Analysis of the Concerns Associated with the Rapid Release Cycle

Abhishek Kushwaha
Indian Institute of Information Technology
Allahabad

Sushil Kumar Verma
Indian Institute of Information Technology
Allahabad

Chandan Sharma
Indian Institute of Information Technology
Allahabad

ABSTRACT
Organizations have always considered the product release cycle an important factor in the development of the product. Most of the organizations have well defined release schedule in the documentation and are constantly updated whenever changed. There is a recent trend in the organizations to switch to rapid release cycle from earlier slower cycle. In this paper the probable concerns that can arise from these rapid schedules is discussed along with certain reasons that might be pushing the organizations to move to rapid cycle. Probable solutions and suggestions have been outlined that can help to reduce the severity of concerns, if applied largely and properly.

Keywords
Software updates, rapid release cycle,e-waste, updatevs. upgrade.

1. INTRODUCTION
Software industry and some portion of electronics market has quickly moved to the rapid release cycle. The objective of staying leader in the market has always pushed the organizations to release new products quickly or at least pretend to release new product with not very useful minor changes which could otherwise be provided by over the internet updates in case of softwares. Caught in the frenzy, users are constantly expecting more from the company. This coupling of organizational objective and users' expectations has been a major factor in driving the market but the recent trend seems to remove the bar on the speed limit. Weekly new products though paint a good picture of the market, but altogether give rise to certain concerns, which will further aggravate with time. Some major steps would be required to address the concerns in a broader view.

2. UPDATES AND UPGRADES
A software update provides bug fixes and minor software fixes to add new features, improve security, patch over a potential flaw in a program's code, fixing security vulnerabilities and improving the usability or performance.

Software updates sometimes include new drivers to support the latest hardware such as printers, CD drives and DVD drives. A software update is sometimes called a software patch too. The updates are bundled with the software, till it is supported i.e.its lifetime. Since the users are not charged for these updates, the most cost effective way to distribute them remains the internet. The users can download the updates at their leisure. Release of Updates makes the market competitive and the developers are busy to find new solutions for the applications. Software update does not require full software packagere-installation.

An upgrade is an entirely new version of aprogramorit can bes aid, are placement of a software product with a newer version of the same product. Upgrades are considerably more time consuming and can mean that user will need to reinstall everything that goes on top of them. It usually contains more functionalities and support more platforms than previous version. As an upgrade might require a complete re-install of the product, it might bring with itcertain new features and UIs which may worth the time. Also, upgrades are generally not free, since they contain features or UIs for which the users have not paid.

Both the updates and upgrades are part of the business. New versions are required due to technological advancements, user demands, rolling out new functionalities, etc. Users buy the new versions to explore the new features, UIs and security offered by the product. It is these versions that generate revenue for the company. The vulnerabilities missed out during development phase are needed to be fixed, otherwise they can be exploited causing a probable loss to the user. To overcome such threats updates come in to picture, users are provided with free fixes and patches to make for companies inability to detect those beforehand. Updates are free, therefore they do not generate revenue for company, but helps in winning the consumer confidence that the product is well supported by the company and that the company is serious for giving a good return on users' investment.

Summarily, update:

1. Is the optimized way to fix the bugs.
2. Helps in fixing security vulnerabilities.
3. Improves the usability and performance.
4. Improves the user interface in more reliable and cheaper way.

Upgrades:

1. Given to the previous user in cheaper rate, possibly.

2. It overwrites the exiting software to provide fresh and new facilities.
3. Helps the company to gain more market share.

Auto and Manual updates: Software updates and upgrades may be automatic as well as manual. Automatic updates and upgrades are those which are programmed so that the patch files or other files are installed on its desired place without any effort from user e.g. virus database updates of antivirus, security updates from Microsoft for windows OS etc.

On the other hand manual updates and upgrades are also available, upgrades are generally manual, basically they requires some technical efforts from the user hence for manual updates and upgrades it is expected that the user is technically sound. Linux updates and some types of games updates are the examples of updates and upgrades which are manual in nature.

Types: Updates can be divided into few types such as **Security updates** (these generally fix the bugs and provide patches for the loopholes), **Functional updates** (these updates add some new to features to the running version), **UI updates** (these kinds of updates add some changes to the user interface of the running version).

