Evaluation of the

TestProduct

Customer Configuration Updating Processes

Created for:
TestCompany
United States

Completed by:
Test Respondent

e-mail:
TestUser@cs.uu.nl

Prof. Sjaak Brinkkemper	s.brinkkemper@cs.uu.nl
Prof. André van der Hoek	andre@ics.uci.edu
dr. Slinger Jansen	slinger@cs.uu.nl
Wouter Buts Bsc.	wbuts@uci.edu

Institute for Software Research, UC Irvine.
Institute of Information and Computing.
Sciences.
University of California, Irvine

www.isr.uci.edu/
www.cs.uu.nl/groups/OI/
secr: +31.30.253.1454
Utrecht University

ISR Institute for Software Research
UNIVERSITY OF CALIFORNIA, IRVINE

Center for Organisation and Information
INSTITUTE OF INFORMATION AND COMPUTER SCIENCES Utrecht University

Table of contents

Chapter 1: Management Summary

This document is a generated report from the survey executed by the University of California, Irvine in cooperation with the University of Utrecht, Netherlands. This document concerns the release, delivery, deployment and usage of software products. The survey is completed by 78 organizations and the results of these companies are compared to the results of the product TestProduct, for the company TestCompany.

Based on your companie's survey answers we will give you possible "quick win" process improvement advice. These are related to your processes score on the different Customer Configuration Updating Processes.

Based on your companies' release, delivery, deployment and usage processes your product scored as indicated in the tables underneath. The green cells indicate that your process scores better than average. The red cells indicate that your process performes under average.

CCU Process	Your score	Max respondent score	Average Score
Release	24,0	41,0	24,7
Delivery	14,0	18,0	10,3
Deployment	25,0	51,0	29,6
Usage	18,0	43,5	17,6
All processes	81,0	153,5	82,2

Table 1: CCU Process scores.

Theses process scores can be compared to the average and maximal scores per industry:

	System man./business productivity		Home use		Cross industry software		Utilities software		Vertical market	
	Average	Max	Average	Max	Average	Max	Average	Max	Average	Max
Release	26,1	41,0	18,5	32,0	27,5	41	24,5	41	33,0	36
Delivery	11,3	16,0	9,4	13,5	10,7	17,5	10,0	15,5	9,0	10
Deployment	31,4	49,0	23,9	37,0	36,8	51	31,6	48	36,0	49
Usage	18,9	44,0	10,4	30,0	23,2	44	19,3	40	31,0	44
All processes	87,7	150,0	62,2	112,5	98,2	153,5	85,5	144,5	109,0	139,0

Table 2: Average and max CCU scores per industry.

Based upon the TestProduct answers we give your organization the following advice per process:

Release	
Formal release planning:	Use formal release planning.
Publishing and accesibility of planning:	Publish a formal release planning to all stakeholders.
Release customer convenience:	Process is OK
Explicit management of tools:	Process is OK
Repository use:	Save components in a release database or source tree.
Delivery	
Product download from every location:	Process is OK
Inform your customer frequently:	Get in touch with the customer more often.
Product availability:	Process is OK
Deployment	
Update tool usage:	Introduce a installation or updating tool.
Explicit dependency management:	Process is OK
Data management (seperation data and product):	Process is OK
Customer Configuration completeness:	Process is OK
Cope with customizations, extensions and customer specific solutions:	Make sure your updating tool is able facilitate in making customer specific changes.
Updating during runtime:	Create the possibility to update during runtime.
deploy in DTAP environment:	Process is OK
Usage	
Usage license agreement:	Process is OK
Usage report generation:	Make use of Usage reports.
Licence extention:	Process is OK
Temporary licenses:	Process is OK
Automatic license generation:	Generate licenses automatically form sales contracts.
Automatic error report:	Send automatic error reports from the customer side.
Knowledge hard/software platforms:	Process is OK

Table 3: CCU practice assessment.

These advices are elaborated upon in chapter 3.

With regard to the questions which development technologies TestCompany uses for the product TestProduct you answered:

Program Languages		
Language	Used	Percentage
C	No	0
C++	No	0
Java	Yes	100
dotNet	No	0
ASP	No	0
PHP	No	0
Perl	No	0
Pascal(Delphi)	No	0
Basic(Visual)	No	0
C#	No	0
Progress	No	0
Modula2	No	0
Foxpro	No	0
Oracle	No	0
Clarion	No	0
Python	No	0
Other	No	0

Table 4: Pogram language usage.

The product Application Express is classified in the company category:

Industry	
Business productivity	Yes
Home use	No
Cross-industry software	No
Utilities software	No
Vertical market	No
Other	No

Table 5: Active industries.

In the next chapter you will be introduced to the Customer Configuration processes. This will enable you to get a better understanding of the advices for change dispenced earlier in this document. This chapter includes a generic overview of all the participating companies in the CCU survey as well. In the third chapter we will also elaborate further on the change advice generated for the product TestProduct. In chaper four your survey answers are compared to those of other product software companies.

This enables you to make a precise comparison between the differences in your processes competitors in the same market. In chapter 5 we will reveal what other product software companies have answered to the different open questions. In chapter 6 a detailed overview of your score is shown and the total of points that can be earned for every answer. Finally, in chapter 7 an overview of literature used to create this report is presented.

Chapter 2: Proces Description and Overall Results.

In this section the customer configuration updating processes are modelled. This model explicitly defines customer actions, enabling a software vendor to improve managing and predicting the practices that need extra focus. Much akin to the CMM, the model uses the concepts of practices, features, and process. Practices stand for practices of a software vendor which enables capabilities. Capabilities are defined as a property of a process that improves product quality and quality of service. Each process identifies a cluster of related capabilities which, when performed collectively, achieve a set of goals considered important for enhancing process overall capability. A software vendor enables a capability within a process, once it responds correctly to one of these customer triggered actions.

To describe the practices for CCU, its processes need to be established. These processes are found using a previously established model for software updaters that focuses on the customer. Due to the fact that software maintenance and deployment focuses solely on the customer, the model is extended with the organisational interactions that are required to fully support a customer's actions after an update is released. The CCU model, as depicted in Figure 1, displays several states a customer can move through after a product or update release on the right side. On the left side, the organisational structures that facilitate interaction are displayed. Within the CCU model four processes are distinguished: release, delivery, deployment, and the usage process. The processes are separated by dotted lines in figure 1 and are further described in the sections below.

Processes in the model are triggered by customer actions. These actions are becoming aware of, downloading, deploying, reconfiguring, activating, and deactivating the release. When a vendor receives a customer request, the customer relationship management (CRM) system is used to identify the customer. The vendor then handles the request and interacts with the customer. The customer moves through a number of states when about to update its configuration. At first the customer is unaware of the update, until the customer requests information about a product. Once received, the customer hopefully downloads, deploys and activates it for use. In the mean time he communicates with the vendor by means of software, licenses, feedback, and product knowledge. For more information please see the following article:

Slinger Jansen and Sjaak Brinkkemper. **Definition and validation of the key process areas of release, delivery and deployment of product software vendors: turning the ugly duckling into a swan**. In proceedings of the International Conference on Software Maintenance (ICSM2006, Research track), September 2006. Also; see chapter 7 for more information.

2.1 Proces description

We define CCU as the release, delivery, deployment, and usage processes of a software vendor. These processes consist of two to four practices, each with a number of elementary capabilities. For instance, the release process is made up of four practices. One of these practices is release frequency and quality. The capabilities residing under the release frequency practice are; a vendor must frequently release major, minor, and bugfix releases; a vendor must synchronize these releases with customer demand, and releases are tested by pilot customers before they are made publicly available.

The **Release process** is based on four release practices. The first practice is how often versions and updates of a product are released and how this is planned within the organization. The second practice is how releases are shared within the company and between customers and the software vendor. Thirdly, all components; be they external or internal products and components, or products that are required for the development process alone, must be managed by making explicit dependency between these products and components. Finally, versions of external components, such as Components off the shelf (COTS), must be managed explicitly to maintain high quality releases.

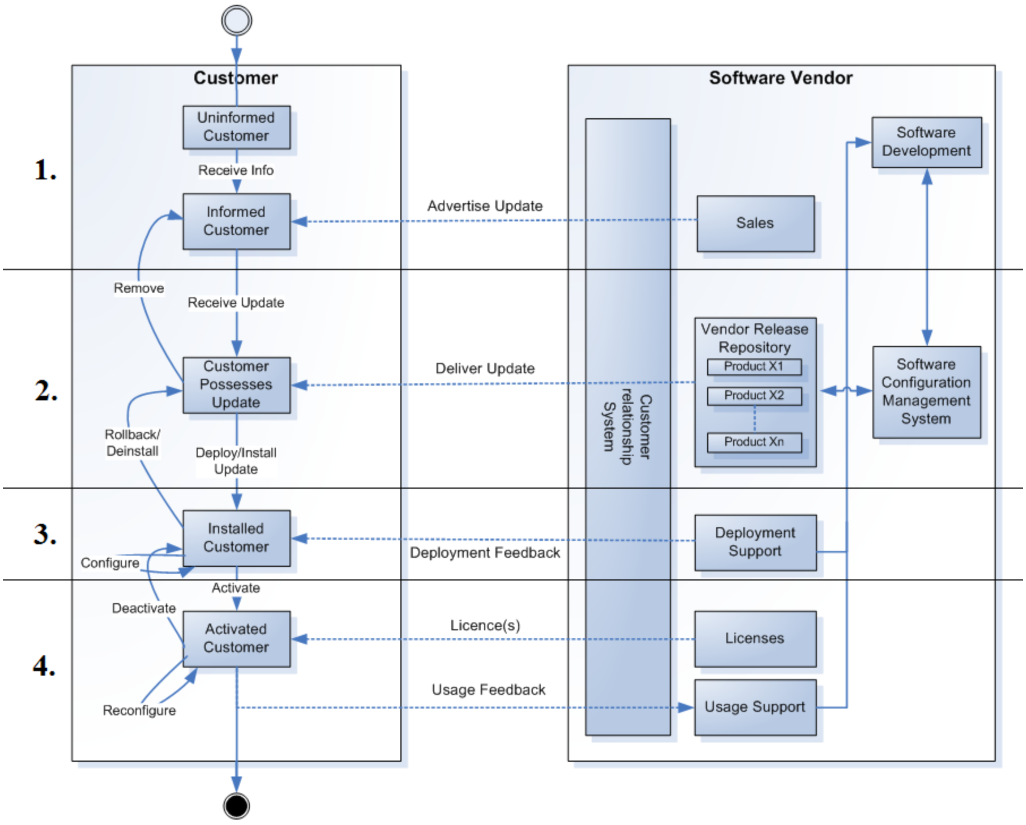


Figure 1: The Generic CCU model (based on Jansen, 2006).

With regards to the **delivery process** there are two practices: the first practice prescribes that vendors must use every possible channel for the distribution of products and product updates. The second practice states that every possible method for delivery must be applied, such as automatic push or pull.

