SWE40006
Software Deployment and Evolution

Lecture 2

Deployment Fundamentals

Related Learning objectives

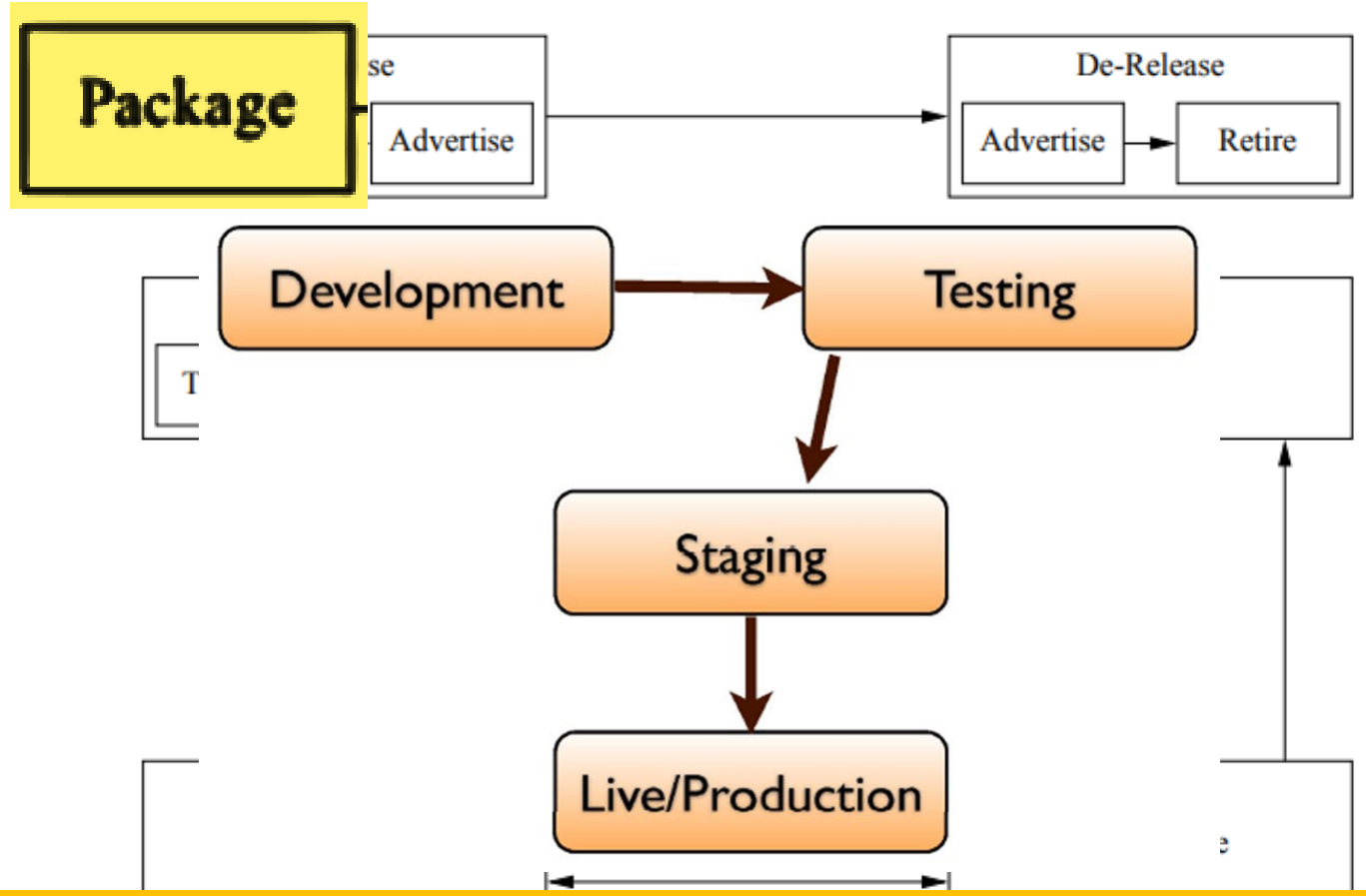
1. Describe the activities in software deployment and apply these in a problem context.
2. Apply knowledge of software environments to plan development and deployment.

Lecture overview

This lecture will cover,

- **Deployment - Release (Package)**
- Deployment - Release (Advertise)
- Installation of software
- Activation

Software Deployment - Packaging



Package Creation Depends on Target Environment

Packaging Software

- Software needs to be *packaged* and *versioned* before transfer to any non-developer environment
- Package model used for staging and production needs to be identical
- Packaging used for ’testing infrastructure’ need not be formal
  - but typically is done via scripts

What is in a Package?

“A package is a collection of files, with instructions on what to do with them"

A software package contains:

- The actual executable software and all associated libraries/components (for target environment)
- Meta-data that identifies the software as well as the environment that it needs
- Data files and Configuration information
- Scripts/Code that help transfer software to the target environment (optional)

Software Package in Windows - MSI

- MS- Windows MSI Package contains:
- Products (identified using a GUID)
- Components (identified using GUIDs)
- DLLs, COM components, Data files
- Key Paths / Key Files
  - Data sources, Registry Keys, Directory Path etc. (Only one allowed and enforced)
- Features (hierarchical group of components or features)

Software Package in Linux - RPM

```
Name                : vnc-server
Relocations         : (not relocateable)
Version             : 4.0
Vendor              : Red Hat, Inc.
Release             : 0.beta4.3.2
Build Date          : Mon 17 Nov 2003 04:54:54 PM BRST
Install Date        : (not installed)
Build Host          : daffy.perf.redhat.com
Group               : User Interface/X
Source RPM          : vnc-4.0-0.beta4.3.2.src.rpm
Size                : 3094107
License             : GPL
Signature           : DSA/SHA1, Mon 17 Nov 2003 05:31:09 PM BRST, Key  ID b44269d04f2a6fd2
Packager            : Red Hat, Inc.
URL                 : http://www.realvnc.com
Summary             : A VNC server.
Description         :
The VNC system allows you to access the same desktop from a  wide variety of
platforms. This package is a VNC server,  allowing others to access the desktop on
your machine.
```

- An Example of Metadata from a Linux package

Linux Packages Example - Python

Required by Python

```
/bin/sh
/usr/bin/env
/usr/bin/python
libbz2.so.1  libc.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1)
libc.so.6(GLIBC_2.1.2)
libc.so.6(GLIBC_2.1.3)
libc.so.6(GLIBC_2.2)
libc.so.6(GLIBC_2.3)
libcrypt.so.1
libcrypt.so.1(GLIBC_2.0)
…
```

Provided by Python

```
rgbimgmodule.so
rotormodule.so
selectmodule.so
shamodule.so
shmmodule.so
strop.so
structmodule.so
termios.so
timemodule.so
timingmodule.so
unicodedata.so
python-abi = 2.3
python-optik = 1.4.1
…
```

This information is embedded into the package

Linux Packages Example - Python

Required by Python

```
/bin/sh
/usr/bin/env
/usr/bin/python
libbz2.so.1  libc.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1)
libc.so.6(GLIBC_2.1.2)
libc.so.6(GLIBC_2.1.3)
libc.so.6(GLIBC_2.2)
libc.so.6(GLIBC_2.3)
libcrypt.so.1
libcrypt.so.1(GLIBC_2.0)
…
```

Valid versions

Provided by Python

```
rgbimgmodule.so
rotormodule.so
selectmodule.so
shamodule.so
shmmodule.so
strop.so
structmodule.so
termios.so
timemodule.so
timingmodule.so
unicodedata.so
python-abi = 2.3
python-optik = 1.4.1
…
```

Version provided

This information is embedded into the package

Why Package?

