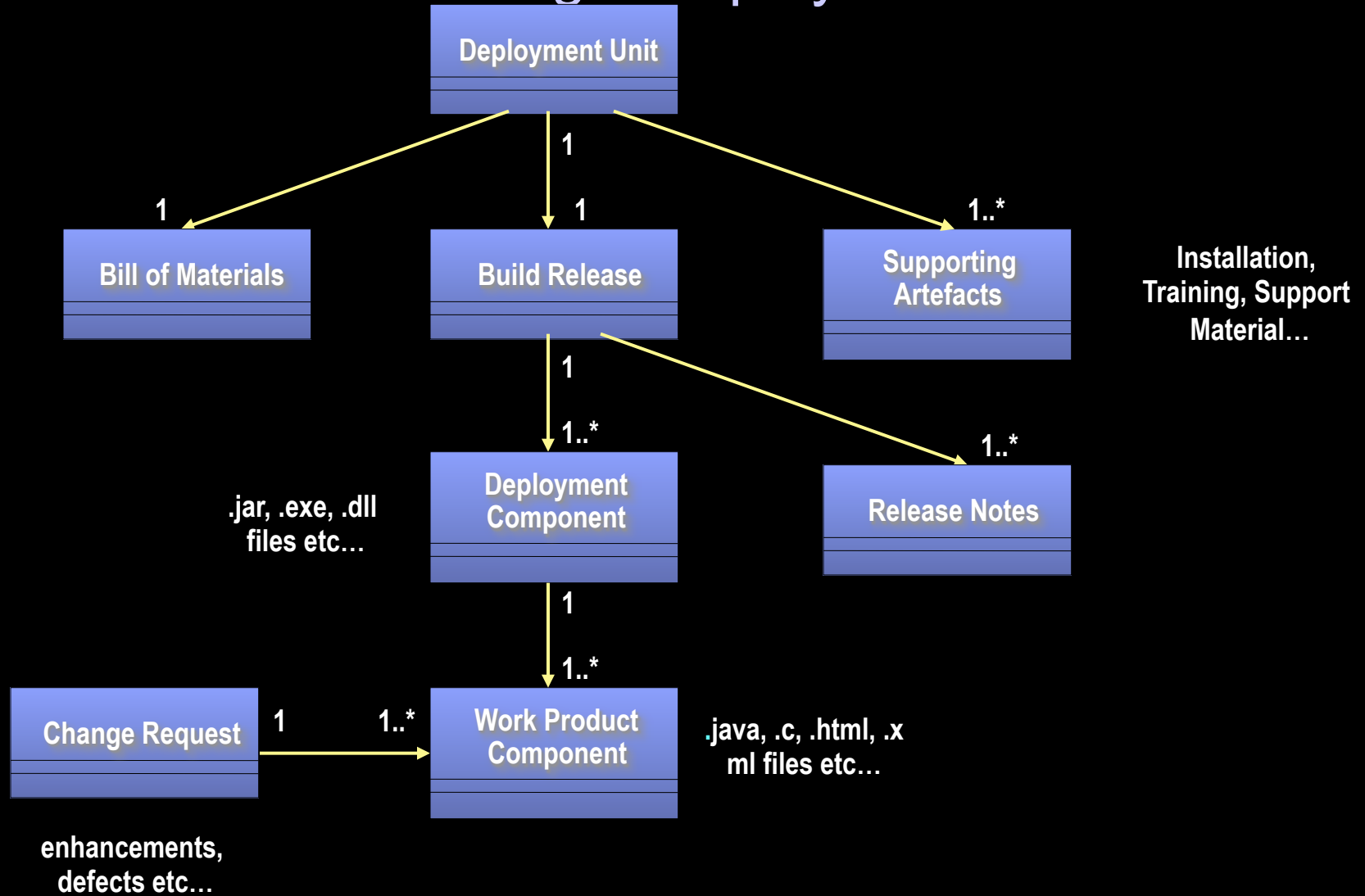
- Terminology
- Implementing a Release Record in ClearQuest
- Referencing the Release Record
- Release Actions and States
- Automating Packaging
- Deployment
- Completing the Release
- Demo