There are four practices for the **deployment process** of a vendor. To begin with, a product must be removable without leaving any remnants of data on a system. This is required because a new installation preferably must not be contaminated with old data. Secondly, if issues are encountered during the deployment of a software product automatic resolution must take place to resolve these issues. Such resolution mechanisms can be automatic downloading of missing components, freeing up resources when required, or even automatic renewal of licenses. The third practice for the deployment practice is that updates and installations must be able to cope with customisations made by customers or third parties. A vendor is mature in this practice when a special software architecture is in place that enables customisations. The fourth practice is deployment reliability, which is can be ensured by validity checks, feedback reports, and externalisation of customer specific changes and data.

Finally, a vendor's **activation** is based on three practices. First, a vendor must (semi-) automatically handle all license requests and distribute licenses with a maximum amount of flexibility within the organization. A vendor has adequately implemented this practice once customers are able to explicitly manage their licenses, licenses expire. Temporary licenses can be generated for sales and test purposes, and licenses follow automatically once a sales contract is signed. Secondly, vendors must make use of feedback to gain as much knowledge about the product in the field as possible. A vendor is considered adequate for this practice once he makes use of both usage reports and error feedback. The third practice is that a vendor must be aware of his customers' configurations. A vendor scores for this practice when he is aware of the components most used by customers and the operating environments and hardware of customers.

2.2 Reseach set-up and Respondents

The respondents that have been selected are based on 2 criteria: First the submitter must be a product manager or a development manager who is close to the process and who knows answers to each question. Secondly, the software product must be specifically a software product that is delivered to customers and executed at their site. These requirements are specified in the invitation e-mail, the invitation letter, the general CCU benchmark website and at the beginning of the benchmark survey itself.

The vendors have been selected through the Web Platform CM Crossroads.com, LexisNexis packaged software index, the yellow pages, and the connections through the University of California, Irvine (http://www.isr.uci.edu/). The potential respondents have been approached by e-mail. None of the respondents completed the survey more than once.

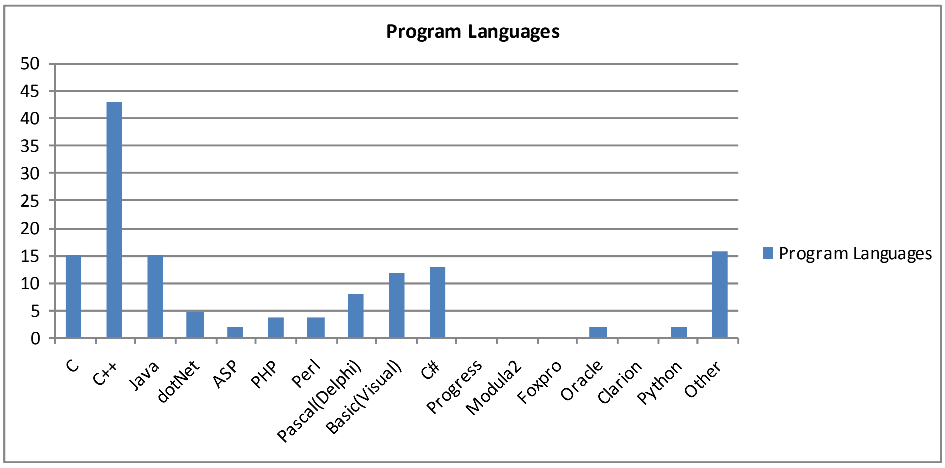


Figure 2: General program languages used.

In total 78 product managers submitted the survey, from software vendors ranging from 1 to 100.000 employees. 6 companies were excluded from the dataset for not actually delivering product software. The biggest part of the respondents are active in the business productivity industry. The vertical markets industries earn a second place.

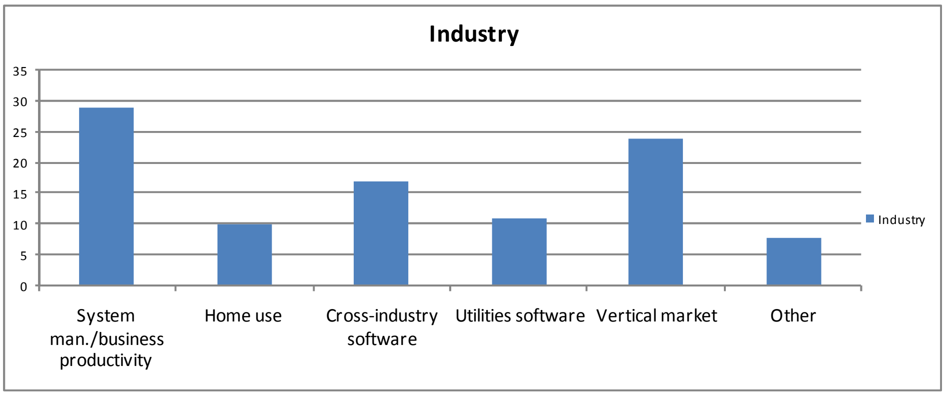


Figure 3: General active industries.

Finally; smaller companies turned out to be responsible for the biggest part of the dataset.

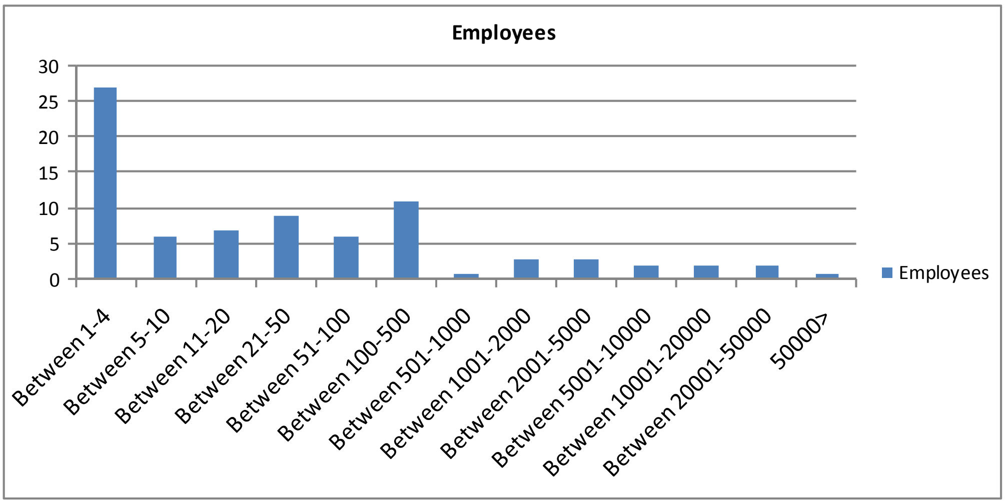


Figure 4: General numbers of employees.

Chapter 3: CCU advice for process improvement.

Now some advices with regard to your CCU process will be presented. An interesting fact is that you think your process is: More advanced to/than products of competitors.

This advice is tailor made for the TestProduct product. Mapping the processes on which your product scores low or below average results in the process revision advice as presented below.

3.1 Release

Table 6 compares your CCU delivery score against the survey average and the higherst possible score in this process area. When you score above average you are doing well. When your process scores below average on a particular practice we advice you to invest in improvements.

Release Proces Practice	Your Score	Average Score	Highest Possible Score
Release frequency.	0	1,46	11
Release planning.	4	5,33	13
Release scenario.	9	9,60	13
Release management.	11	8,33	15
Total	24	24,73	52

Table 6: Release process scores.

With regard to the Release process we give you the advice depicted in Table 7. When the column on the right indicates that the process is OK you have implemented enough release process practices in this area. When this is not the case an advice will be distributed to improve on this particular practice. Reasons for improvement are included in this advice, this enables you to understand why you should arrange resources to implement these advices.

Release	Formal release planning	Use formal release planning. A formal release planning is a usable tool when planning release dates, efforts and release process improvement. A formal release planning consists of a few quality checks, creating a bill of materials, packing the release and related marketing activities. When this overall planning is formally completed it will be possible to improve, delete and automate parts of this process. Such a planning can contribute to a better time to market for a specific product feature.
	Publishing and accesibility of planning	Publish a formal release planning to all stakeholders. Publishing a release planning for all stakeholders is very valuable. This way: a sales person knows when new features will be implemented and usable, a developer knows when he/she can plan his/her vacation and marketing colleagues know when to start new champagnes. It is advisable to construct a publication policy, this way confidential information will not prematurely leave the company.
	Release customer convenience	Process is OK
	Explicid management of tools	Process is OK
	Repository use	Save external components, which are delivered with the product, in a release database or source tree. By doing this you make sure that you always know what components and component versions you need during building, testing and integrating the product. An easy way to do this is to use an integration build system, like Sisyphus or Cruise Control.

Table 7: Release process assessment.

3.2 Delivery

Table 8 compares your CCU delivery score against the survey average and the higherst possible score in this process area. When you score above average you are doing well. When your process scores below average on a particular practice we advice you to invest in improvements.

Delivery Proces Practice	Your Score	Average Score	Highest Possible Score
Delivery distribution channel.	9	5,90	15
Delivery distribution method.	5	4,42	13
Total	14	10,32	28

Table 8: Delivery process scores.

With regard to the Delivery process we give you the advice depicted in Table 9. When the column on the right indicates that the process is OK you have implemented enough delivery process practices in this area. When this is not the case an advice will be distributed to improve on this particular practice. Reasons for improvement are included in this advice, this enables you to understand why you should arrange resources to implement these advices.

Delivery	Product download from every location	Process is OK
	Inform your customer frequently	Make your product available through the web. When your product is downloadable and testable it will create potential for a greater client base. Furthermore, you give your customer the possibility to acquire your product, updates and its content in a more flexible way.
	Product availability	Process is OK

Table 9: Delivery process assessment.

3.3 Deployment

Table 10Table 8 compares your CCU delivery score against the survey average and the higherst possible score in this process area. When you score above average you are doing well. When your process scores below average on a particular practice we advice you to invest in improvements.

Deployment Proces Practice	Your Score	Average Score	Highest Possible Score
Deployment dependency management.	0	4,46	8
Deploy removal.	7	5,31	7
Deployment customization management.	3	4,77	14
Deployment product reliability.	15	15,01	22
Total	25	29,55	51

Table 10: Deployment process scores.

With regard to the Delivery process we give you the advice depicted in Table 11. When the column on the right indicates that the process is OK you have implemented enough delivery process practices in this area. When this is not the case an advice will be distributed to improve on this particular practice. Reasons for improvement are included in this advice, this enables you to understand why you should arrange resources to implement these advices.