- Allows for a standard method for software distribution and installation
- Ensure that all dependencies are valid (i.e. components and libraries needed)
  - If properly done, will avoid the “Missing DLL" or “Invalid/Incorrect DLL" problem
- Allows installation of a proper and valid set of data
- Package *installers* will check for compatibility in required components and guarantee that install is successful
- The more information in the package, the better

Simpler Packaging Options

- The simplest packaging option is a ZIP file
  - Eclipse still uses this for developers, but the user has to do all of the hard work
- java has JAR files -- a very simple package structure
  - JAR files are ZIP files (extension is renamed)
  - JAR files contain compiled class files, configuration data
- Dependencies and other requirements need to be managed by the developers (or) by using another packaging model on top

Firefox Extensions

- A Firefox extension is distributed as an XPI file
- Extensions are created using javascript, XHTML, DTD, CSS
  - compressed using ZIP format
- Example: Adblock Extension:

XPI File: adblock_plus-1.1.1-fx+sm+tb.xpi
Directories: chrome, components, and defaults
Data Files: chrome.manifest, install.rdf

Contents of install.rdf:

```
...
<!-- FireFox -->
<em:targetApplication>
    <Description>
    <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
    <em:minVersion>3.0.9</em:minVersion>
    <em:maxVersion>3.6b1pre</em:maxVersion>
    </Description>
</em:targetApplication>
...
```

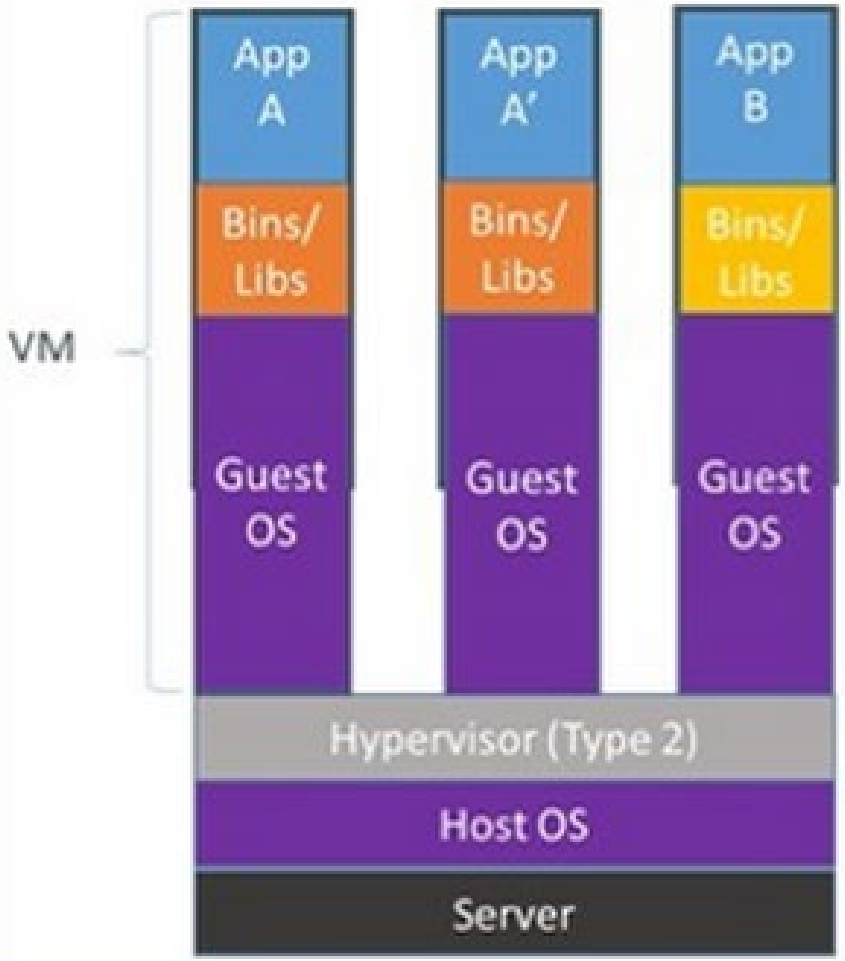
Packaging Models

- Microsoft recommends using MSI
  - Windows Installer can process this package and install the contents
- Linux has RPM as well as DEB format
  - UPS - Red Hat Package Manager
  - DEB - Debian Package Format (used by Ubuntu)
- Apple OSX uses DMG format (Disk Image) as the container
  - Complex installs are done using a Package Manager
- Custom packages are possible as well

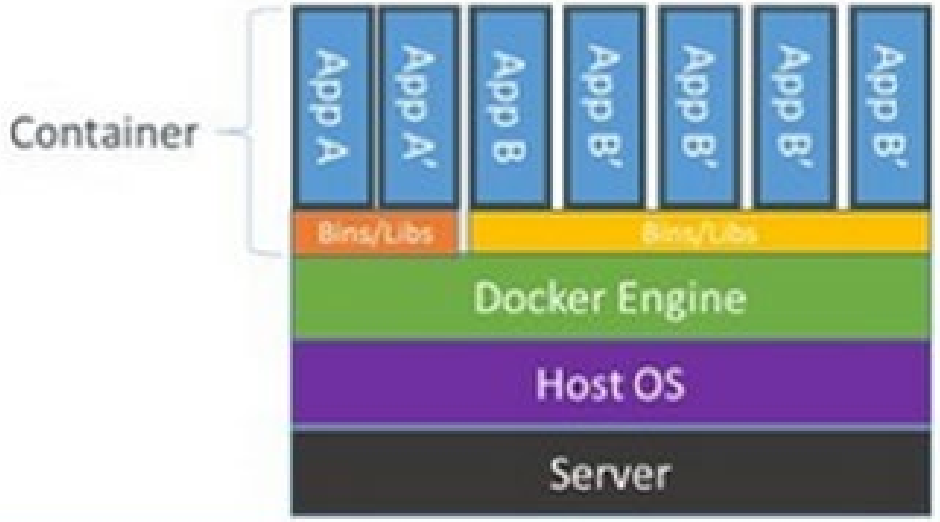
Lightweight Containers

- A new generation of light weight containers are becoming mature
- Example: Docker.io
  - Light weight portable, self-sufficient container
  - The container will contain what would normally go into a “virtual machine"
  - You just deploy the entire container
  - This saves a lot of time & configuration mess
- See: https://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/

Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries

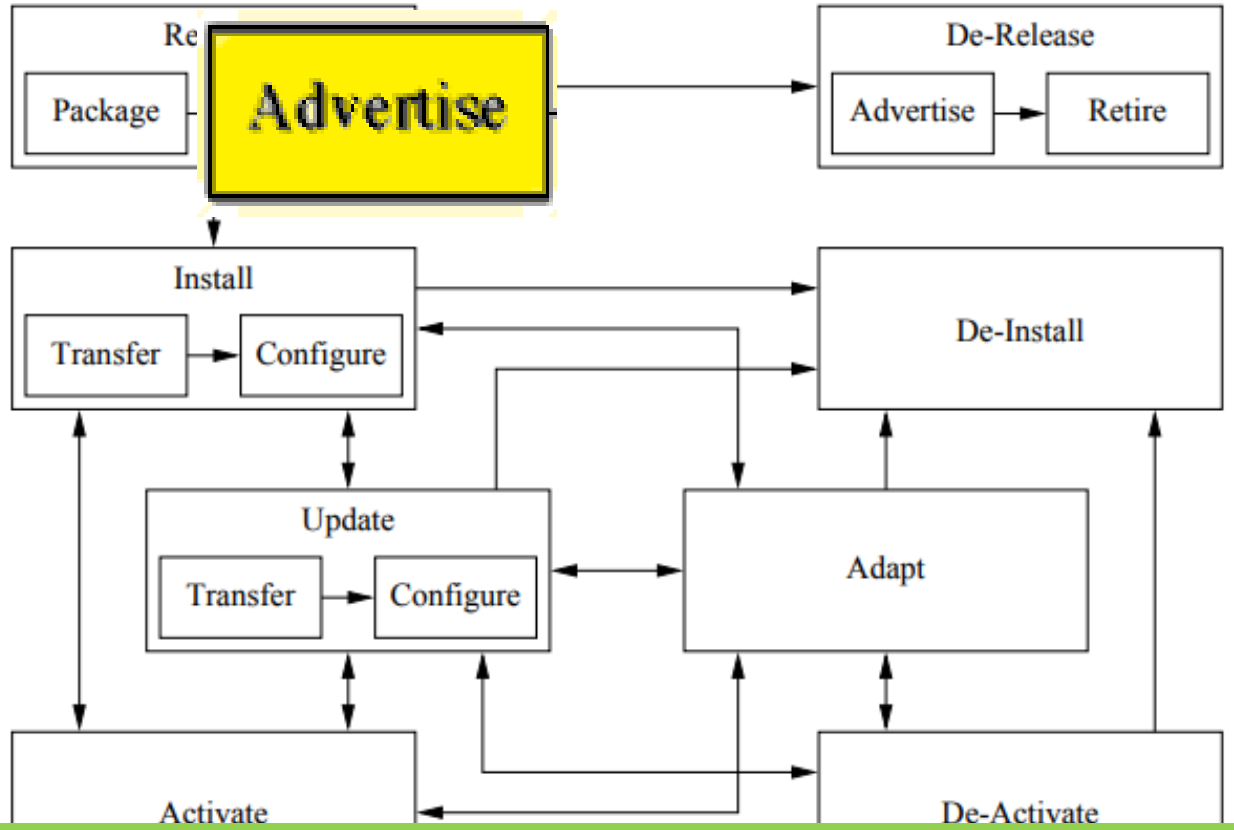


Lecture overview

This lecture will cover,

- Deployment - Release (Package)
- **Deployment - Release (Advertise)**
- Installation of software
- Activation

Software Deployment - Packaging



Software is packaged, next steps are ...

Advertise

- This step is beyond the traditional ‘marketing’ pitch
- Areas that customers want to know are:
  - Licensing (Is this what I want?)
  - Support (If I get stuck, can I get help?)
  - Roadmap (Is this a dead-end product?)

Software Licensing

- What are the consideration?
  - Commercial
  - Open Source
- Potential Issues:
  - License used by the libraries and components
  - Some licenses are viral (for instance, GPL -- if you use a GPL component, then you must release your product under GPL as well)
- Commercial for server use *vs* client use
- Some licenses are for users, not installations

Open Source License Types

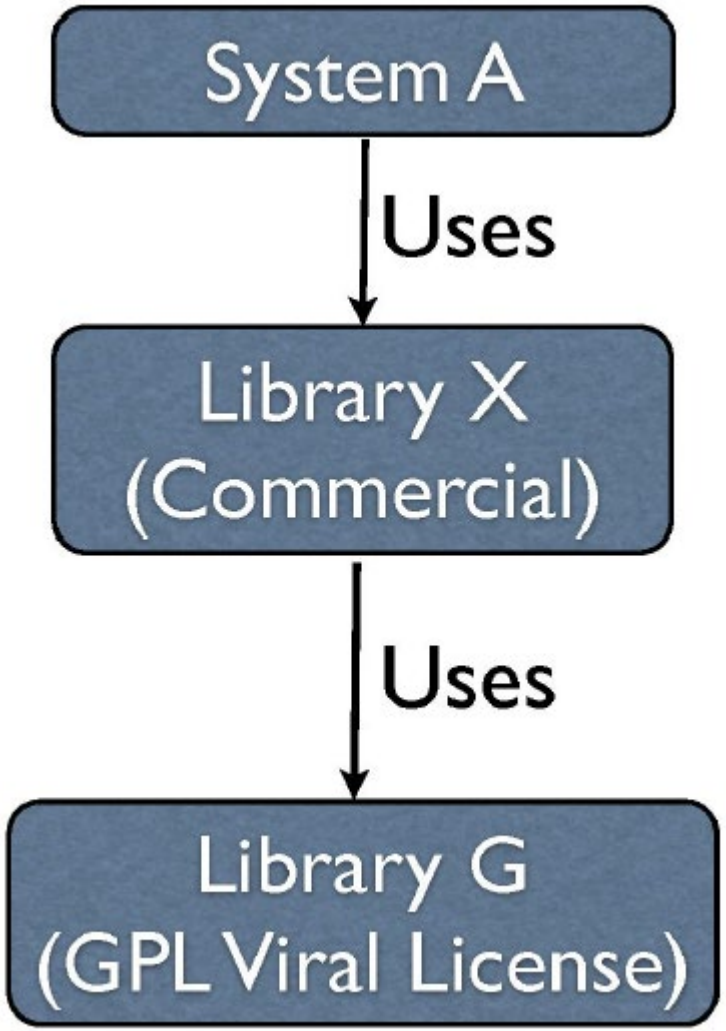
- Broadly three types:
  - Highly Restrictive /Viral / Reciprocal (e.g. GPL)
  - Restrictive (e.g. LGPL, Mozilla)
  - Un-restrictive (e.g. BSD, Apache)
- Restrictive licenses come with strings attached on how you can use them
- These licenses can impact on any code you distribute

Restrictive Licenses in Open Source

- Viral licenses (e.g. GPL) force developers to distribute your source code if they use a library that has a GPL license
  - So, if you use a GPL library -- and want to distribute your code -- then your code is infected by GPL
  - High risk for commercial software
- Restrictive licenses allow you to use and distribute third-party libraries as long as you make no modifications to these libraries

Short Problem

- Is System A infected by GPL?



