Software Deployment and Evolution

Deployment Data Collection Question Set

This document provides a starting point for the type of information that needs to be collected to analyse the deployment process. This information is presented in a question format. Once answered, you can create a table for presentation and analysis. If you are not able to answer certain questions -- just indicate that and move forward.

Note: *Teams can expand and add more questions or items that need data collection for different deployment activities.*

Software Deployment Model - Certain clarifications

The activities captured by the deployment model outline the key activities that are undertaken as part of a software deployment. This model does not or require that certain activities be performed by a tool. Some activities may be ideal for automation, but many are better undertaken manually. For instance, installation instructions that indicate how to transfer the files may be provided in a text file for a human administrator to use (or) encoded into the meta-data of a package for a tool to use. The only difference is in how this instruction is processed and executed. The model abstracts out the executor -- that is it treats a computer program transferring files and a human using a set of instructions equally.

When you are assessing a products deployment, look for the information, not the fact that it is tool based. Meta-data within a package can be provided in any format. A simple text file with instructions for a developer (human) can still be considered to be meta-data. Similarly, an XML or binary file with meta-data information targeted at a computer program would also constitute as meta-data. When answering the questions, if the information is provided (in some form) then the deployment process satisfies the models expectation. You can however, indicate that the information provided is vague, insufficient, assumes background knowledge or it is inconsistent.

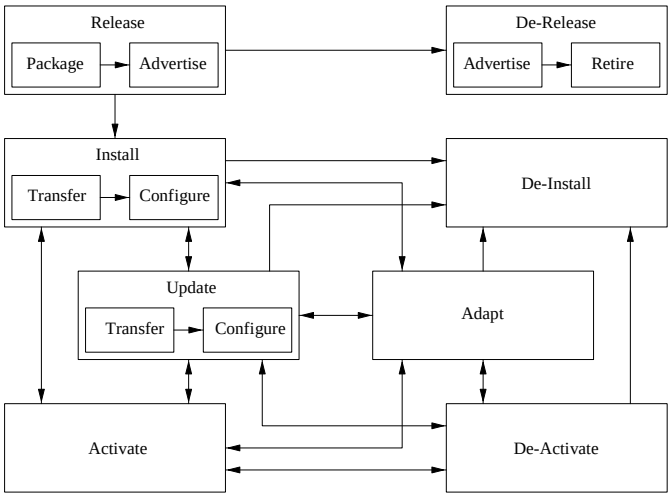

Figure 1: Activities of the Software Deployment Process.

Product information

1. Product Name, Version and Brief Description and Web site
2. Does this product require a multi-tiered setup? or Is the product targeted for a single machine?

Release - Package

3. What is the container format for package
4. Are there multiple environments for this product? Can this information be obtained easily?
5. Is there any use of virtualization for this product?
6. Does the package format used contain meta-data?
7. What is the meta-data that is embedded with in this package?
8. What is inside the package?
9. Does the meta-data contain identification information (like provided for RPM)?
10. Does package contain information on dependencies?
11. Does the package contain components that need to be copied into directories controlled by operating system (or) directories that require additional privileges?
12. Does package contain information on the level of privileges required to install and activate this software system?
13. Does the packaging format used require an additional installation tool (e.g. Windows Installer, RPM installer)?
14. Is the installation instruction provided within the package for humans to read and action?
15. Are there any likely design implications because of the package format chosen?

Release - Advertise

16. What is the license type that is used? Is the license timed or restricted by time?
17. Is it an open-source license or a commercial one?
18. If an open-source license is used, what is the type (Highly restrictive, Restrictive or Open)?
19. What are the support services provided for this product? (FAQ, Communication methods, Defect tracking, Issue logging/management, discussion boards, crash reporting, web site, user manuals, user documentation, installation guide, getting started guide etc..)
20. Does the product have a published roadmap?
21. Can you contact the support staff via email, telephone, Instant messaging?
22. Does the product have a blog? Do they provide a Twitter feed?
23. Do any of the support services offered embedded into the product and hence have been designed to be part of it?)
24. Is the product offered for distribution via multiple methods? (Internet, CD, ISO -- that is a higher-level container)

Install

25. Is installation a manual or semi-manual or fully automated process (i.e. double-click an executable and forget about it)?
26. Does the product require or assume or bundle a tool for the installation?
27. What is the tool (if it can be identified) that was used for creating the installation package?
28. Does the installation require a external tool (e.g. Windows Installer, RPM) that will read the package meta-data and action on the tasks? If so, what is the tool that is used?