Deployment	Update tool usage	Introduce a installation or updating tool. In order to improve your CCU processes a great deal you should start using such a tool. Such a tool (like InstallShield, FlexNet or RPM) can make the installation process reliable. There are freeware tools available for these kind of tasks as well, such as: Wix, InnoSetup and Pheme. All thee tools are capable to support the installation and updating process on different platforms. Some of these tools even offer extra functionalities like automatic error resolution, and knowledge management.
	Explicid dependency management	Process is OK
	Data management (seperation data and product).	Process is OK
	Customer Configuration completeness	Process is OK
	Cope with customizations, extensions and customer specific solutions	Make sure your updating tool is able facilitate in making customer specific changes. This way, these changes will not be harmed or undone by future updates. When your update tol is not able to cope with customer specific changes, there is a great risk that the customer will lose his specific settings through an update. This is hard to match with the concept of platform thinking, where the overall vision is, that a product is more successful when third parties are able to build extensions upon a software product. In this way a common interest is created, which boosts the market position of the product.
	Updating during runtime	Create the possibility to update during runtime. Updating during runtime can be of different scales of importance to a company. Where an ASP organization cannot handle a single moment of downtime, a more traditional organization with monolithic applications will have less problems in this area. This does not eliminate the importance of automatic updating. A customer who can't update at runtime may demand to always have the version with the latest security files on his system. Research by Jansen & Brinkkemper even showed respondents that never shut down the project, which made it impossible for them to update the product.
	deploy in DTAP environment	Process is OK

Table 11: Deployment process assessment.

3.4 Usage

Table 12Table 8 compares your CCU delivery score against the survey average and the higherst possible score in this process area. When you score above average you are doing well. When your process scores below average on a particular practice we advice you to invest in improvements.

Usage Practice	Your Score	Average Score	Highest Possible Score
License usage.	8,0	7,6	16,0
Organizational license management.	6,0	2,1	9,0
Usage feedback/product knowledge.	4,0	7,9	19,0
Total	18,0	17,64	44,0

Table 12: Usage process scores.

With regard to the Delivery process we give you the advice depicted in Table 13. When the column on the right indicates that the process is OK you have implemented enough delivery process practices in this area. When this is not the case an advice will be distributed to improve on this particular practice. Reasons for improvement are included in this advice, this enables you to understand why you should arrange resources to implement these advices.

Usage	Usage license agreement	Process is OK
	Usage report generation	Make use of Usage reports. With these usage reports you will be able to identify your most valuable product features, charge customers on usage base, and analyze errors. All product knowledge in the field is extremely valuable and should be treated as such. You should however keep in mind that constantly keeping up with usage will put performance pressure on your product.
	Licence extention	Process is OK
	Temporary licenses	Process is OK
	Automatic license generation	Generate licenses automatically form sales contracts. This will make your organization more efficient which makes it possible to deal with a greater customer base. It will also force you to build standardized customer models, which will result in less overhead.
	Automatic error report	Send automatic error reports from the customer side. By using these reports you will be able to analyze how many times your product crashes at the customer side, and you will be able to offer customer specific services more efficiently. It is very valuable when a customer receives a solution for a problem (trough an update) directly after encountering it! Microsoft statistics showed that 5% of the errors are responsible for 95% of the error messages. Finally, you will be able to make a better estimation to what extend your product is used illegally.
	Knowledge hard/software platforms	Process is OK

Table 13: Usage process assessment.

Chapter 4: Benchmark data CCU processes

4.1 Organizational indicators (Context)

Some interesting generic company charecteristics are visualized below:

	Product age	FTE Developers	Languages	Customers
Average	8	8	4	7
TestProduct	2	2	0	2

Table 14: Generic company charecteristics.

4.2 Company profile

Question 2: Please characterize you're job function.
Your answer: Product Manager

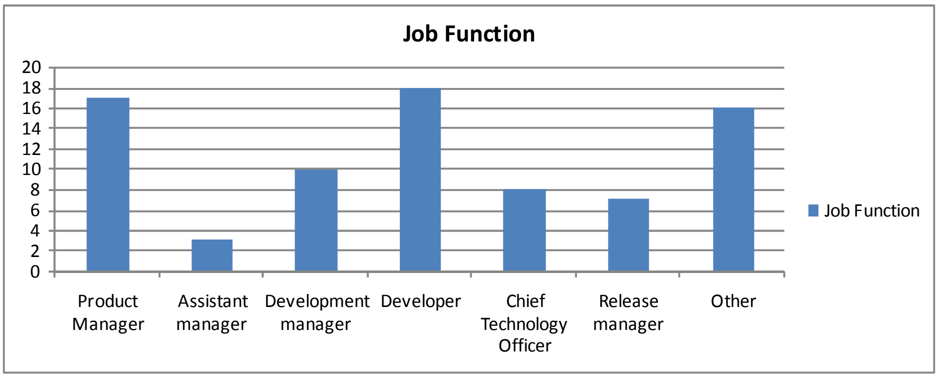


Figure 5: Job function.

Question 4: What is the name of your company?
Your answer: TestCompany

Question 5: Please provide the country where your corporate headquarters is located.
Your answer: United States
Other answers:

Headquarters	
Country	#
Argentina	1
Armenia	1
Aruba	1
Belgium	1
Bhutan	1
Bosnia and Herzegovina	1
Canada	1
China	2
Czech Republic	1
Denmark	1
Finland	1
France	1
Germany	1
Hong Kong SAR	1
India	1
Israel	2
Macedonia	1
Netherlands	2
Russia	1
Singapore	1
Spain	1
Sweden	1
Turkey	1
Ukraine	1
United States	52

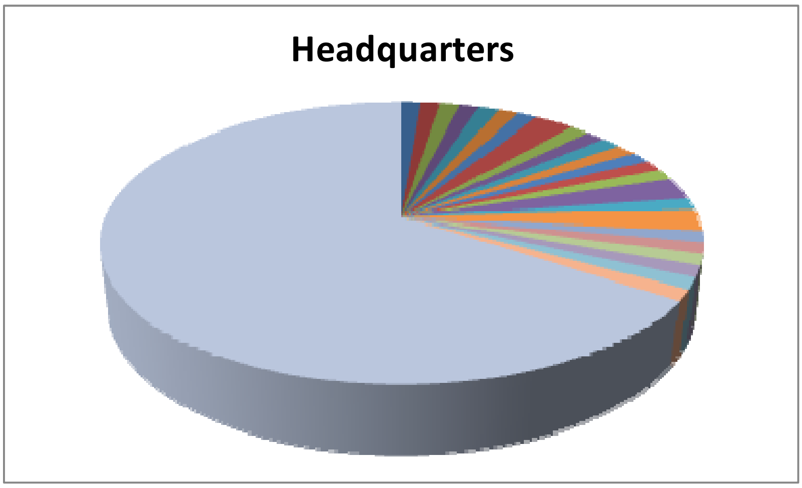


Figure 6: Location of corporate head quarters.

Question 6: Please indicate how many people are currently working at your company.
Your answer: 2000

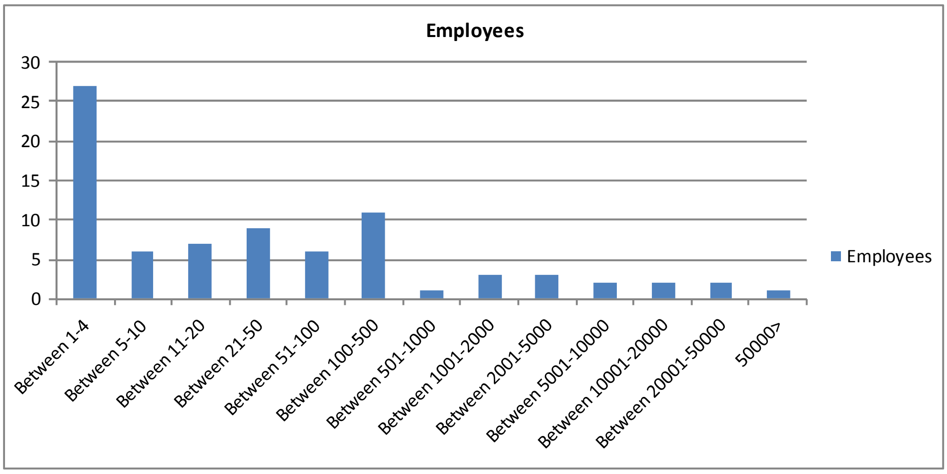


Figure 7: Number of employees.

Question 7: Please indicate the turnover scale your product generated in USD in the last closed book year. (Including Licenses / License maintenance and services).
Your answer: No answer

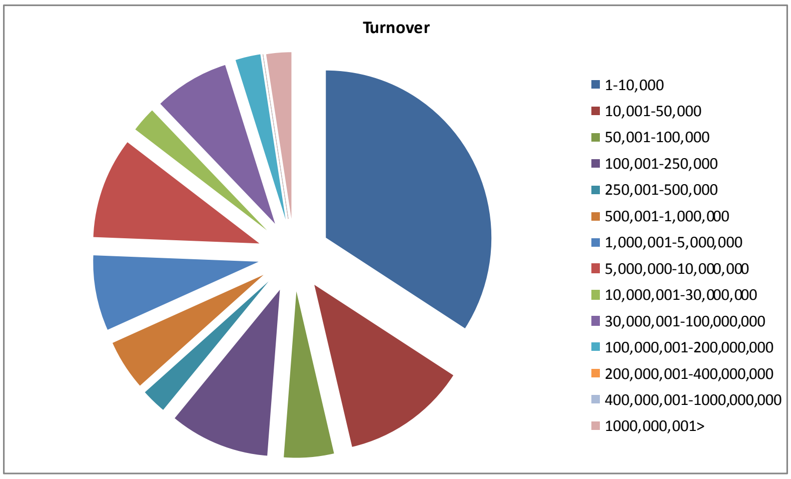


Figure 8: Turnover.

Question 8: What percentage of your last closed book year's turnover was generated by existing customers?
Your answer: No Answer

As Figure 9 shows, the majority of the respondents chose not to answer this question.

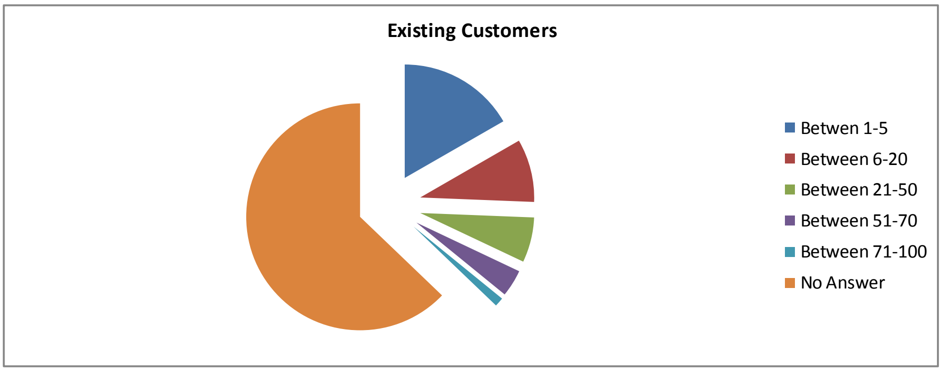


Figure 9: Turnover generated by existing customers.

4.3 Product Profile

Question 1: What is the name of your product?
Your answer: TestProduct

Question 2: Please indicate in what industry you are active.
Your answer:

Industry	
Business productivity	Yes
Home use	No
Cross-industry software	No
Utilities software	No
Vertical market	No
Other	No

Table 15: Active industry(ies).