Terminology

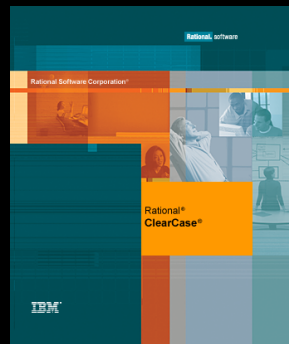


Software Release = Single Deployment Unit



Manage Change to Manage Releases

- *If change is managed during the software development lifecycle, so that the creation or update of Configuration Items is always associated with an appropriate Change Request, then Software Release Management can be automated.*
- This can only be achieved with tool support...



Implementing a Release Record in ClearQuest

- Create a stateful record to encapsulate “release lifecycle”
- Recommended minimum set of fields:
 - *Name* [SHORT_STRING]
 - *Description* [MULTILINE_STRING]
 - *Manager* [REFERENCE] to users
 - *UCM_project* [REFERENCE] to UCM_Project
 - *UCM_baseline* [SHORT_STRING]
- Applicable to either Base ClearCase or UCM



Implementing a Release Record cont...

- Populate *UCM_baseline* automatically with a list of ClearCase baselines using a choice entry list hook:

```
Dim objClearTool
Dim strProject, strStreams, arrStreamList, strBLs, arrBLList
Dim i, j
Set objClearTool = CreateObject("ClearCase.Cleartool")
strProject = GetFieldValue("UCM_Project").GetValue
If (strProject <> "") Then
    ' get streams in our project first
    strStreams = objClearTool("lsstream -short -in " & strProject & \
        "@\VOBNAME")
    arrStreamList = Split(strStreams, vbCrLf)
    For i = 0 to (UBound(arrStreamList)-1)
        ' get all the baselines in the stream
        strBLs = objClearTool("lsbl -short -stream " & \
            arrStreamList(i) & "@\VOBNAME")
        arrBLList = Split(strBLs, vbCrLf)
        For j = 0 to (UBound(arrBLList)-1)
            ' strip out "deliver" baselines
            If (Left(arrBLList(j), 9) <> "deliverbl") Then
                choices.AddItem(arrBLList(j))
            End If
        Next j
    Next i
Next
End If
```

Example ClearQuest Release Record

Submit Release RTST00000039

Main

id:	RTST00000039	State:	Submitted
Name:	Release 1.0	Manager:	kevin
UCM project:	RatllBank_Rel_1	UCM Baseline:	RIB_inter_04_09_200

Description:

First Release of System

OK

Cancel

Values ▾

Referencing the Release

- Three ways to reference a release:
 1. as the release where a defect was found (**Found_In Release**)
 2. as the intended release for the request (**Targetted_At Release**)
 3. as the actual release for the request (**Delivered_In Release**)
- As we are using a stateful record, need to create shadow choice lists for the release names (otherwise we would just see the ClearCase record ID in the pull-down lists); for example on a defect record:
 - *Found_In_Rel* [REFERENCE] to Release
 - *Found_In_Shadow* [SHORT STRING]
 - *Targetted_At_Rel* [REFERENCE] to Release
 - *Targetted_At_Shadow* [SHORT STRING]
 - *Delivered_In_Rel* [REFERENCE] to Release
 - *Delivered_In_Shadow* [SHORT STRING]



Referencing the Release Record cont...

Shadow
fields

Submit ChangeRequest RTST00000058

Main Analysis Tasks Unified Change Management

Priority: High

Request Type: Defect

Product: Product A

Severity: Critical

Found In Release: Release 1 ID: RTST00000002

Targetted At Release: Release 1.1 ID: RTST00000003

Delivered In Release: ID:

Owner: kevin

OK

Cancel

Values

Release
Record IDs
(normally
removed from
form)

Referencing the Release Record cont...