Licensing - Other Considerations

- Volume Licensing *vs* Individual Use Licensing
- Licensing for usage *vs* installed copies
  - 100-user database server license
  - Quite complex, if you have application servers in-front of a database server
- Restricted run-time licensing
- Licenses that lease software
- License of libraries used must permit re-distribution
- Licenses that restrict per CPU, per Core

Licensing and Deployment

- Critical that all libraries used have licenses that permit deployment in the target environment
  - E.g. If you use a GPL library for Image processing in your application -- your code is now infected with GPL license as well.
- Libraries and Components have *re-distribution licensing* arrangements
  - If you use a commercial encryption library in your application, you will need a re-distribution license in order to bundle that in your package.

Short Problem - 1

- You are building a photo library application for Windows 11
  - plan to sell it for $10. You make use of a **GPL image processing library** for *'face detection'* but don't disclose it
- Questions:
  - Should you provide the source code along with your application binary?
  - Does Windows 11 app. store permit this?
  - Will you get into (*serious*) legal problems?
  - Is this morally correct? / ethically correct?

Morals *vs* Ethics

- Simply put (for our purposes),
  - Morals: Set of personal beliefs (right / wrong)
  - Ethics: Codified framework (by others)
- Examples of relevant ethical frameworks:
  - ACS Code of Ethics
  - IEEE Code of Ethics
- Most professional groups have their own ethical frameworks (e.g. lawyers, doctors, politicians?)

IEEE Code of Ethics

We, the members … agree:

- to hold paramount the **safety, health, and welfare** of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment;
- to avoid real or perceived **conflicts of interest** whenever possible, and to disclose them to affected parties when they do exist;
- to be **honest and realistic i**n stating claims or estimates based on available data;
- to **reject bribery** in all its forms;
- to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;
- to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full **disclosure of pertinent limitations**;
- to seek, accept, and offer honest criticism of technical work, to acknowledge and **correct errors**, and to **credit properly the contributions of others**;
- to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;
- to avoid injuring others, their property, reputation, or employment by **false or malicious action**;
- to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

ACS Code of Ethics… TLDR?

- 4.1 To uphold and advance the honour, dignity and effectiveness of the profession of information technology and in keeping with high standards of competence and ethical conduct, a member must:
- a.be honest, forthright and impartial, and
- b.loyally serve the community, and
- c.strive to increase the competence and prestige of the profession, and
- d.use special knowledge and skill for the advancement of human welfare.
- 4.2 The personal commitments set out in NR4.3 and NR4.4 bind each member with regard to that member's professional conduct.
- 4.3 Values and Ideals: I must act with professional responsibility and integrity in my dealings with the community and clients, employers, employees and students. I acknowledge:
- 4.3.1 Priorities: I must place the interests of the community above those of personal or sectional interests.
- 4.3.2 Competence: I must work competently and diligently for my clients and employers.
- 4.3.3 Honesty: I must be honest in my representations of skills, knowledge, services and products.
- 4.3.4 Social Implications: I must strive to enhance the quality of life of those affected by my work.
- 4.3.5 Professional Development: I must enhance my own professional development, and that of my colleagues, employees and students.
- 4.3.6 Information Technology Profession: I must enhance the integrity of the information technology profession and the respect of its members for each other.
- 4.4 Standards of Conduct
- The standards of conduct set out in these National Regulations explain how the Code of Ethics applies to a member's professional work. The list of standards is not necessarily exhaustive and should not be read as definitively demarking the acceptable from the unacceptable in professional conduct in all practical situations faced by a member. The intention of the standards of conduct is to illustrate, and to explain in more detail, the meaning of the Code of Ethics in terms of specific behaviour. The fact that a member engages in, or does not engage in, these standards does not of itself guarantee that a member is acting ethically, or unethically, as applicable. A member is expected to take into account the spirit of the Code of Ethics in order to resolve ambiguous or contentious issues concerning ethical conduct.
- 4.5 Priorities
- In accordance with NR4.3.1:
- 4.5.1 I must endeavour to preserve continuity of information technology services and information flow in my care.
- 4.5.2 I must endeavour to preserve the integrity and security of the information of others.
- 4.5.3 I must respect the proprietary nature of the information of others.
- 4.5.4 I must endeavour to preserve the confidentiality of the information of others.
- 4.5.5 I must advise my client or employer of any potential conflicts of interest between my assignment and legal or other accepted community requirements.
- 4.5.6 I must advise my clients and employers as soon as possible of any conflicts of interest or conscientious objections which face me in connection with my work.
- 4.6 Competence
- In accordance with NR4.3.2:
- 4.6.1 I must endeavour to provide products and services which match the operational and financial needs of my clients and employers.
- 4.6.2 I must give value for money in the services and products I supply.
- 4.6.3 I must make myself aware of relevant standards, and act accordingly.
- 4.6.4 I must respect and protect my clients' and employers' proprietary interests.
- 4.6.5 I must accept responsibility for my work.
- 4.6.6 I must advise my clients and employers when I believe a proposed project is not in their best interest.
- 4.6.7 I must go beyond my brief, if necessary, in order to act professionally.
- 4.7 Honesty In accordance with NR4.3.3:
- 4.7.1 I must not knowingly mislead a client or potential client as to the suitability of a product or service.
- 4.7.2 I must not misrepresent my skills or knowledge.
- 4.7.3 I must give opinions which are as far as possible unbiased and objective.
- 4.7.4 I must give realistic estimates for projects under my control.
- 4.7.5 I must qualify professional opinions which I know are based on limited knowledge or experience.
- 4.7.6 I must give credit for work done by others where credit is due.
- 4.8 Social Implications
- In accordance with NR4.3.4:
- 4.8.1 I must protect and promote the health and safety of those affected by my work.
- 4.8.2 I must consider and respect people's privacy which might be affected by my work.
- 4.8.3 I must respect my employees and refrain from treating them unfairly.
- 4.8.4 I must endeavour to understand, and give due regard to, the perceptions of those affected by my work.
- 4.8.5 I must attempt to increase the feelings of personal satisfaction, competence, and control of those affected by my work.
- 4.8.6 I must not require, or attempt to influence, any person to take any action which would involve a breach of the Code of Ethics.
- 4.9 Professional Development
- In accordance with NR4.3.5:
- 4.9.1 I must continue to upgrade my knowledge and skills.
- 4.9.2 I must increase my awareness of issues affecting the information technology profession and its relationship with the community.
- 4.9.3 I must encourage my colleagues, employees and students to continue their own professional development.
- 4.10 Information Technology Profession
- In accordance with NR4.3.6:
- 4.10.1 I must respect, and seek when necessary, the professional opinions of colleagues in their areas of competence.
- 4.10.2 I must not knowingly engage in, or be associated with, dishonest or fraudulent practices.
- 4.10.3 I must not attempt to enhance my own reputation at the expense of another's reputation.
- 4.10.4 I must co-operate in advancing information processing by communication with other professionals, students and the public, and by contributing to the efforts of professional and scientific societies and schools.
- 4.10.5 I must distance myself professionally from someone whose membership of the Society has been terminated because of unethical behaviour or unsatisfactory conduct.
- 4.10.6 I must take appropriate action if I discover a member, or a person who could potentially be a member, of the Society engaging in unethical behaviour.
- 4.10.7 I must seek advice from the Society when faced with an ethical dilemma I am unable to resolve by myself.
- 4.10.8 I must do what I can to ensure that the corporate actions of the Society are in accordance with this Code of Ethics.
- 4.10.9 I acknowledge my debt to the computing profession and in return must protect and promote professionalism in information technology.