Install - Transfer

29. Does the user have control on where the files are transferred into?
30. Can the transfer take place if a previous version is currently in use?
31. Does the transfer method over-ride existing files? If it is a manual process, do the instructions provide any information on what to do in this situation?
32. Can multiple versions of this product co-exist?
33. Does the transfer process check for dependencies required before commencing the transfer?
34. Does the transfer process copy all of the libraries needed (i.e. they are bundled within the package)?
35. Do files have to be transferred to multiple machines in order for the product to be setup and configured properly?
36. Is all of the content needed transferred (or) is some information downloaded later?
37. Does the transfer method used check (or) require the user to check that all required software (at required level) exists? That is, are the instructions provided clear about the required software and versions? (e.g. Need MySQL 5.0.x or higher).
38. Does the transfer require specific authorisation and authentication?
39. Does the installer need to download additional components not provided in the package? (The installer can be a human -- that is, it need not be a software that does this task)
40. Is any file integrity information provided? For instance, is an MD5 hash provided by the developers to ensure we can check integrity?

Install - Configure

41. Once files are transferred is there a separate configuration step that needs to take place?
42. Do any components need to be registered with the operating system?
43. Do any components need specific configuration information?
44. Is all configuration information provided within the package? Is the information aimed at humans or can a tool read and process it?

45. Is the configuration achieved via an automated process? How mature is this automation? (that is, is it a script that you can execute (or) does it require manual steps?)

46. Does the configuration require setup of connectivity, mapping between servers or systems? (e.g. setting up a connection string for the database or web server or mail server or proxy server etc..)

47. Does the configuration require additional security privileges (than that for transfer)?

Update

48. Does the update take place once activated (that is, triggered by the application)?

49. Does the update process require that all of the application be de-activated? (or) Can certain parts remain activated?

50. Once updated, does it re-start the application?

51. Does the software inform user that a new update is available?

52. Are there instructions provided on how to update the software system? (Upgrades also fall under this). That is, is the instruction clear on how to update an existing software system?

53. Is the transfer supported from within the tool?

54. Do the instructions clearly indicate what is transferred and what aspects are updated?

55. What happens in case of failure during transfer?

56. Does the update migrate data?

Activate

57. Is there a difference between first time activation and normal activation?

58. Does the software setup a cache and configuration information on first activation? (Specifically important for Browsers, IDEs)

59. Does the software check runtime integrity? That is, check that all components that are needed exist -- for instance, if the application needs a web server and a database to work properly. Is there any instructions provided in the user guide on how to undertake this?

60. Does the software undertake a license verification?

61. Does the software lock (or acquire a lock) on any resources?

62. Does the software require a multi-step activation? or Has a script been provided that will activate all components.

De-Activate

63. Does the de-activation return all locks on resources properly?

64. Do locks get relinquished of the software crashes? (A lock may not be returned, only if the fact that a particular file has been locked is stored in a persistent data store. Typically achieved by writing a lock file).

65. Does the software support/permit partial de-activation?

Adapt

66. If a library of external database that is required by the software has been removed/modified -- how does the software react? Does it show a useful error message?

De-Install

67. Does de-installing it perform a backup? Can this backup be used to restore product to the state before removal? Is this automated or tool supported?
68. Is an uninstall script/application/instructions provided?
69. Does the uninstall unregister any components?
70. Does the removal require removing software from multiple machines?

Transactional Integrity

71. Is there a log of the installation process available?
72. If installation is manual (or semi-manual), should the person executing the install write a log?
73. Is is possible to recover from failure of deployment?

Data Migration

74. Does the software require any data migration upon update?
75. Does it provide a method to import and export data into a format for later use? Is the data format documented?
76. What is the data migration (if required) model?
    a. Direct database import from older schema to new one. (How are NULL values addressed in this approach? What is the strategy to handle new columns if data is relational?)
    b. Data exported by old systems, and imported by an API in the new system. (This is considered safe, but what is the process to handle errors?)
77. What is the database migration trigger?
    a. Data is imported when user signs into the new system for the first time. If so, what is the handling approach if the import fails?
    b. System is shut down for all users and data is imported into new, then opened for access to all users at the same time. What is the handling process in case of failure?

Design Implications

78. Are there any design implications arising because of deployment choices? (Such as packing format, support, installation process or update process etc.)

General

79. Is the configuration of the environment checked before install -- to ensure that the software can actually be installed?
80. Does the deployment automatically resolve dependencies? Does the tool used download dependencies is needed? Does the tool have sufficient information in order to download these dependencies? Is the dependency information defined assuming an automated tool? Can a human read and work with this information?
81. Is there an automated update process available?
82. Is roll-back of an update possible? If so, are instructions provided? Is there tool support?
83. Is roll-back of an install possible? If so, are instructions provided? Is there tool support?
84. Does installation require down-time?

85.Does an update require down-time?

86.Has any information been provided about de-release?