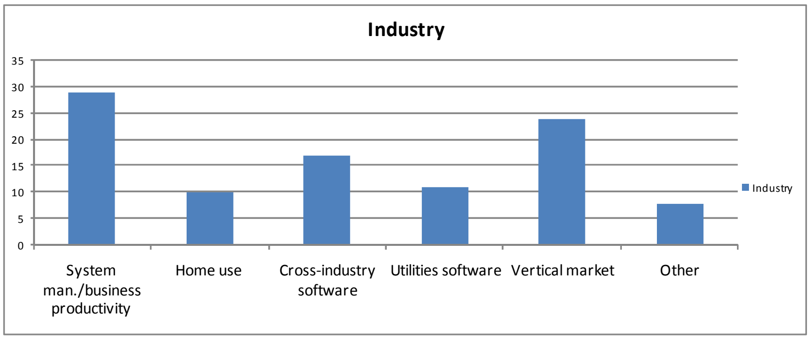


Figure 10: General active industries.

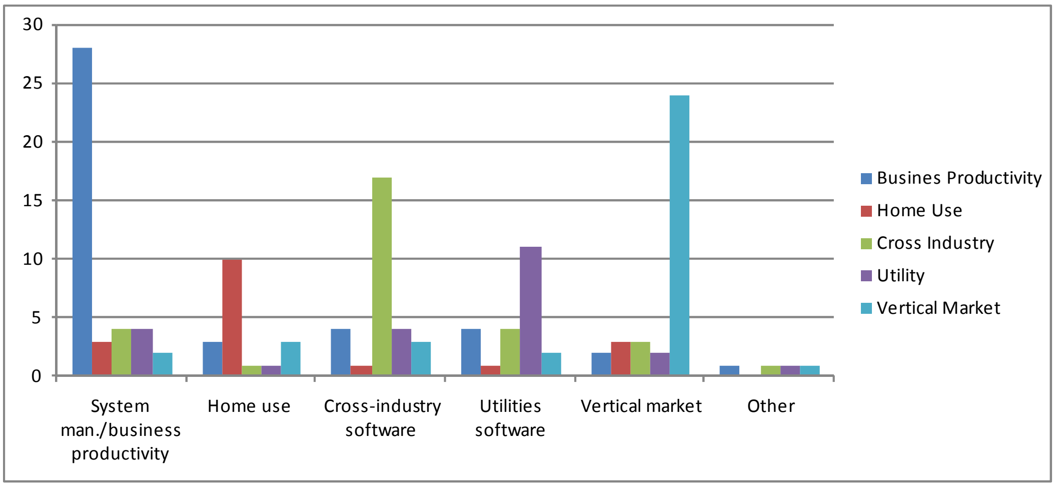


Figure 11: Activity in other industries.

Question 3: Please indicate in what/which economic region(s) your company releases her product.

Your answer:

Africa	No
America (North)	Yes
America (Central)	No
America (South)	No
Asia	No
Asia subset: China, India, Japan, Pakistan or Taiwan	No
Australia	No
Europe	Yes
Middle East	No

Table 16: Release per economic region.

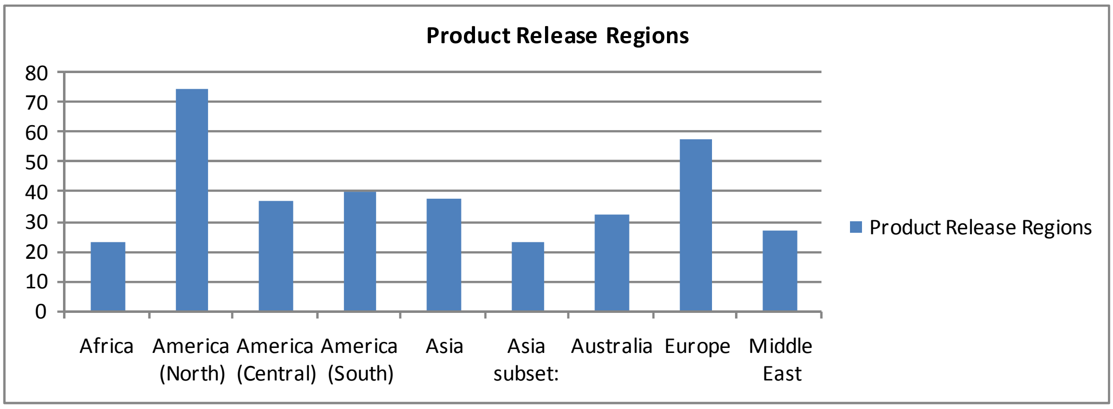


Figure 12: Release per economic region.

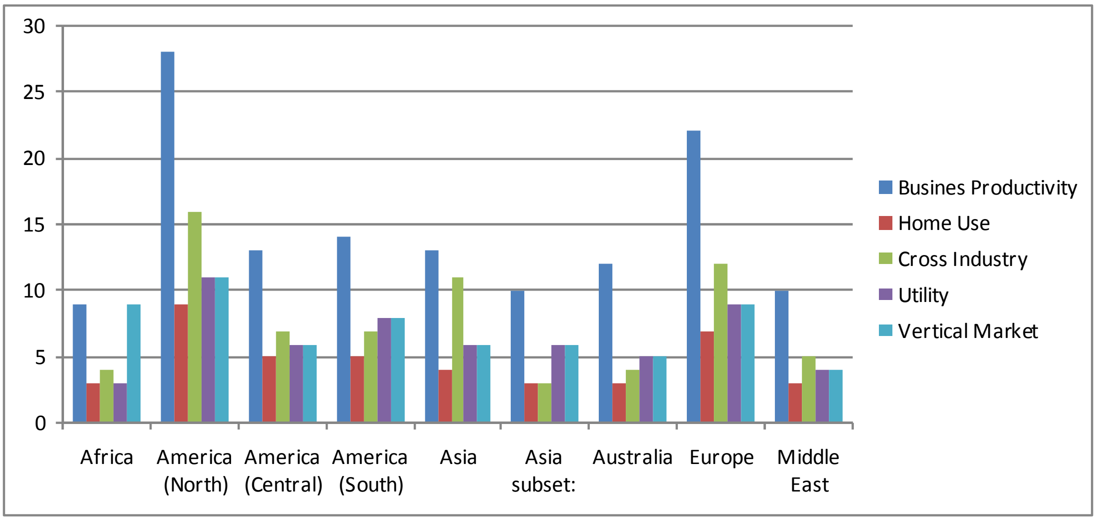


Figure 13: Release in economic region per industry.

Question 4: Please indicate in what/which economic region(s) your company builds her product.
Your answer:

Africa	No
America (North)	Yes
America (Central)	No
America (South)	No
Asia	No
Asia subset: China, India, Japan, Pakistan or Taiwan	No
Australia	No
Europe	No
Middle East	No

Table 17: Product building region(s).

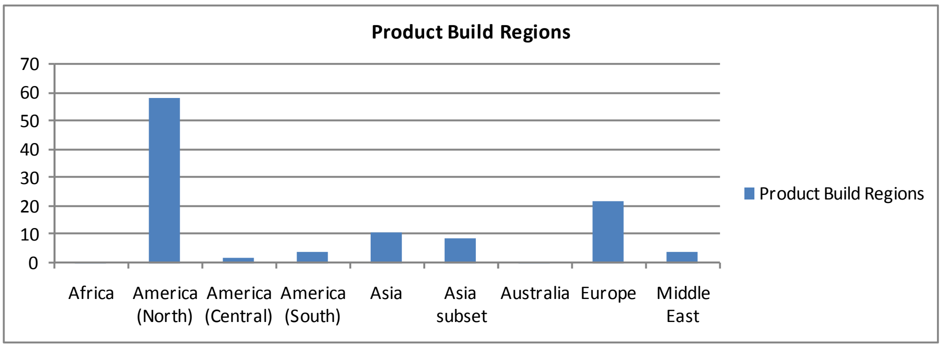


Figure 14: Product building region(s).

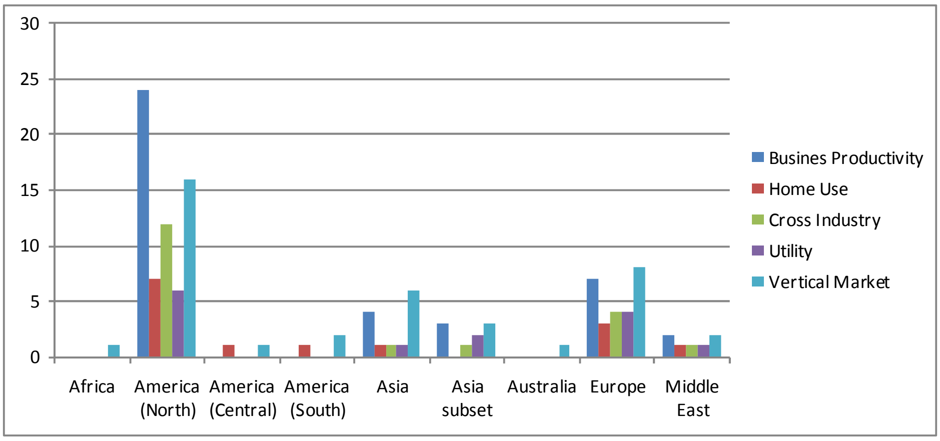


Figure 15: Product building region(s) per industry.

Question 5: Please indicate in what/which economic region(s) your product's property right is managed (e.g. our product is in European hands).
Your answer:

Africa	No
America (North)	Yes
America (Central)	No
America (South)	No
Asia	No
Asia subset: China, India, Japan, Pakistan or Taiwan	No
Australia	No
Europe	Yes
Middle East	No

Table 18: Regional management of property rights.

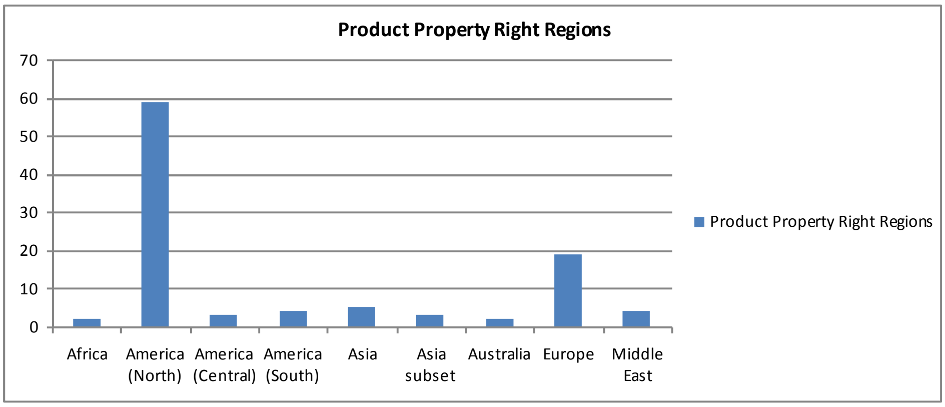


Figure 16: Regional management of property rights.