Short Problem – 1 (again)

- You are building a photo library application that will be hosted in the cloud (i.e. it is a web app.)
  - You make use of a GPL ’face detection’ library -- but do not release the source code
  - Images of people will be uploaded to the cloud without the user knowing…
- Questions:
  - Should you make your source code available!
  - Are there any legal issues to navigate?
  - Moral / Ethical concerns!

Lecture Overview - Where are we?

- This lecture will cover,
  - Deployment - Release (Pacl‹age)
  - **Deployment - Release (Advertise)**
    - Licensing
    - **Support**
    - Roadmap
  - Installation of software
  - Activation

Support

- What types of support is provided?
  - Frequently asked questions?
  - Communicating modes: e-mail, telephone, IM
  - Can customers report issues?
- Does support have good tools?
  - Distributing information (Web site)
  - Communication (discussion boards)
  - Issue reporting and tracking (Bugzilla)
  - Crash reporting and resolution

Product Support

- General product support provides users with:
  - Frequently asked questions
  - Quick start guide and User manual
  - Knowledge Base (wikis or web pages)
  - Discussion boards
  - Online defect reporting (e.g. via Bug Trackers)
  - Sending message to support staff
  - Telephone support staff
  - IM/Chat support

Product Support

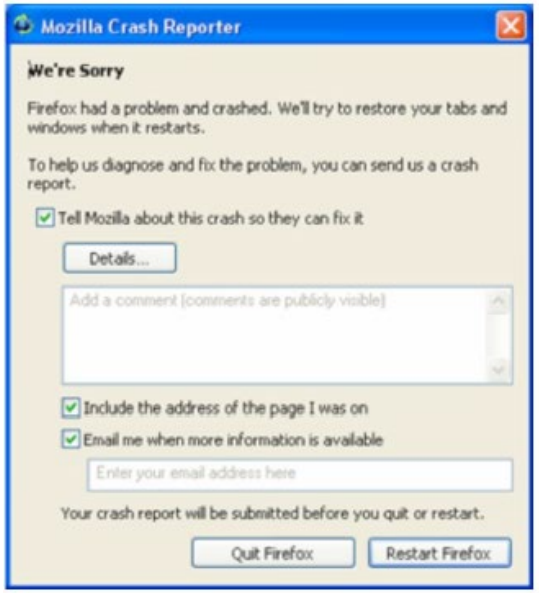
- Support staff almost always need a way into various aspects of the software currently deployed
  - Remote access
  - Ability to change passwords
  - Ability to replicate issue on ’staging environment’
- Some products are easier to support than others

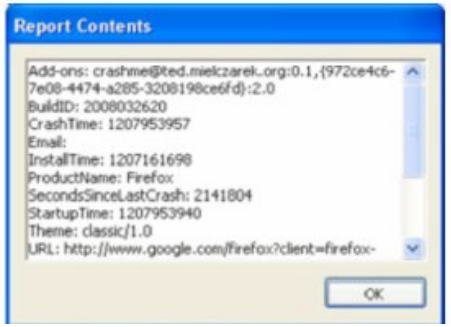
Support - Reporting a Defect

- Customer View:
  - I have an issue; can this be reported?
  - Will I get an e-mail once it is resolved?
  - How will the issue be processed?
- Tools typically used are:
  - Bugzilla or JIRA (for issue tracking and reporting resolutions)
  - Change logs (that report on all issues that have been resolved)

Support - Crash Reporting and Tracking

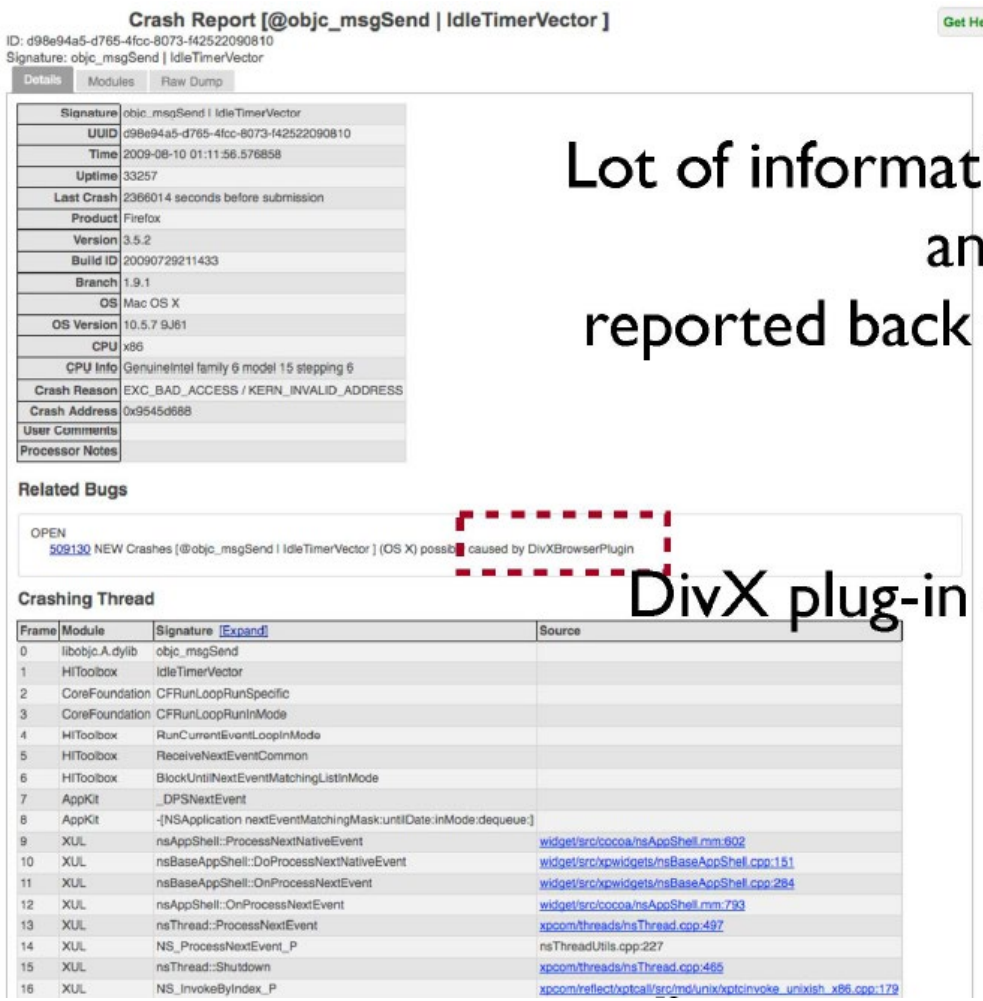
- Contemporary software has started to build in crash-reporting modules