As for the upgrades, since it replaces the old version it upgrades everything, security, functionalities and UIs and everything else. But there might be few upgrades that might require some extra hardware for their full utilization and in certain cases some new features might require additional training for the user.

Drawbacks: Certain drawbacks associated with the updates and upgrades can be summarized as follows:

a) Incur extra cost
b) Software becomes vulnerable with open ports for updates
c) Software may lose its originality
d) Updates and upgrades may degrade its stability
e) Downloading and installing the updates does take up some of computer's resources
f) Consumes large amount of internet bandwidth
g) Upgrading and updating is cumbersome job for the user and may decrease productivity of the user
h) Updates may introduce some new bugs
i) Makes changes in predefined setting and make system destabilized.

3. UPDATES AND UPGRADES CYCLE

Updating the software products or applications are important because the technology is changing very rapidly and this raises certain questions like, in reality is the end user getting the actual benefit? Is the end user compatible with new and rapid changes in the applications? No doubt the new update gives the advance features to the user but did the end userreally needed the updates based on his role of work? New updates may change the look and feel of GUI or may change the tags and sometimes changes the way of using the application. So end users need some time to understand the changes practically by using these changes in their work. If the application providers start releasing the new updates very rapidly, it might be very difficult to the end user to understand the changes. It might be possible, some updates are of no use for some end users and the updates require extra space, so it is using the space on the disk and has no use. Timing and cost factors: The timing of updates is generally not predefined, it can appear any time after a vulnerability has been detected. Though, the timing of upgrades of applications is predefined. Sometimes the companies release the updates in the intervals of few days because the previous update caused a new vulnerability. Microsoft Windows major versions have been released at about a time-interval of 4-5 years, with updates being released almost every day right after the release of the major version.

Providing bug fixes and updates to software incur cost on developers' side, the developers must be paid for their effort. Each version is maintained for a given period of time, referred to as lifetime of product, and the user is licensed for free updates during the lifetime of product. Also end user waste the working hours on downloading and installing the new patches. So, it will be extra cost to the end user in terms of bandwidth and productivity.

Type dependence: Update and upgrade cycle depends on type of application. The updates are released very rapidly if the applications are open source because there is no fix on the number of users involve in developing the new functionalities for the application around the world. While in closed source applications the updates can only be released by the person who has the knowledge of the code. Updates sometime create the problem for the other important software and for proper functioning of these software now these software also requires some patches. For example, turbo C is not able to open in full screen in window7.

The upgrade cycle of the paid, costly applications is more systematic, they releases the new version (upgrade) based on the competition on the market and only after a considerable time has passed. The upgrades cycles are also affected by the ultimate use of the product, for an enterprise grade application, it is not supposed to release new versions very often, enterprises cannot spend the resources on upgrades very often, without any major change in the technology.

4. THE RAPID RELEASE CYCLE

Since last few years there is an increase in the tendency of moving towards the rapid release cycle among the software developers and in some cases among hardware manufacturers too. Pondering over the situation, it does not take much time to figure out certain reasons for both hardware and software vendors. One of the reasons for the above migration can be attributed to the "Internet".

In the 90's the Internet was maturing. People were busy experimenting and learning with the limited bandwidth of the

Internet. The slow Internet caused the vulnerabilities to slow to surface and exploits to slow to spread. Limited knowledge of Internet coupled with limited needs caused the companies to stick to longer upgrade cycles of their software/hardware products. Also it was costly for software companies to disseminate quick upgrades to their customers. Maintaining the current version and developing a new version both involves cost, and maintaining this parallel development involves higher cost. So it forced the companies to stay slow on their market releases.

But after the 2000 boom, things started changing. The experience gained in the 90's was now being put in action largely all around the world. The bandwidth of the Internet had increased considerably. Computers were increasingly becoming more useful. With growing number of users and their needs many companies released different versions of same type of software like audio/video player, which led to competition. To grab the larger share in the market, companies quickly released newer versions with added new functionalities. With the passing of time companies organized their release cycles to create an order within the company and in the market. Resultant, the new versions were released at regular interval of time.

However, this was not the case with smaller updates. With the increased deployment of computers in varied fields, there were more vulnerabilities and exploits and Internet helped them spread quickly. This forced the companies to release small updates and patches as quick as possible. Other reasons such as varying UI demands and increasing complexity of software also contributed to quick and large amount of smaller updates. The current scenario that has been observed is releasing all kinds of updates as soon as they are available from the developers.