- Choice-list hook to populate the shadow choice list with the release name:

```
Dim objCQSession, objCQQuery, objCQResultSet
Dim lngStatus
Set objCQSession = GetSession
' execute query to get names of all releases
Set objCQQuery = objCQSession.BuildQuery("Release")
objCQQuery.BuildField("Name")
Set objCQResultSet = objCQSession.BuildResultSet(objCQQuery)
objCQResultSet.Execute
' move through result set and populate list
lngStatus = objCQResultSet.MoveNext
Do While (lngStatus = AD_SUCCESS)
    choices.AddItem(objCQResultSet.GetColumnValue(1))
    lngStatus = objCQResultSet.MoveNext
Loop
```

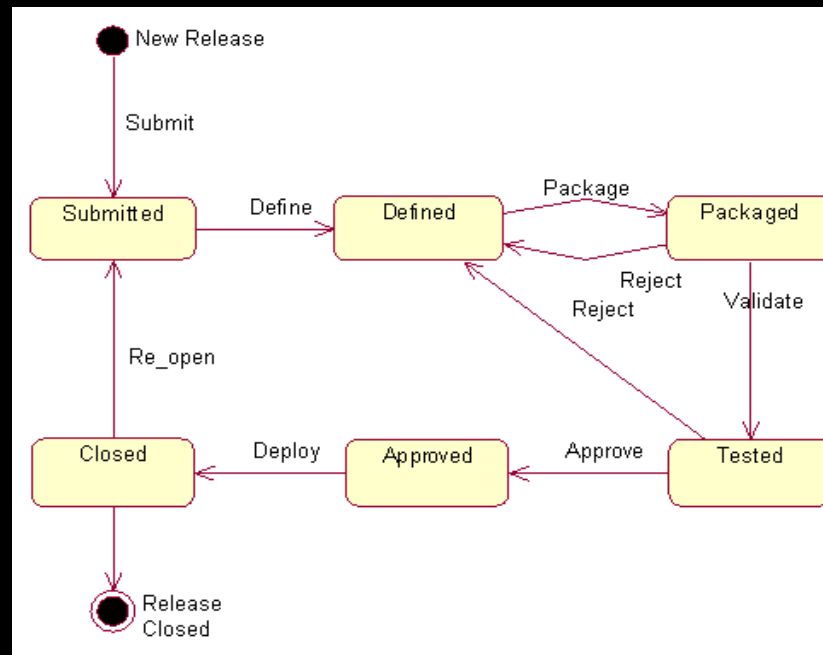
Referencing the Release Record cont...

- Value changed hook for each of the shadow fields, which updates the original reference list (use commit hook in production use)

```
Dim objCQSession, objCQQuery, objCQFilter, objCQResultSet
Dim strName
Set objCQSession = GetSession()
' execute query to get our release based on the name
strName = GetFieldValue("Found_In_Shadow").GetValue()
Set objCQQuery = objCQSession.BuildQuery("Release")
objCQQuery.BuildField("id")
Set objCQFilter = objCQQuery.BuildFilterOperator(AD_BOOL_OP_AND)
objCQFilter.BuildFilter "Name", AD_COMP_OP_EQ, strName
Set objCQResultSet = objCQSession.BuildResultSet(objCQQuery)
objCQResultSet.Execute
' inset the ID of the record into the shadow field
If (objCQResultSet.MoveNext = AD_SUCCESS) Then
    SetFieldValue "Found_In_Rel", objCQResultSet.GetColumnValue(1)
End If
```