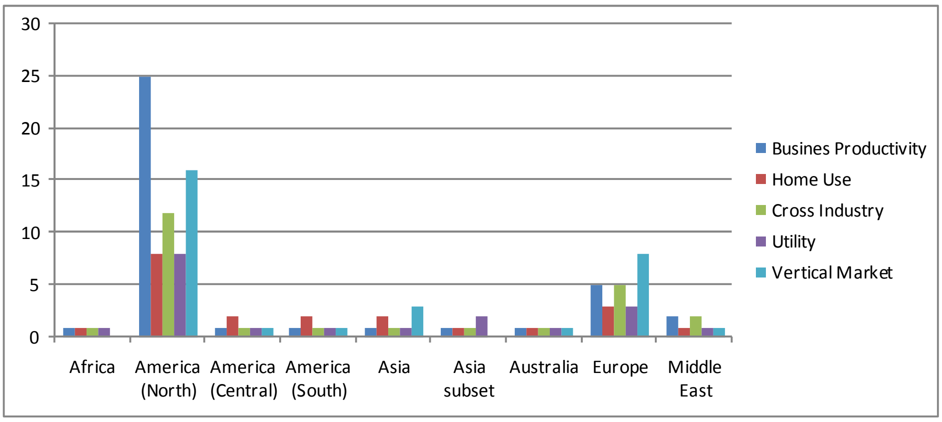


Figure 17: Regional management of property rights per industry.

Question 6: Please indicate how many customers you have for your product.
Your answer: Between 5-10

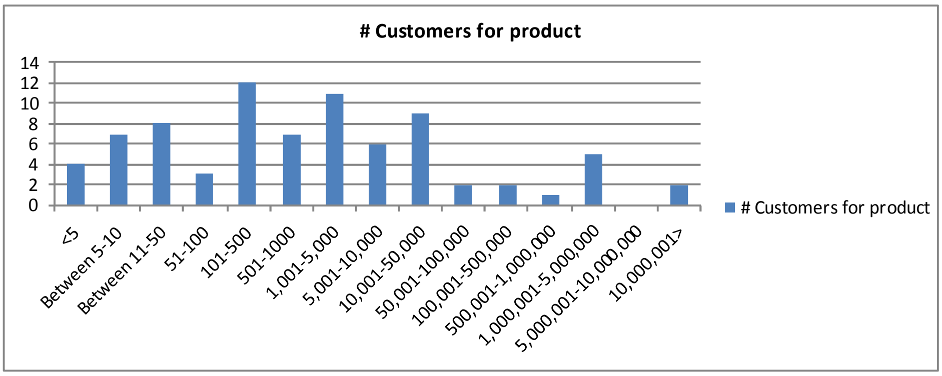


Figure 18: Number of customers for product.

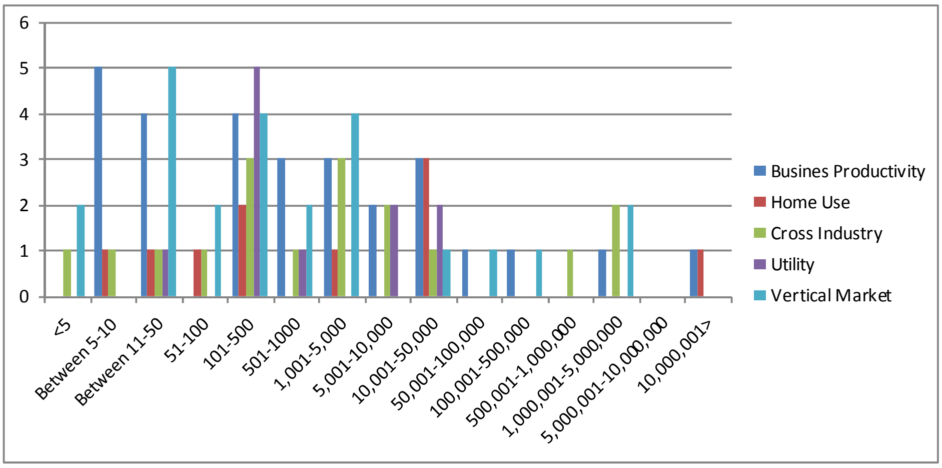


Figure 19: Number of customers for product per industry.

Question 7: Please indicate how many end users your product has presently.
Your answer: Btwn 11-50

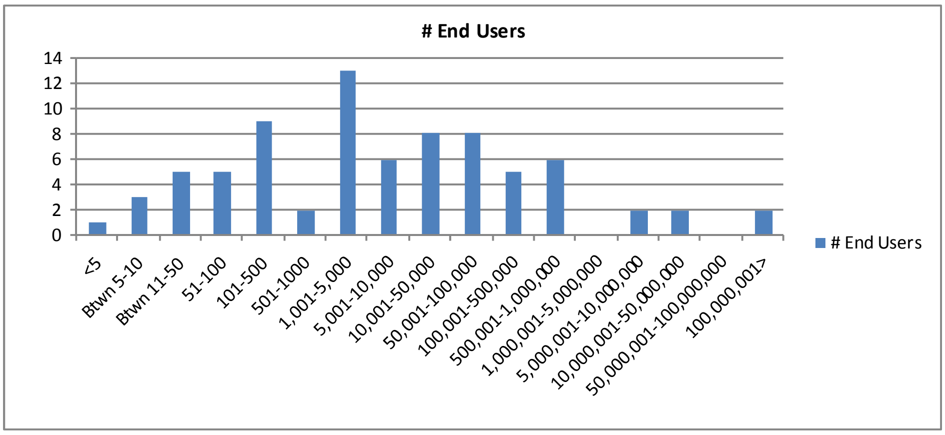


Figure 20: Number of end users.

Question 8: Please indicate of how many lines of code your full product consists in KLOC.
Your answer: 121-200

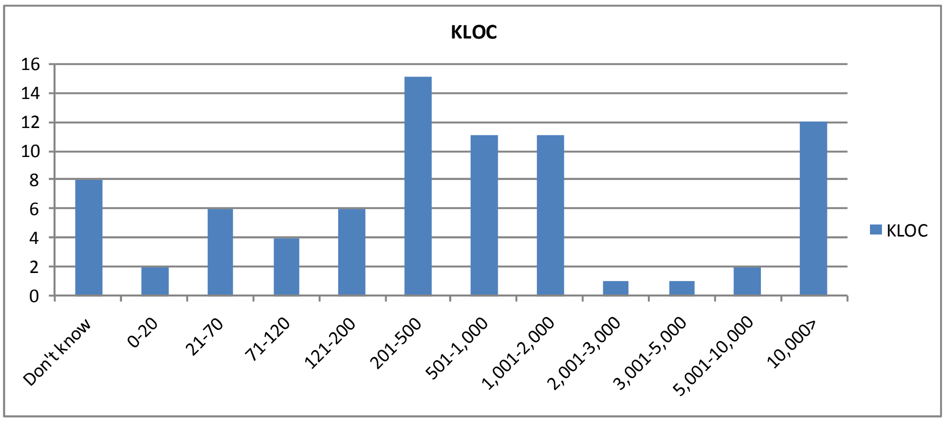


Figure 21: Number of KLOC.

Question 9: Please indicate in what programming languages and with which development technologies your product is built? Please indicate as well what percentage of your product is build with each language or technology in the textbox. % should add up to 100. If your answer is 'other' please provide the language and a percentage in the text field.
Your answer:

Program Languages		
Language	Used	Percentage
C	No	0
C++	No	0
Java	Yes	100
dotNet	No	0
ASP	No	0
PHP	No	0
Perl	No	0
Pascal(Delphi)	No	0
Basic(Visual)	No	0
C#	No	0
Progress	No	0
Modula2	No	0
Foxpro	No	0
Oracle	No	0
Clarion	No	0
Python	No	0
Other	No	0

Table 19: Program language usage.

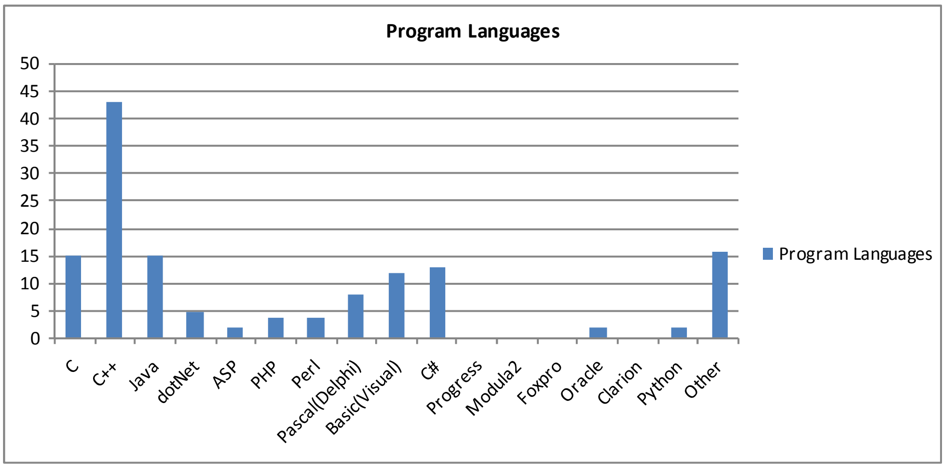


Figure 22: Program language usage.

For program language usage per industry please see the next page.

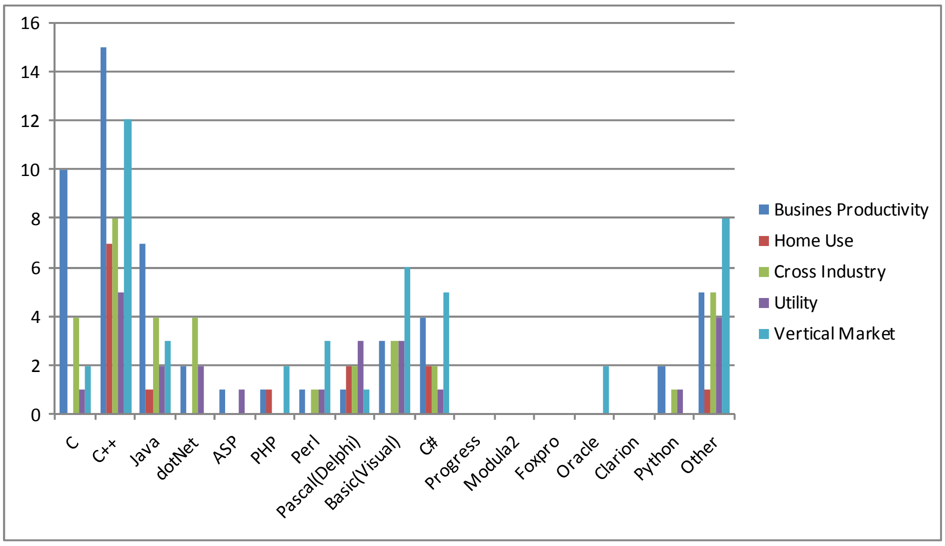


Figure 23: Program language usage per industry.