- https://youtu.be/yX8yrOAjfKM
- https://opensource.org/licenses

Crash Reporting - Captured by Firefox



Crash Report [@objc_msgSend | IdleTimerVector]

Get Help

ID: d98e94a5-d765-4fcc-8073-f42522090810
Signature: objc_msgSend | IdleTimerVector

Details | Modules | Raw Dump

Signature	objc_msgSend	IdleTimerVector
UUID	d98e94a5-d765-4fcc-8073-f42522090810
Time	2009-08-10 01:11:56.576858
Uptime	33257
Last Crash	2366014 seconds before submission
Product	Firefox
Version	3.5.2
Build ID	20090729211433
Branch	1.9.1
OS	Mac OS X
OS Version	10.5.7 9J61
CPU	x86
CPU Info	GenuineIntel family 6 model 15 stepping 6
Crash Reason	EXC_BAD_ACCESS / KERN_INVALID_ADDRESS
Crash Address	0x9545d688
User Comments	
Processor Notes	

Related Bugs

OPEN
509130 NEW Crashes [@objc_msgSend | IdleTimerVector] (OS X) possibly caused by DivXBrowserPlugin

Crashing Thread

Frame	Module	Signature [Expand]	Source
0	libobjc.A.dylib	objc_msgSend	
1	HIToolbox	IdleTimerVector	
2	CoreFoundation	CFRunLoopRunSpecific	
3	CoreFoundation	CFRunLoopRunInMode	
4	HIToolbox	RunCurrentEventLoopInMode	
5	HIToolbox	ReceiveNextEventCommon	
6	HIToolbox	BlockUntilNextEventMatchingListInMode	
7	AppKit	_DPSNextEvent	
8	AppKit	-[NSApplication nextEventMatchingMask:untilDate:inMode:dequeue:]	
9	XUL	nsAppShell::ProcessNextNativeEvent	widget/src/cocoa/nsAppShell.mm:602
10	XUL	nsBaseAppShell::DoProcessNextNativeEvent	widget/src/xpwidgets/nsBaseAppShell.cpp:151
11	XUL	nsBaseAppShell::OnProcessNextEvent	widget/src/xpwidgets/nsBaseAppShell.cpp:284
12	XUL	nsAppShell::OnProcessNextEvent	widget/src/cocoa/nsAppShell.mm:793
13	XUL	nsThread::ProcessNextEvent	xpcom/threads/nsThread.cpp:497
14	XUL	NS_ProcessNextEvent_P	nsThreadUtils.cpp:227
15	XUL	nsThread::Shutdown	xpcom/threads/nsThread.cpp:465
16	XUL	NS_InvokeByIndex_P	xpcom/reflect/xptcall/src/md/unix/xptcinvoke_unixish_x86.cpp:179

Lot of information is captured
and
reported back to developers

DivX plug-in crashed

Short Problem - 2

- Bart feels that asking permission to send reports each time the application crashes is irritating users. So, he decides to make the default option 'send crash reports' & adds a few sentences into the license agreement (knowing that most users will not read it).

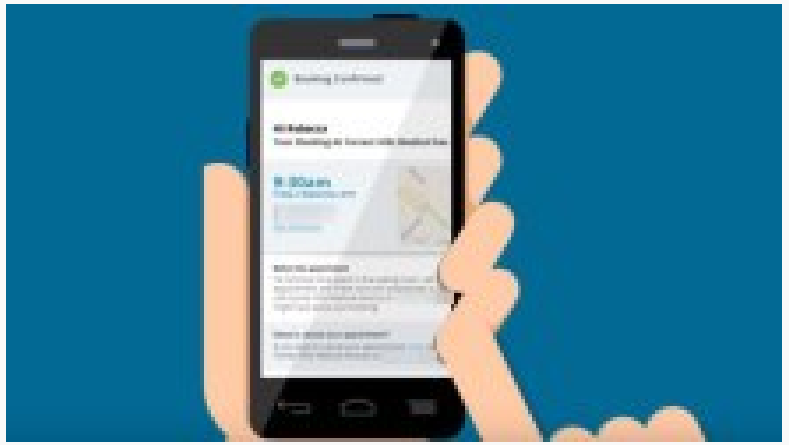
- Is this morally acceptable (to you)?
- Does this violate ethical standards of your profession?

Short Problem - 3

- Should a software system be given blanket permission while installing to:
- Change terms of the agreement in the future?
- Update the software forever and ever!
- Is the following sentence in a license acceptable!
- “CSG-Soft is constantly innovating in order to provide the best possible experience for its users. You acknowledge and agree that the form and nature of the Services which CSG-Soft provides may change from time to time without prior notice to you.”