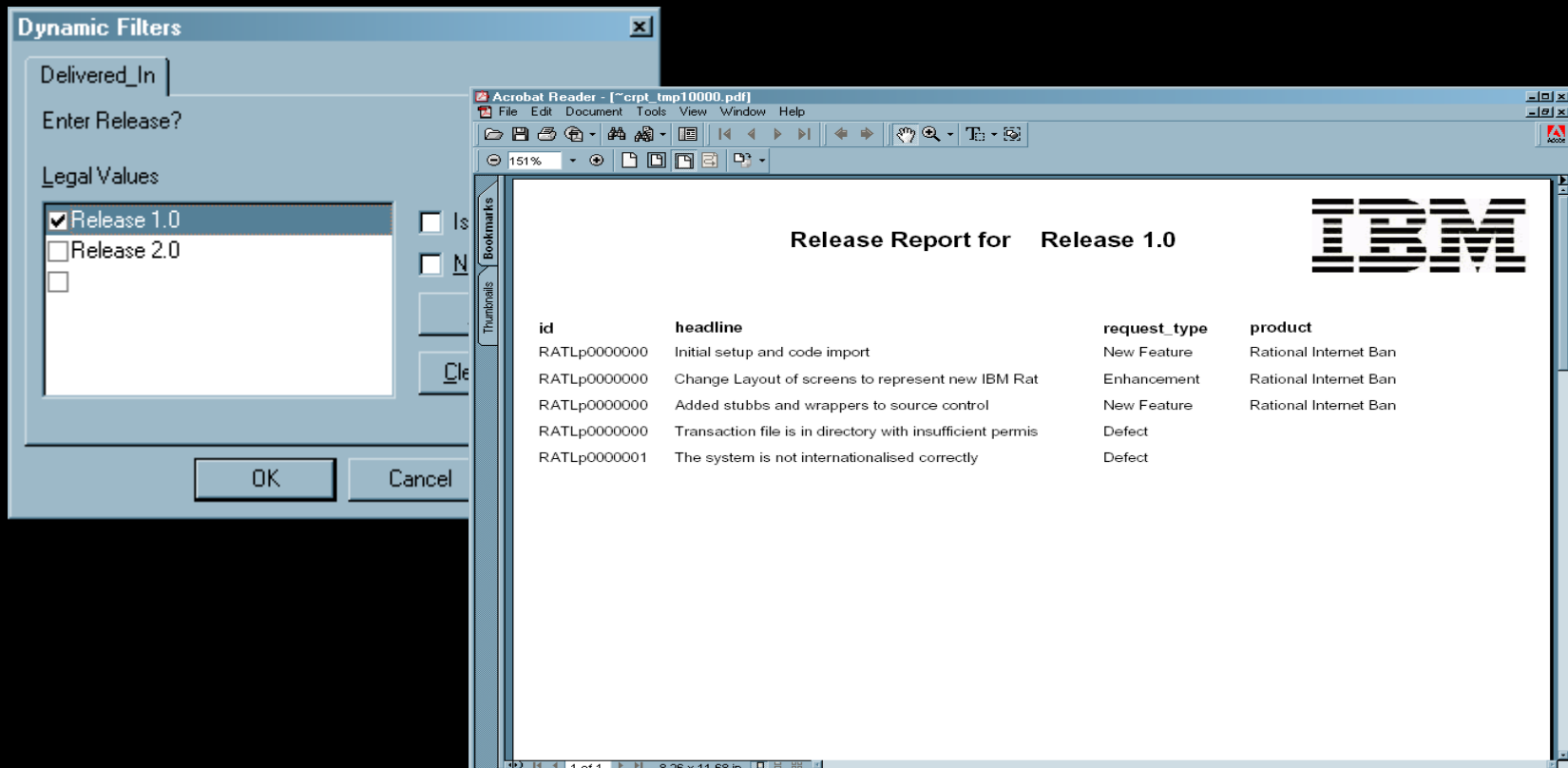
Release Actions and States

- Create an appropriate state transition matrix for the release record
- Use the lifecycle to enforce actions on child tasks, i.e. don't allow packaging until all change requests have been tested...