Question 10: In how many translations for international (non−English) markets is your product available?
Your answer: 0

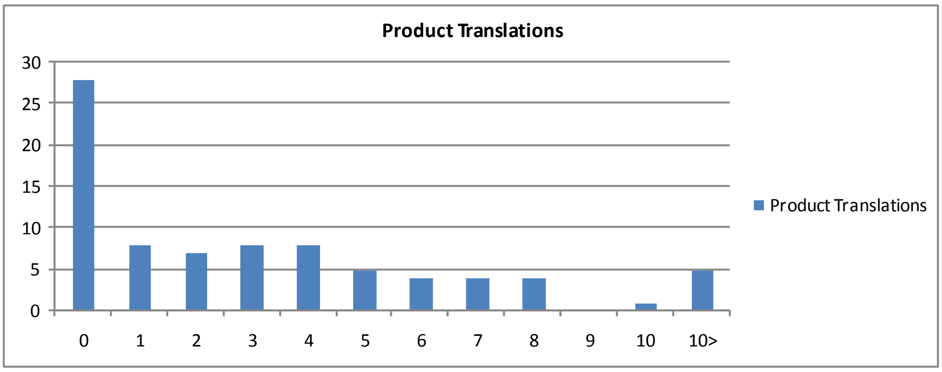


Figure 24: Number of non english translations.

Question 11: How many developers (in full−time equivalent) work on this product, at this moment?
Your answer: 2

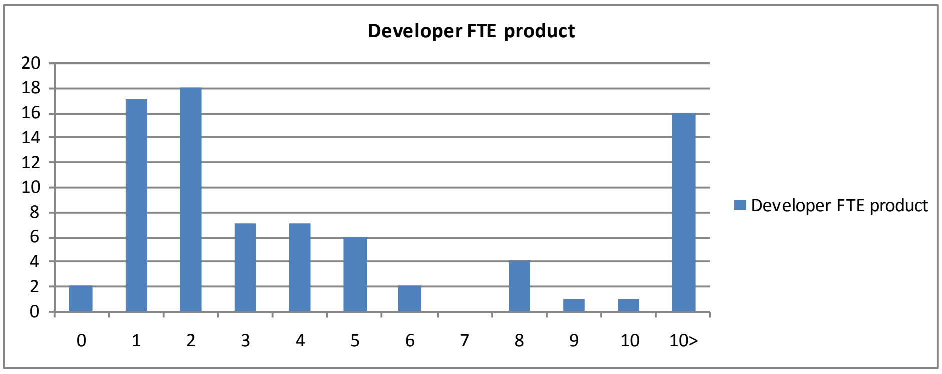


Figure 25: Developers (in full−time equivalent).

Question 12: How many years ago was the first line of code written for this product?
Your answer: 2

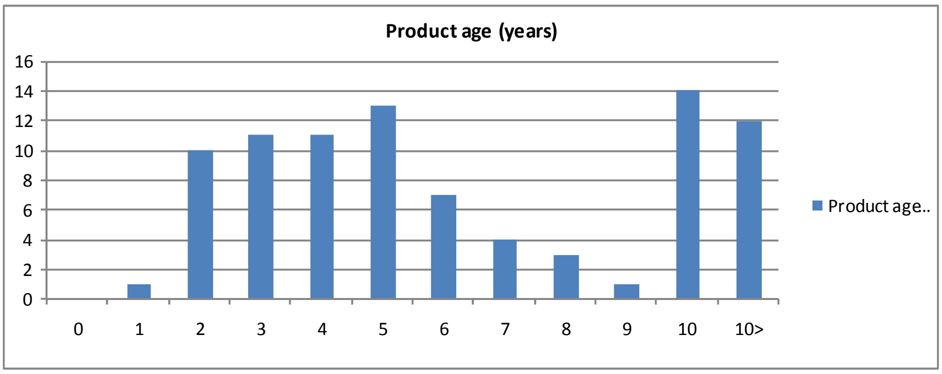


Figure 26: Product age (in years).

Question 13: What is the software architecture of the product?
Your answer:

Client-server	No
Service-Oriented Architecture	Yes
Stand-alone	Yes
Peer-to-peer	No
Web-based	No
Other	No

Table 20: Software architecture.

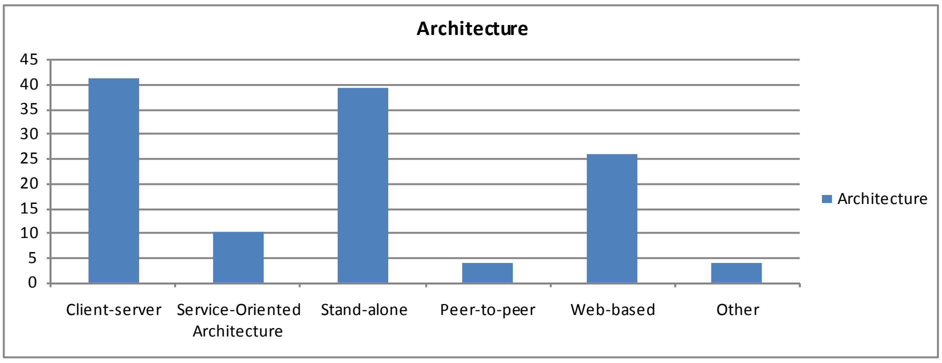


Figure 27: Software architecture.

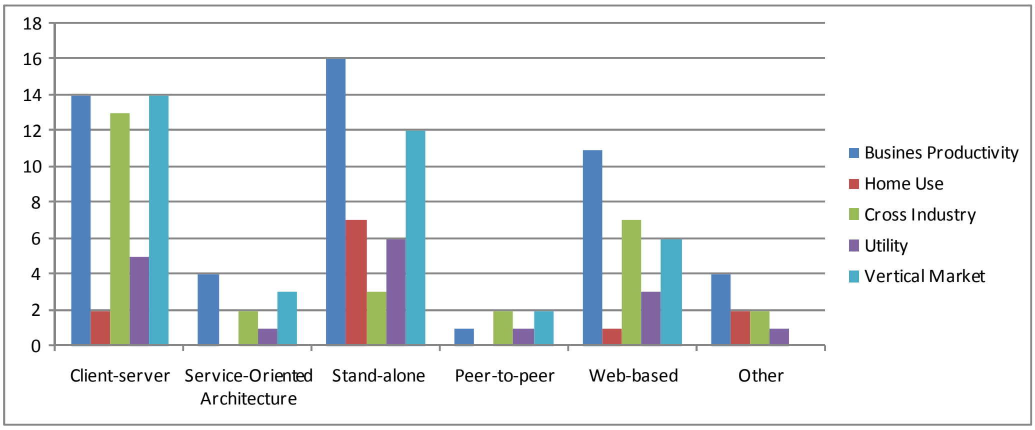


Figure 28: Software architecture per industry.

Question 14: Is your product Open Source?
Your answer: We use some open source components

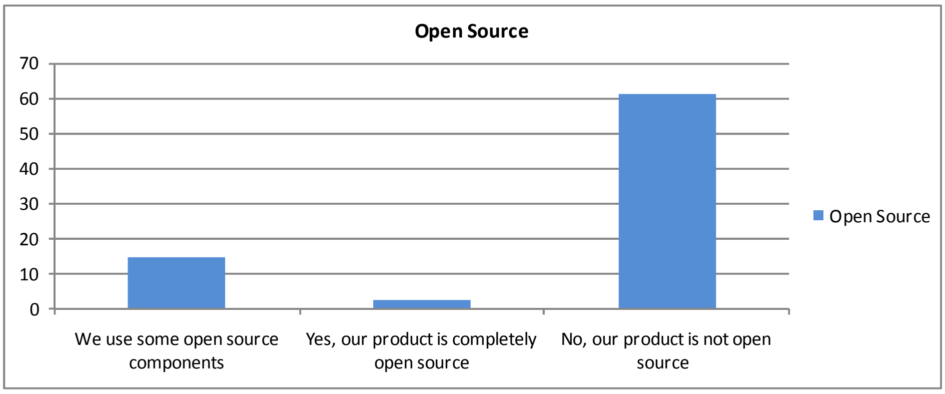


Figure 29: Open Source.

4.4 Release

Question 2: Please provide the dates of what your company would concider as the last three major release updates of your product (If youre answer indicats 0-1-1900, you did not provide an answer).

Your answer:

1	1-4-2008
2	21-8-2007
3	3-7-2007

Table 21: Major release(s).

Question 3: Please provide the dates of what your company would concider as the last three minor release updates of your product (If youre answer indicats 0-1-1900, you did not provide an answer).

Your answer:

1	1-4-2008
2	18-3-2008
3	22-2-2008

Table 22: Minor release(s).

Question 4: Please provide the dates of what your company would concider as the last three bugfix release updates of your product(If youre answer indicats 0-1-1900, you did not provide an answer).

Your answer:

1	26-3-2008
2	18-3-2008
3	5-3-2008

Table 23: Bugfix release(s).

Question 5: How many pilot or beta customers do you use to test the product?
Your answer: 0

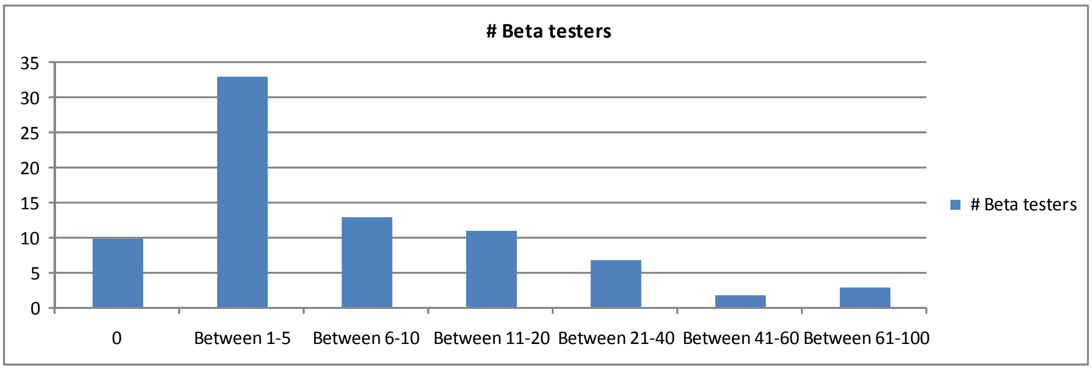


Figure 30: Number of beta testers.

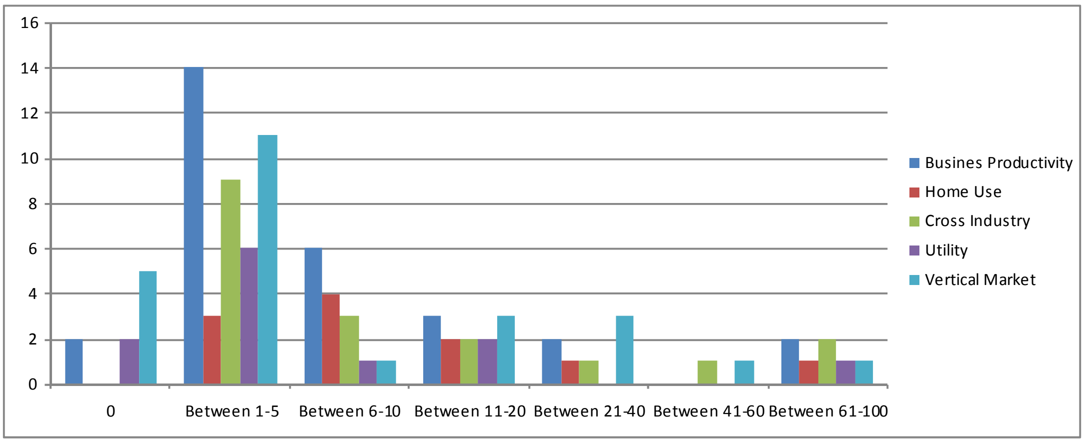


Figure 31: Number of beta testers per industry.