- Should a Health Stats app share medical information with personal injury lawyers?
- There was no opt-out option

Health app used by millions is sharing users' information with lawyers



An Australian medical appointment booking app, which boasts more than 15 million users a year, is passing on patients' information to personal injury lawyers and other third parties.

http://www.abc.net.au/news/2018-06-25/healthengine-sharing-patients-information-with-lawyers/9894114

Product Roadmap

- Customers will want to know a general plan for the product
  - I have purchased Version 1.0 -- what is the plan for the next release?
- Roadmaps are useful for improving customer confidence
- Also helps for customer retention
  - Customers in general try not to move to a new product
- Must communicate: Goals and Schedule

Roadmap Example: Firefox Goals (2014)

- Technical Roadmap: https://github.com/mozilla/bedrock/wiki/Technical-Roadmap
- Focus changes from time/year to time/year, eg, for 2022:

  focused on technical debt reduction, modernization, and stability improvements, with specific goals like removing legacy code, improving CSS bundling, and evaluating new front-end frameworks
- Goals evolve from year to year … (see above)

Roadmap Example: Firefox Schedule

- This information is useful for marketing

Roadmap Example: Firefox Schedule

- This information is useful for marketing

2021

Priority	Q1 2021	Q2 2021	Q3 2021	Q4 2021
P1	Content publishing: Contentful roadmapping.	Content publishing: Contentful implementation.	Content publishing: Contentful implementation.	Content publishing: Contentful home page launch.
	Product support: Add Mozilla VPN as a new product surface.			
P2	Analytics: Add GA client download ID to stub attribution.	L10n: Enable Smartling vendor pipeline.	Modernization: Front-end build system replacement.	Technical debt: Server-side geolocation improvements.
P3	L10n: 100% completion for Fluent migration.	Technical debt: Remove jQuery as a dependency.		Modernization: Automatic code formatting (Black, Prettier).
				Technical debt: Remove legacy .lang code.

Lecture overview

- This lecture will cover,
  - Deployment - Release (Pacl‹age)
  - Deployment - Release (Advertise)
  - Software Design Implications
  - **Installation of software**
  - Activation

Deployment Choice and Design

- Release choices WILL require changes to the software design/architecture
- Releasing has two steps - Package and Advertise
- Considerations:
- Target environment & Virtualization
  - Packaging model implications
  - Licensing implications
  - Crash Reporting
  - Roadmap

Software Deployment Model

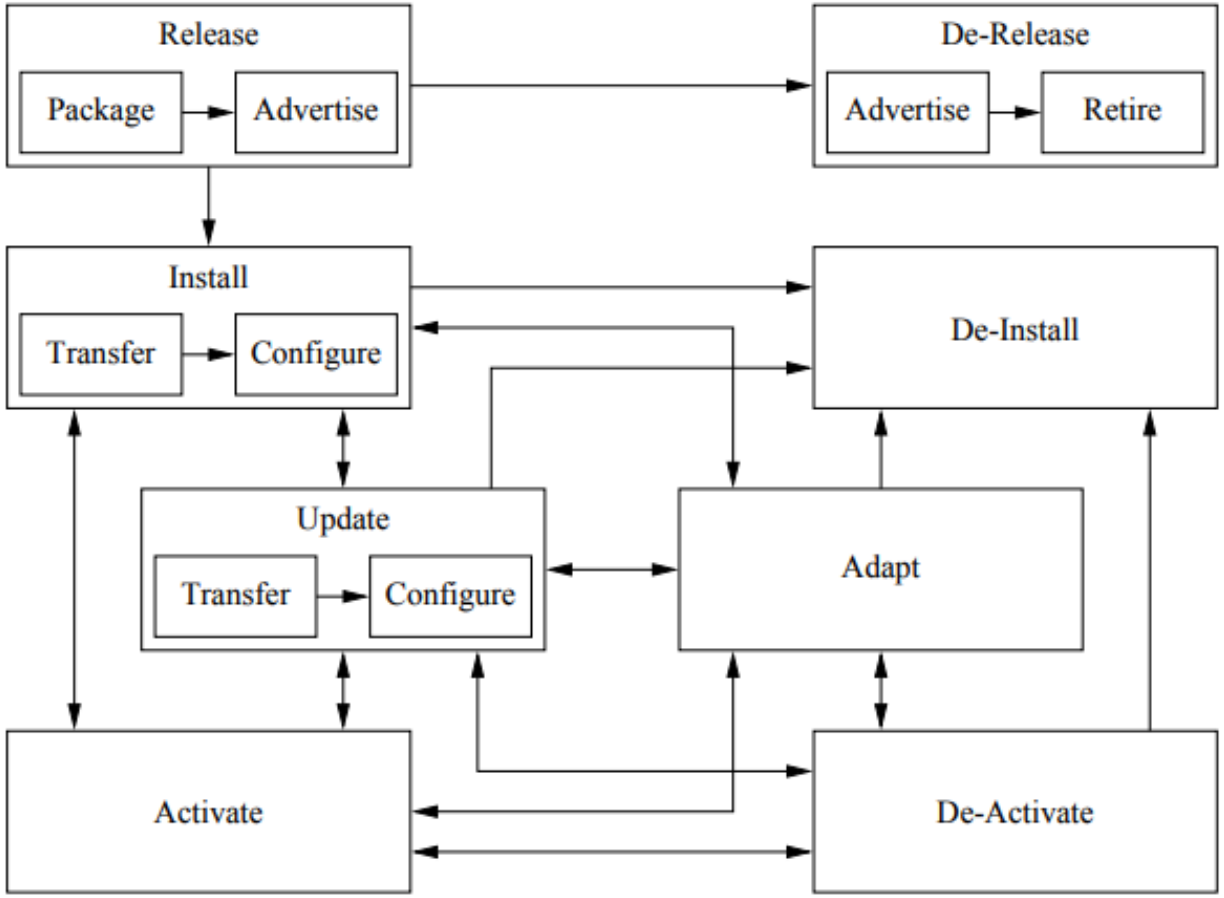


Figure 1: Activities of the Software Deployment Process.

https://www.ics.uci.edu/~andre/papers/T3.pdf

Software Distribution Review

- Software that needs packaging and distribution
- (zip, MSI, DMG, PKG, DEB, XPI…)
- Transfer is completed by the installation software e.g. Windows Installer