Generating Release Notes

- Generate Dynamic Release Reports

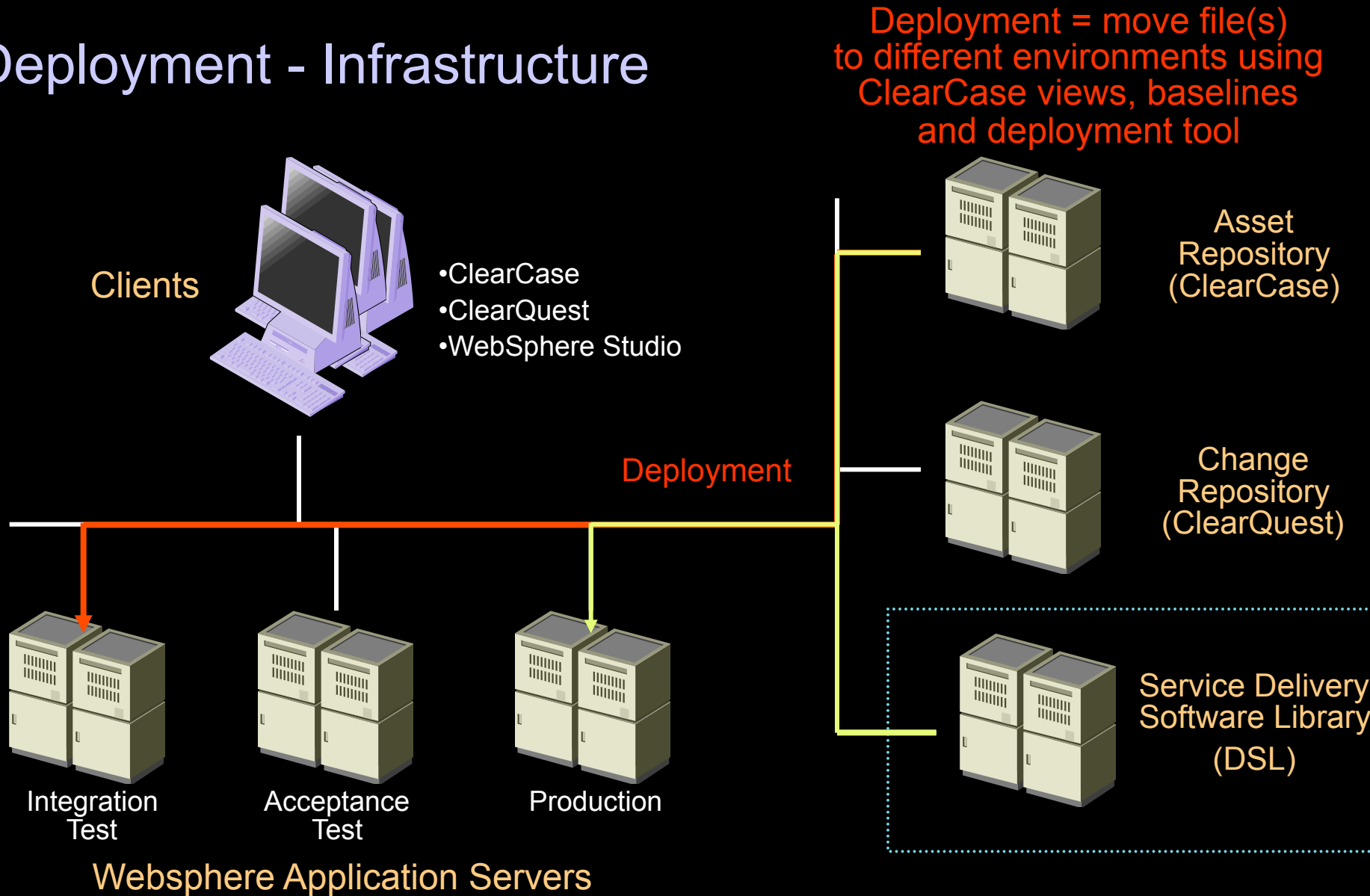


Automating Packaging

- Use the ClearCase Baseline from Release Record to construct a read-only view then present to a deployment tool:

```
Dim objClearTool
Dim strRelName, strMyRel, strProject, strBaseline, lngStatus
Set objClearTool = CreateObject("ClearCase.Cleartool")
strRelName = GetFieldValue("Name").GetValue
' lowercase and replace spaces in the release name
strMyRel = LCase(Replace(strRelName, " ", "_"))
strProject = GetFieldValue("UCM_Project").GetValue
strBaseline = GetFieldValue("UCM_Baseline").GetValue
If (strProject <> "" and strBaseline <> "") Then
    lngStatus = objClearTool("mkstream -in " & strProject & \
"@\VOBNAME -baseline " & strBaseline & "@\VOBNAME -readonly \
" & strProject & "_" & strMyRel & "@\VOBNAME")
    lngStatus = objClearTool("mkview -stream " & strProject & \
 "_" & strMyRel & "@\VOBNAME -tag " & strProject & "_" & \
strMyRel & "_view -stgloc -auto")
    ' package up files using view just created
End If
```

