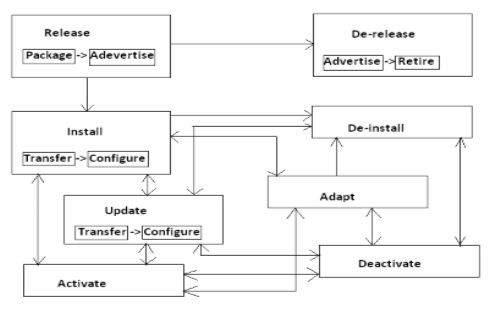
Name and describe the different deployment activities.

* Architecture-driven software development comprises a process that must be carefully planned, modeled, analyzed and finally executed.
* Planning: It is critical that deployment of a software system be carefully planned. Many important system properties will be affected by the system’s deployment. All parameters that affect the system in any way should be taken into account and planned in an efficient manner.
* Modeling: A successful deployment plan for a large, long-lived and distributed software system can be made and affected if the system architects create a detailed model comprising all concerns pertaining to the system’s deployment.
* Analysis: After the modeling process, the system’s deployment model will have to be analyzed for properties of interest. A careful analysis has to be done if there is more than one acceptable option.
* Implementation: Once the deployment model has been modeled, a specific software system’s deployment needs to be effected. Some of the activities take place on the host that is the source of deployed process and some activities takes place on the host that is destination of the deployed component.

The activities which focus on the role of software architecture in the deployment process are as follows:

1. Release
2. Install
3. Activate
4. Deactivate
5. Update
6. Adapt
7. De-install
8. De-release

The various deployment processes along with its interactions are shown below:



**1. Release**:

A software system’s release is the initial activity in the systems deployment process. It takes place on producer site after the system’s development has been completed. The system is packaged so that it can be transferred to the consumer sites. The packaged system will typically contain the following:

* The system’s description including its software architectural configuration, dependencies on system level facilities and any external components and requirements specific to individual software system elements and the entire system.
* All the necessary software modules
* A deployment model indicating which component need to be deployed on which processor or host

**2.Install:**

Once the system has been packaged at the producer site and transferred to the consumer site, it is ready to be configured and installed for operation.

**3.Activate**:

Once the system is installed, it needs to be activated for use on the target hosts. Activation consist of providing a command, or sequence of commands that will be required to start up the system.

**4.Deactivate:**

It involves disabling or shutting down a system of any of the system’s facilities that are still active on the target hosts.

**5.Update:**

Once the system has been installed and activated on target hosts, it may need to be updated overtime for different reasons. The system may need to be deactivated before it is updated and thereafter re-activated. Updates are initiated by the system’s producers and involve the same activities as the original installation. It is critical that the update must be properly reflected in its architectural models. If it is not ensured, the architecture will degrade and any subsequent updates may result in system defect.

**6.Adapt:**

Adaptation encompasses a wide range of activities and results in dynamically changing the system in response to events in the system’s execution environment.

**7.De-install:**

If the system is no longer needed on the consumer sites, it is removed. The de-installation is done by simply reversing the steps taken during installation. It should be noted that, before the system is de-installed, it may need to be deactivated first.

**8.De-release:**

The producer of the given system may decide not to support the system or retire the system any longer i.e. the producer may decide to retire the system. This may be because of evolution of a superior version of that software or market size is too small or the producer has discontinued the product and gone out of business etc. The withdrawal of the producers support for the system is usually advertised. The system’s consumer can then decide whether they still want to use the system with the accompanying risk or deinstall the system.

 Which facets of a system should be modelled in order to solve the deployment problem?

Top of Form