4.5 Release Planning

Question 1: Are updates and products released at times that are convenient with regard to your customers?
Your answer: Yes

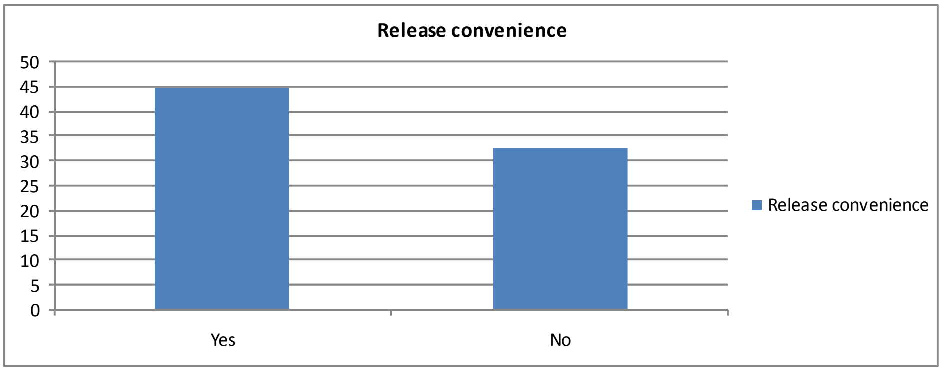


Figure 32: Release convenience.

Question 2: Does your organization utilize a formal release planning process in which specific dates are indicated with regard to the upcoming major, minor, and bugfix releases?
Your answer: No

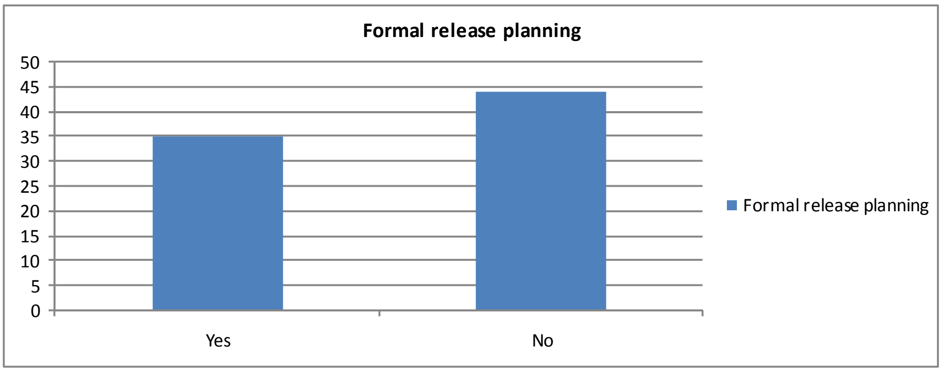


Figure 33: Formal release planning.

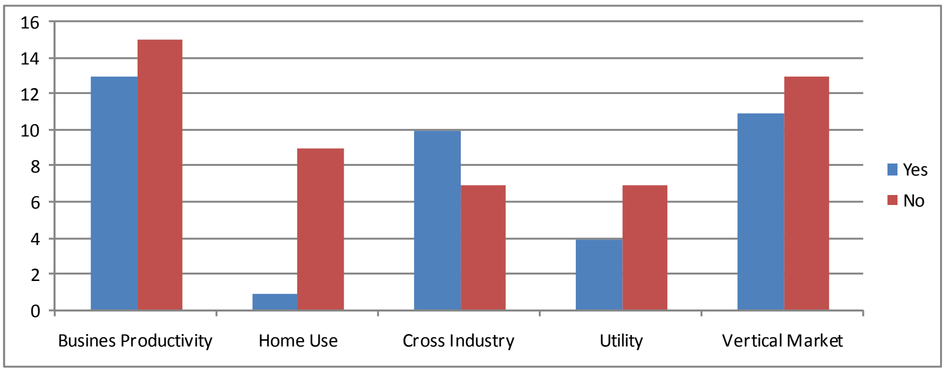


Figure 34: Formal release planning per industry.

[Conditional] Question 3: Is the release planning published in such a way that all relevant internal/company product stakeholders can access this planning at all times?
Your answer: Question skipped

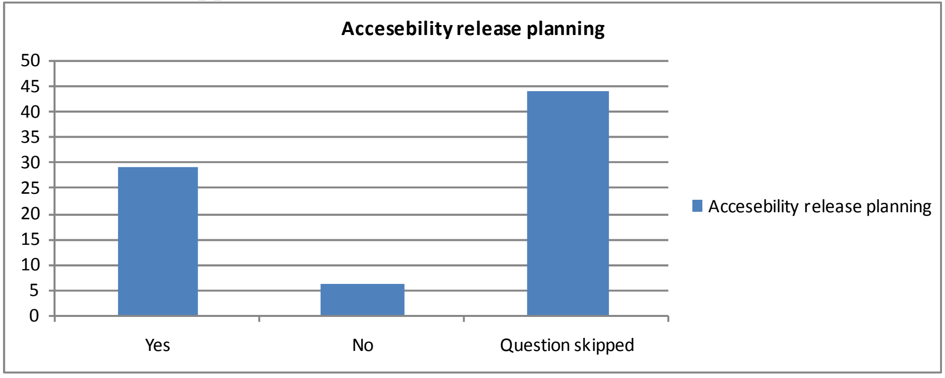


Figure 35: Accesebility release planning.

[Conditional] Question 4: Is there a formal publication policy with regard to this release planning document, which specifies policy decisions important for a specific release?
Your answer: Question skipped

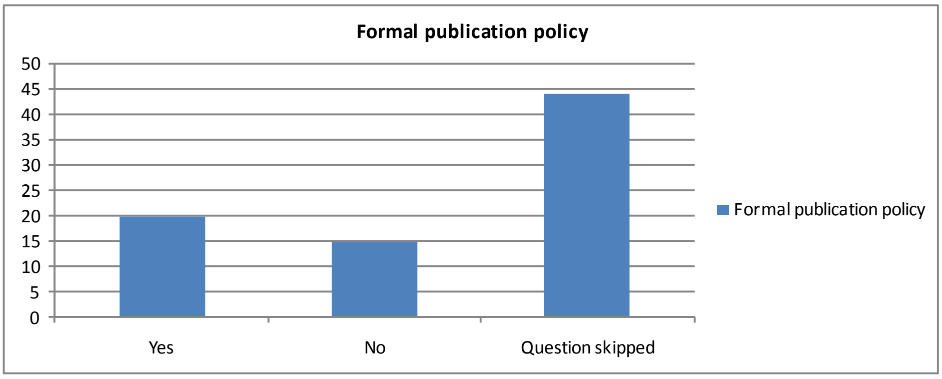


Figure 36: Formal publication policy.

4.6 General Release

Question 1: Is a formalized release scenario present within the organization that describes what happens step by step on release days?
Your answer: No

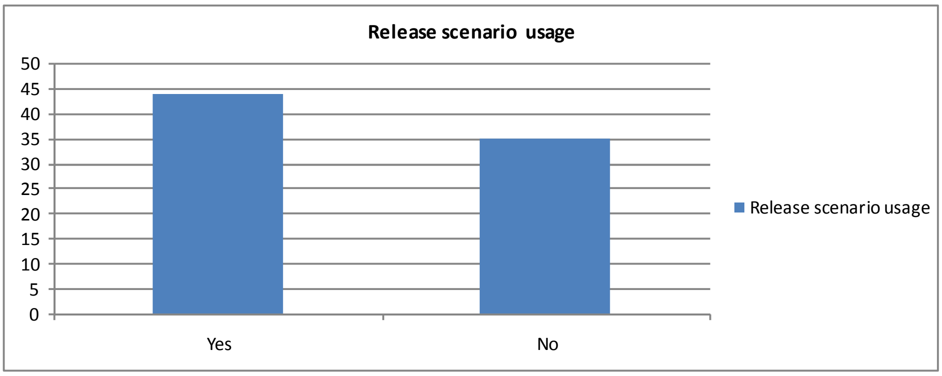


Figure 37: Release scenario usage.

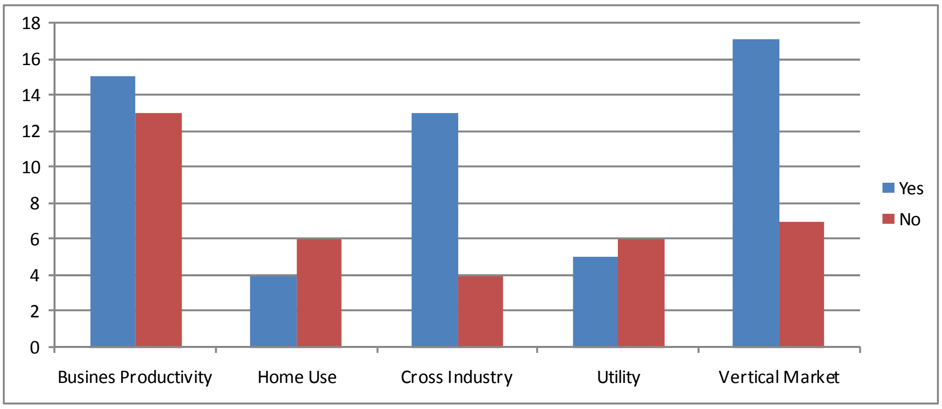


Figure 38: Release scenario usage per industry.

Question 2: Releases are saved:
Your answer:

At the customer site	No
On CDs/DVDs	No
On a network drive	Yes
In a configuration management system like CVS or SubVersion	Yes
Other	Yes

Table 24: Release repository.

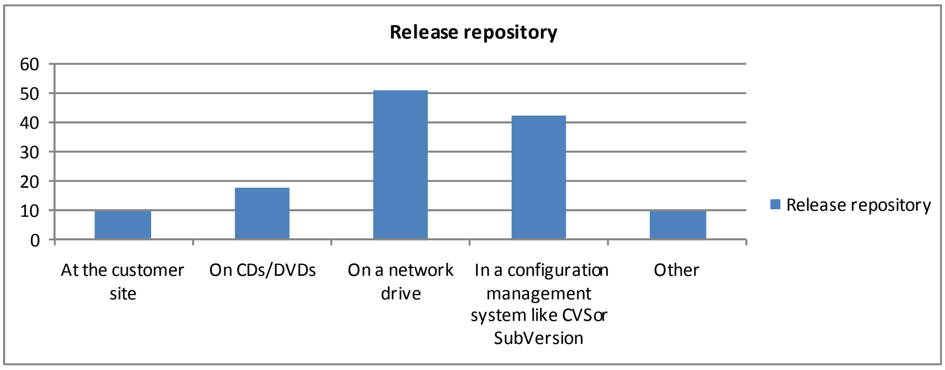


Figure 39: Release repository.