Few examples can be sighted here such as, Microsoft Windows, which is planning to or already moved to 2-3 years release cycle of major version of Windows OS from previously 4-5 years cycle. It may be a part of company strategy to counter the smaller release cycles of various Linux flavours and Apple Mac OS X. Long-time runner up of very famous and ever going on browser war, Mozilla Firefox has recently switched to rapid release cycle. It took considerable period of time to move from version 3 to version 4, but it took less than a year to move from version 4 to version 9. According to Mozilla's chairman, Mitchell Baker, "If we want the browser to be the interface for the Internet, we need to make it more like the Internet. That means delivering capabilities when they are ready. That means a rapid release process. If we don't do something like this the browser becomes a limiting factor in what the Internet can do." For some years now Apple has strengthened its stand on its annual update cycle which can be very well considered rapid given the complexities of their products.

As for the hardware vendors, the ever increasing performance needs of the users are pushing the manufacturers to quickly roll out newer versions to gain more market. Especially, this trend can be very well observed in gaming graphic cards and mobile phones.

5. BENEFITS FROM RAPID RELEASES

Few of the benefits from the rapid releases can be discussed here since they have greatly pushed the graph in upward direction. The benefits are:

Higher quality (new bug fixes and new functionalities can be provided quickly which increases the usability of the product), **Better service** (developers and manufacturers are more responsive to user needs, which strengthens the trust between them), **Lower cost** (in certain cases there is no need to maintain the older versions, the newer version altogether replaces the older version with reduced development cost), **Enhanced competitiveness** (companies respond quickly to the changing market needs which gives them a lead on their competitors sticking to longer release cycles), **Enhanced marketing** opportunities (each time a new version of a product is released it is extensively covered by technology websites and blogs which enhances the marketing opportunities free of cost).

6. CONCERNS WITH RAPID RELEASE CYCLE

Difficulties to users: Technology has always affected human lives significantly, perception of people changed with each development or change, rapid release cycle is no exception. Rapid releases tend to confuse the users, the trend may make them believe that they are still using a "work in progress" product, whose better version will be available few months or weeks later. This creates a feeling of insecurity among users as to whether they are secure enough to carry out critical tasks over the internet. With each release people grow more eager while waiting for next release.

Many users with limited bandwidth or pay-per-size plan have to make sure that their auto-updates are turned off, which otherwise may very well eat away a good chunk of bandwidth. In large softwares rapid release cycle may cause more bugs, these are fixed through small, continuous, over the internet, updates. These updates consume high bandwidth often affecting users' productivity. The new versions generally disable the older, additional support softwares or add-ons which users generally use for their daily work. This leaves the users with two choices, either to use the previous version of the software or wait for the next version of the add-on.

Rapid release of hardware, most importantly like mobile phones, also raises concerns for users. Mobile phone has come a long way from just a communication utility to a social necessity. With increase in competition, new models are being released every month, if not weeks. Nowadays, the lifetime of a gadget in the hands of the youngsters considerably affects their social life. The peer pressure and related circumstances tends the users to always stay latest with the gadgets, which results in increased expenditures and increased social insecurity to stay ahead among peers. Also,

with every release the end users' hunger for more, in terms of performance increases and companies very well use this tendency for their benefits.

Difficulties to developers: Rapid release cycle is not a complete bliss for developers, though it tends to make themmore competitive, but it also forces them to enable support for older versions of other utility softwares, whose vulnerabilities can still be exploited. The support is sometimes necessary or the product may lose the market and users' interest.

For large and complex softwares, like Microsoft Windows, rapid release cycle may cause the introduction of more bugs. Though the development technologies are continuously updated, the increased size and complexity may still stay unmatched with such upgrades and result in more bugs. This in turn will result in increased amount of online updates along with increase in internet crimes causing wastage of useful resources. For smaller softwares, like Mozilla Firefox web browser, the rapid release causes many add-ons and plugins to stop functioning. Many add-ons and plugins are developed by fans and enthusiasts out of interest, who are then forced to quickly update their add-ons, which is not always possible. The disabling of add-ons and plugins causes the product to lose its functionality, which forces the user to either stay on older version or change the product altogether. Add-ons and plugins developers cannot always stay on rapid release schedule, so it can very well cause the developers to lose their interest in the product stop developing the add-on for that product.