Deployment - Infrastructure



Deployment

- Deployment mechanism usually depends on the technology being used, but essentially two types:
 - ▶ Data Moving - simple point-to-point basis (i.e. servers)
 - ▶ Distribution - complex package to (multiple) endpoints
- Tivoli Data Moving Service, example command line:

```
wspmvdata -s @source -t @destination -r \  
  tpost:c:\temp\bounce_webserver.bat -P \  
  sp:"C:\Views\alex_Rel3_int\Classics\html" -P \  
  tp:"C:\Program Files\IBM HTTP Server\htdocs" -R *.*
```



Deployment cont...

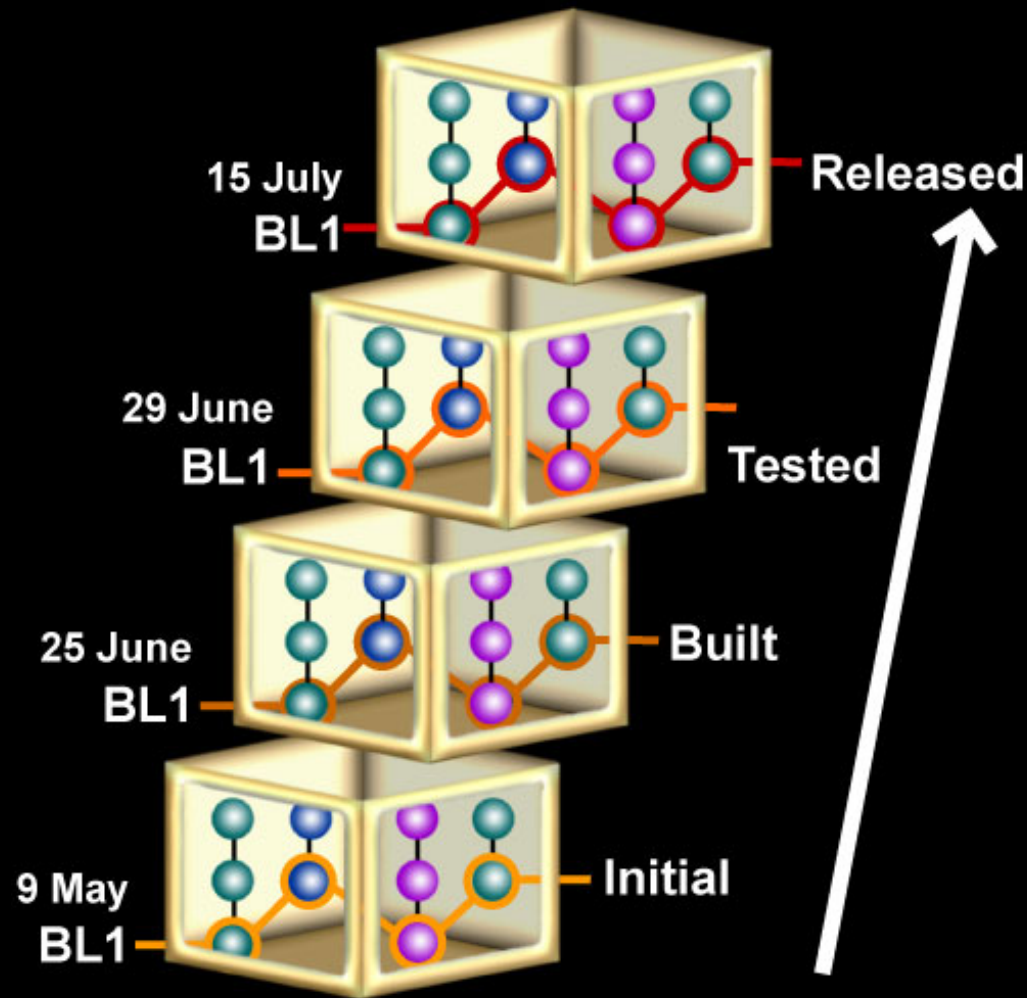
- Add Deployment tab onto Release Record