https://peacekeeping.un.org/en/deployment-and-reimbursement

Software Deployment - Release

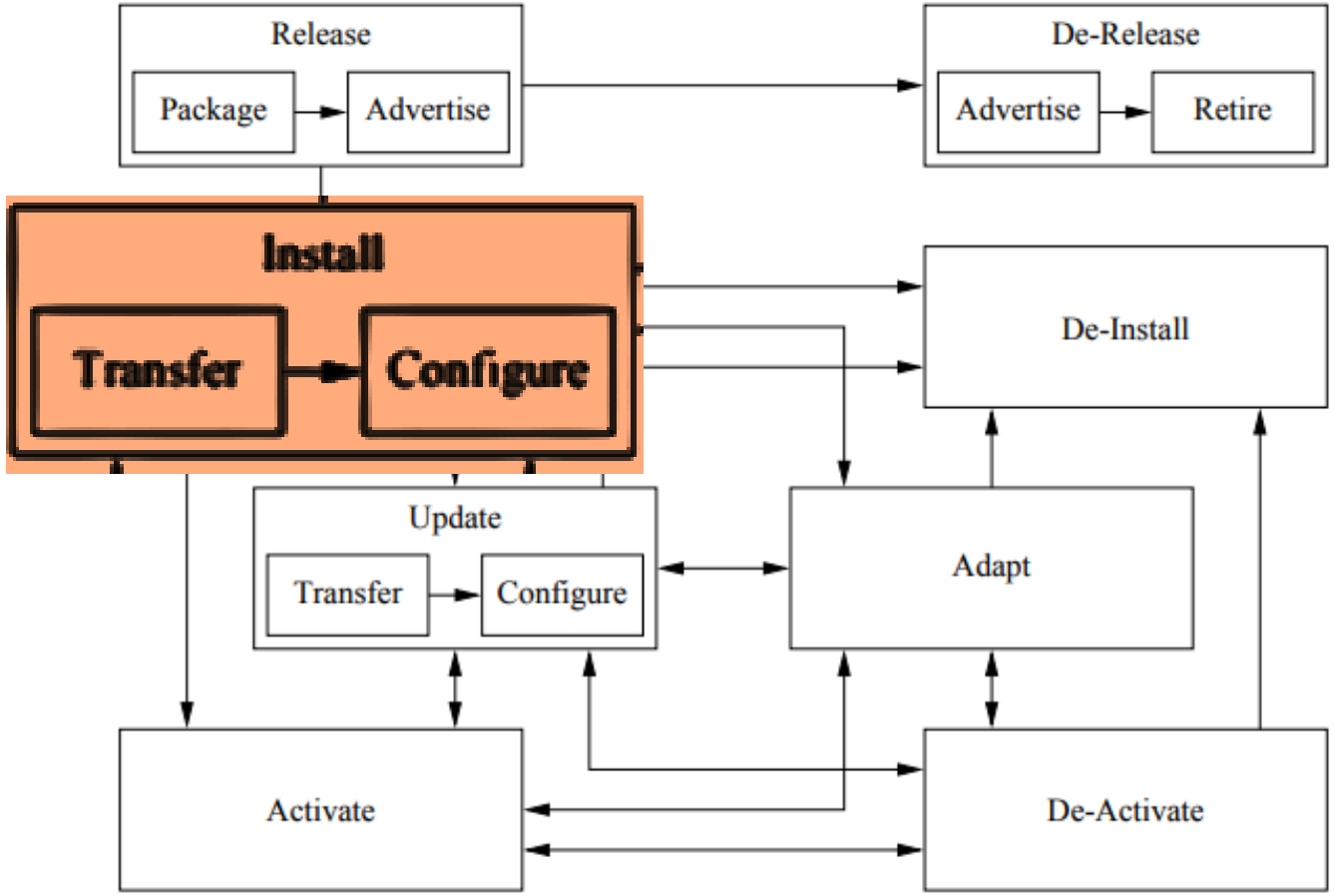


Figure 1: Activities of the Software Deployment Process.

https://www.ics.uci.edu/~andre/papers/T3.pdf

Install - Transfer 1

- Transferring software involves more than copying files:
- Where are we transferring the files to? Can user change this?
- Previous version considerations:
  - Can this transfer execute if previous version is currently in use?
  - Do we over-ride existing files with new?
  - Can multiple versions co-exist?
- Dependency checking among components
  - Libraries provided vs. Libraries needed

Install - Transfer 2

- Co-ordination issues:
  - This is relevant for software that requires an installation across multiple machines or many different types of software
- Example:
  - Installation requires a new version of application, an upgrade to a database system as well as installing a new messaging server on 3 different machines
- Content Delivery issues:
  - Can “Google Earth" transfer all content?

Install - Transfer 3

- Security implications / Privileges
  - Authorization and Authentication (does software require an admin. to install?)
- Installer needs to download application components from the internet
  - Access to the network
- Integrity checking by finger-printing package
  - MD5/SHA checksum (used by Ubuntu)
  - Critical if multiple parties are allowed to distribute your software
- Transactional integrity must be maintained

Install – Configuration 1

- Once files have been transferred and verified to be valid
  - we need to configure software for use
- Configuration must take into consideration the following issues:
  - Registering components with operating system (specifically in Windows)
  - Setup components, data as well as any databases for use in the current environment (Dev., Test, Staging, Production)
  - Setup of connectivity in a multi-system environment (i.e. with mail servers, proxies etc.)

Install - Configuration 2

- Security aspects:
  - Verify that sufficient privileges exist to ensure product can be started up, particularly important in multi-system configuration
- Configuration information must be contained within the package
  - The installation software can then use this information and perform some standard actions (such as registering a COM component)

Short Problem - 4

- You are the creator of a new game – “Shadows from Id". The game engine itself is relatively small at 50MB. However, the game levels are 30MB each and you have 25 levels. You want to distribute the game online to keep costs down.
  - Will you ask the user to download all levels and game engine in a single hit? (or)
  - Will you download the first level and then have the rest downloaded in the background? (or)
  - Download game levels sequentially (on demand)?

Short Problem - 5

- You plan to allow practically anyone to distribute your new game (Shadows from Id) since the first 2 levels are given away for free.

- How can you allow your users to check that hackers have not inserted a malicious piece of code within the executable of your game engine?

Activation - 1

- When a software system starts up it should (ideally) check
  - System run-time integrity (do all required hardware components, software components, databases, data files, drivers and connections exist?)
  - Authorization level (do we have sufficient privileges to run this software? read/write files? read/write to other devices?)
  - License verification (does the user have a valid license to run this software?)

Activation - 2

- Activation may be multi-step, esp. if many different processes have to be started
  - E.g. Web server, Mail server, DBMS server etc..
- PC only software can still require starting up multiple threads all of which need to report back a valid state before the software can be considered ready for use
  - Activation of software may 'lock' resources (typically certain devices) -- can these locks be acquired?
  - E.g. Some software locks use of video card

Short Problem - 6

- You created a new music player software that will change the landscape. Does your software have specific features that are triggered only on first activation?
  - Are these features available at a later date?
- What does a web browser do upon first activation?
- Do you have any specific first-time activation functions in your final year team project software?

Short Problem 7

- You created a web application that uses Postgres as its database engine.
  - When will you create the database schema? -- During the configure step of installation (or) as part of first time activation?

- A power failure after a few weeks of operation has caused the database server hardware to fail. You activate your web application -- what should (ideally) happen?

Key Points

- Software deployment has two steps:
  - Package and Advertise
- What gets packaged changes depending on target environment
  - Dev./ Test I Staging / Live
- Different packaging models and approach exist
- Advertising: Licensing, Support and Roadmap
- Packaging and Advertising choices impact on product design
  - hence these need to be made early
- Some licenses are viral
- Activation can be first-time run (extra config.) or normal.

All done

- Other things to do this week:

- Lecture Q+A
- Tutorial: discuss and understand those pesky tutorial questions.
- Finish that first task and start the second one.