Difficulties to Enterprises: Establishment of IT infrastructure in an enterprise is the most expensive and critical task to perform. IT infrastructure basically comprises of high end servers, firewalls, routers, user systems, softwares, etc. The cost of such hardware is high and the cost of software is still higher and investment is also needed to configure and maintain such systems. The rapid release cycle of hardware and software may prove a headache for IT infrastructure implementers and maintainers because maintaining all the new software and hardware according to enterprise requirements is a tricky job. Certain modules of upgrades may take more time than expected to be tuned to become compatible with other modules. Rapid upgrades in cases can lead to introduction of new unforeseen bugs, because initially it is very difficult to identify all the bugs and this may cost the company.

Upgradation of IT infrastructure usually requires extra skills and training for employees, but due to rapid release the time duration between consecutive training is reduced, this puts an extra burden on IT managers in terms of time and money to make their employees familiar with new environment. Also, new releases have lesser documentation available, this further lengthens the training effort thus affecting the IT budget of the organization. IT upgradation process sometimes affects the working of other departments too, resulting, this affect can lead to delay in completion of certain tasks/projects in time. Despite the organizations best intentions such delays can critically affect the customer relationship and that can affect the organizations' overall performance in the market.

Cost factors: Cost becomes an important factor in deciding the use of asoftware, should a costly software be used or its open-source, free alternative is used. The basic tendency is towards the reduction of average cost of use per person for a software, which is true on development side also. In this case,rapid release does not always help. Small utility softwares like music players, web browsers, etc. are easier to replace, since they are mostly freely available. With the release of newer version, the older version is not maintained. They might be available but are not supported or updated. In certain cases, one or two minor updates are made to a version before the next version is released. This is done to reduce the development cost, since the users do not pay for the software, maintaining multiple versions become a costly affair. The concept of online updates and bug fixes is largely being replaced with rapid release of new version, which allows developers to work on a single version at a time and thus control the development cost.
Large softwares, on the other hand, are not so easy to replace, like Microsoft Windows, Adobe Suite, etc. Generally, unlike Linux OSs, a heavy cost is associated with them, both at the user end and at the development end. Due to the huge cost involved it is not possible to abandon the older version with the release of new version. Sometimes more than one older version is needed to be maintained, which increases the development cost, which in turn is recovered by increasing the cost of the product in the market. Since cost plays an important role in determining the product's market share, so it cannot be increased beyond a reasonable limit.

The chances of introduction of more bugs in rapid release will upscale the need for quick updates and patches over the internet. These updates collectively, over a period of time, can be very large in size. The users with limited bandwidth, if they want to update, will have to pay the extra charges to their internet vendor, a further increase in overall cost of product. Also, sometimes extra hardware is needed to fully utilize the new features like touchscreen, this new hardware introduction adds to the cost for the users. Same is true with gaming consoles and mobile accessories, for different games different consoles might be needed for better experience and this definitely has a cost.

E-waste: Today e-waste is growing exponentially because the market of electronic products is growing exponentially. The rapid innovations and replacement of technologies is also causing the increase in e-waste. The rapid release of a product sometimes renders a not-so-old product useless by offering better features in same and sometimes lower price. This trend can be easily observed in tech-gadgets, like mobile phones. With very quick releases, by various companies and each one trying to take lead on the other, they create an environment where the consumers' desire to buy the new gadget is aggravated. This causes the rapid replacement of not-very-old gadget which in turn is, mostly, not disposed off properly.

Similarly, touch screen enabled monitors are destined to cause large scale replacement of traditional monitors, of which a large portion would be not-so-old.

Recycling sometimes require specialized techniques to maximize resource recovery and minimize harm to humans and environment. However, the use of such specialized techniques is rare, with much of the world's e-waste travelling to far distances from their region of origin, mostly to developing and poor countries, where old and less efficienttechniques are utilized for recycling purposes. These old techniques pose dangers to workers and their local natural environment, also they are very poor in resource recovery where much of the metal is discarded and inevitable lost. Rapid software and hardware upgrades do represent the advancements in technologies but their contribution in generating e-waste cannot be discarded.

7. POSSIBLE REMEDIES

Rapid release cycle raises concerns for everyone involved with the product, the organization, the developer and the user. In this section few possible remedies are discussed which can help in reducing the severity of problems. The remedies discussed are not absolute and may differ in effect among organizations, but they can be tailored to suite organizational needs.