The screenshot shows a Windows-style dialog box titled "Submit Release TST00000002". It has two tabs: "Main" and "Deployment", with "Deployment" currently selected. The dialog contains the following fields and controls:

- Source_host:** A dropdown menu with "conman" selected.
- Endpoints:** A list box showing three items: "conadm", "conadm_ep2", and "conadm_ep3".
- Pre_install_script:** An empty text input field.
- Post_install_script:** A text input field containing the path "c:\temp\bounce_server.bat".
- Buttons:** "OK", "Cancel", and "Values" (with a dropdown arrow) are located on the right side of the dialog.

Deployment – ClearCase Promotion Levels

- Baselines indicate related versions
- Baselines can be promoted to indicate what environment they have progressed to:
 - ▶ Integration Test
 - ▶ Acceptance Test
 - ▶ Production
- Mimics traditional mainframe promotion



Deployment cont...

- Use the Action hooks to change the ClearCase baseline promotion levels, e.g. when the “deploy” action was selected promote the baseline to RELEASED:

```
Dim objClearTool
Dim strBaseline, lngStatus
Set objClearTool = CreateObject("ClearCase.Cleartool")
' get the release baseline and promote it to RELEASED
strBaseline = GetFieldValue("UCM_Baseline").GetValue
If (strBaseline <> "") Then
    lngStatus = objClearTool("chbl -nc -level RELEASED " & \
        strBaseline & "@\VOBNAME")
End If
```

Completing the Release = Single Deployment Unit

- We have already shown how we can generate the release note from within ClearQuest.
- Can add additional fields on the release record that we must complete before the release can be deployed.
- For separate documents, such as Installation Documents we can either attach (via the ClearQuest attachments capability) or reference.
- We can make such fields mandatory so that the release can't be approved until they are included.

Deployment Unit





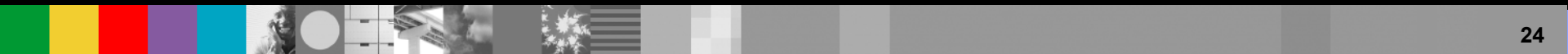
Summary

- Use ClearQuest to create Release record and associate with Change Requests
- Use ClearCase to manage the baselines for the Release
- Implement your own Release process in the tools
- Link to a deployment tool to transition the release

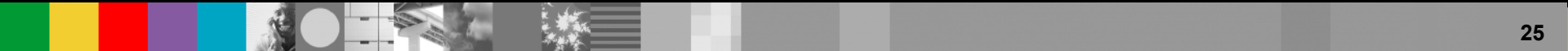


References

- developerWorks article (this presentation in more detail)
http://www-106.ibm.com/developerworks/edu/i-dw-r-automatesrm-i.html?S_TACT=104AHW18&S_CMP=ZHP
- A primer on ClearQuest integrations
<http://www-106.ibm.com/developerworks/rational/library/1051.html>



Questions



Thank
YOU



Backup – Release Best Practices

1. Have regular, targeted Release dates
2. Always have a tested back-out plan
3. Have a documented Release Management process, describing roles, activities and artefacts
4. Construct Deployment Units as early as possible
5. All Releases should be built by an independent team
6. All Deployments should be performed by a team independent of the Development team
7. Test the Deployment process at least once before deploying to Live
8. Automate as much as possible – use integrated tools for Configuration, Change Management and Deployment
9. A mature Software Configuration Management process and tool is required to support the development of multiple releases in parallel
10. The Deployment Unit does not just mean software, it includes hardware and supporting documentation