For normal users, these free updates can cost them in terms of bandwidth, therefore, users should refrain from frequently updating their softwares without any major needs. They should make it a habit of updating their softwares only after a considerable amount of time has passed, in most cases annual or semi-annual updates may suffice. It is expected that software or hardware should be upgraded only when it starts hampering the productivity of the user, when incompatibility and crash rates increase beyond a certain level or if there is some real need of some new feature.

Frequent release of new versions also causes trouble for developers of supporting softwares, most of the time theadd-ons and plugins fail with the new version. Organizations developing large and popular softwares cannot consider their needs due to their large numbers and also the fact, that most of them are hobbyists, organizations limit themselves to providing the newer tools for plugins development. Therefore, it demands some kind of industry accepted standard or framework pertaining to release cycles which could benefit the developers and users at large. Large organizations, both hardware and software, should come together and lay out some guidelines which could bring some order to the release cycles across the market. The criteria can be developed based on the release cycle durations which, for example, can be of four types:

Rapid release cycle: Time duration can vary between 5-10 months. Dedicated (not enthusiasts) supporting app and small software developers would like to stick to this cycle. Also, new start-ups would like to work on this cycle to gain more penetration in the market in the beginning before moving to medium or slow cycles.

Medium release cycle: Time duration can vary between 12-24 months. Web and web application developers, system software developers can stick to such cycle.

Slow release cycle: Time duration can vary between 24-36 months. Gaming industries and certain hardware manufacturers may find this cycle more suitable.

Slower release cycle: Time duration can vary between 40-50 months. Large software developers, like operating systems, and enterprise solutions vendors may find this cycle more suited to their as well as customers' needs.

The policies regarding maintaining or not maintaining a product after release of newer version can be laid out and similar other issues can be analyzed. This will reduce the concerns of enterprises in maintaining their IT infrastructure. As for the e-waste management, much work has to be done which will involve the active participation of the people, the organization and the government. Without effective policies in place and strict monitoring it would be difficult to control the harm from e-waste. All the organizations should actively participate in e-waste management and recycling. Organizations should utilize recycling services from only those vendors who deploy latest technologies for recycling. Certain devices like mobile phones and graphic cards can be used for longer periods than they are actually used, in such cases marketing of pre-owned gadgets should be encouraged at large, this will increase the product life and will help in reducing e-waste to considerable extent.

8. CONCLUSION

Rapid release cycle has picked up fast and is definite to stay due to the varying user demands and increasing competition among companies. In further years, the issues associated with rapid release cycles will further aggravate, if not taken care of. It will require a combined effort of the organizations and customers to contain the situation. It is proposed to develop an industry accepted framework for release cycles based on certain parameters, which will require further research and joint effort of organizations.

9. REFERENCES

[1] Why Mozilla's Firefox Rapid Release Cycle Works and Why it Doesn't. http://www.internetnews.com/blog/skerner/why- mozilla-firefox-rapid-release-cycle-works.html

[2] The Case For Rapid Release Cycles. http://bradley-holt.com/2011/08/the-case-for -rapid- release-cycles/

[3] Firefox extended support will mitigate rapid release challenges. http://arstechnica.com/business/2012/01/firefox-extended-support-will-mitigate-rapid-release- challenges/

[4] Firefox rapid release cycle might improve security. http://www.netmagazine.com/news/firefox-rapid-release-cycle-might-improve-security-111577

[5] As they move to rapid releases, browser bosses bruise IT. http://arstechnica.com/business/2011/09/the-webs-rapid-release-cycleand-how-it-departments-can-tame-it/

[6] Understanding the Corporate Impact. http://mike.kaply.com/2011/06/23/understanding-the-corporate-impact/

[7] Is Web Browser Development Moving too Fast? http://application-performance-blog.com/is-web-browser-development-moving-too-fast-effects-on-browser-testing/

[8] The Growing Importance of the Rapid Upgrade Cycle. http://ostatic.com/blog/the-growing-importance-of-the-rapid-upgrade-cycle

[9] Macintosh Justification. http://www.linuxinsider.com/story/39826.html

[10] 2008 Review of Directive 2002/96 on Waste Electrical and Electronic Equipment (WEEE). http://ec.europa.eu/environment/waste/weee/pdf/final_rep_unu.pdf

[11] eCycling. http://www.epa.gov/epawaste/conserve/materials/ecycling/index.htm