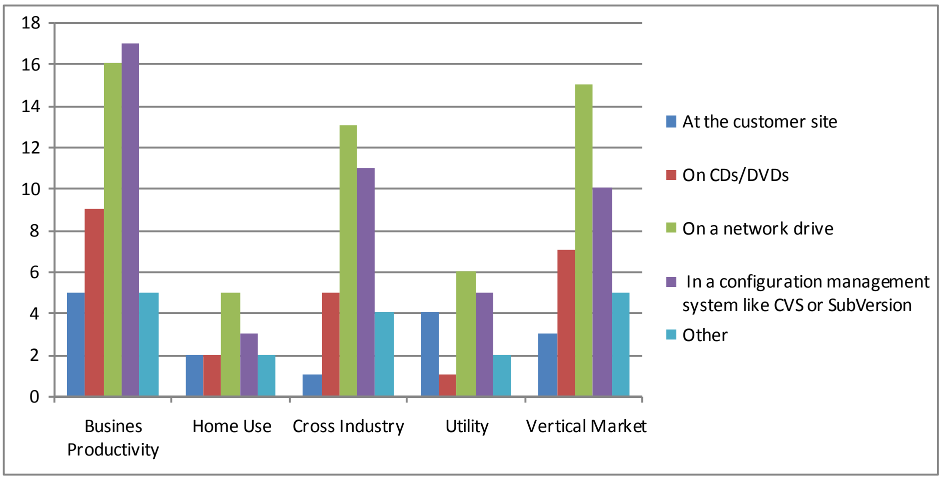


Figure 40: Release repository per industry.

Question 3: All major, minor and bug fix releases can be accessed and used by:
Your answer:

All employees	Yes
All development employees	Yes
All release employees	Yes
All customers	Yes
All partners	Yes
All sales employees	Yes
Other	No

Table 25: Release access.

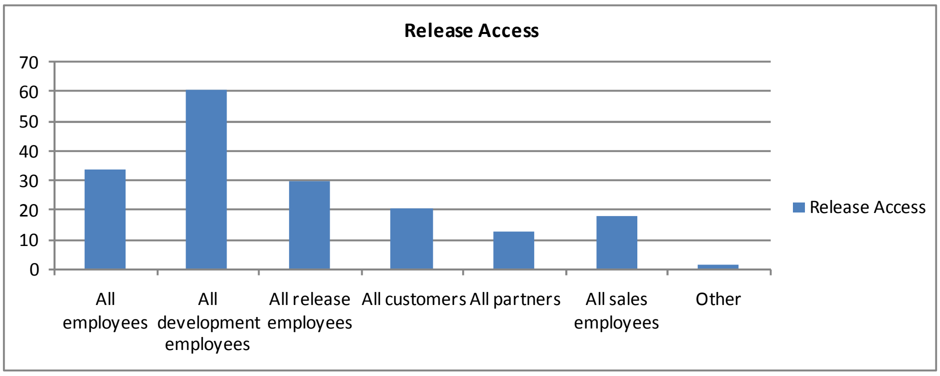


Figure 41: Release access.

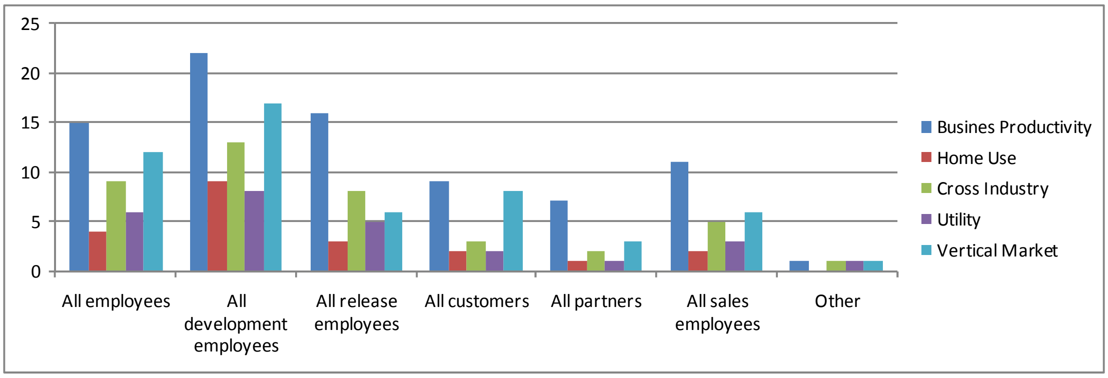


Figure 42: Release access per industry.

Question 4: Can all releases be downloaded by all stakeholders?
Your answer: Yes

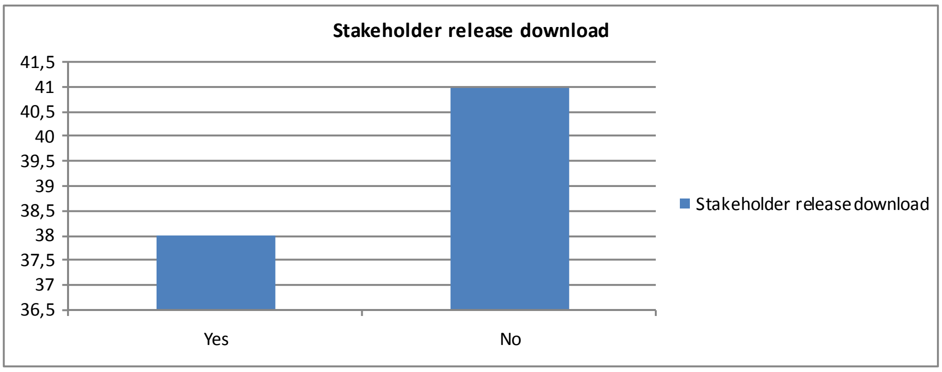


Figure 43: Stakeholder release download.

Question 5: All custom−built tools that are used by the organization to support the CCU process, like installation programs and scripts, are managed explicitly.
Your answer: Yes

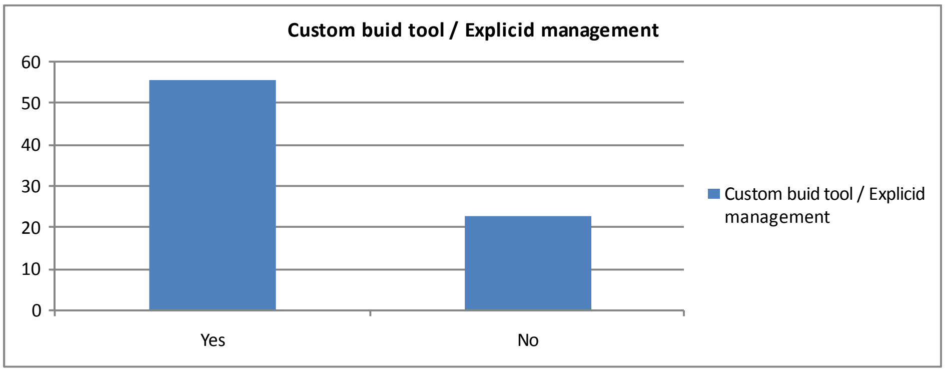


Figure 44: Custom tool / explicid managment.

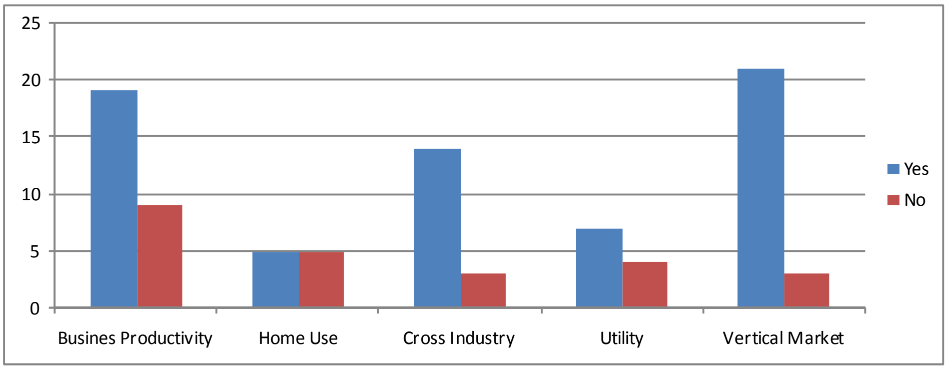


Figure 45: Custom tool / explicid managment per industry.

Question 6: All commercial and open−source tools that are being used for development and support of the CCU process are managed explicitly.
Your answer: Yes

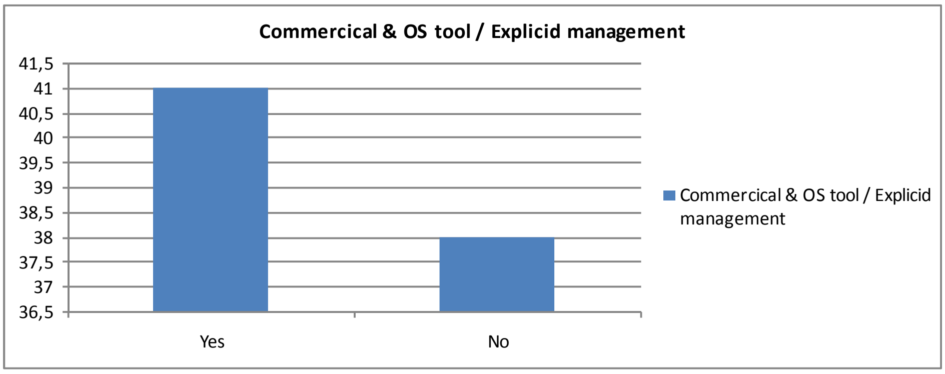


Figure 46: Commercical & OS tool / Explicit management.

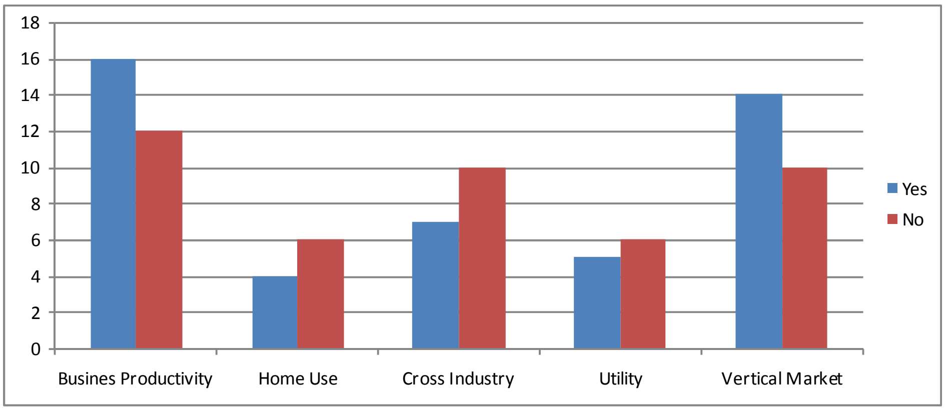


Figure 47: Commercical & OS tool / Explicit management per industry.

Question 7: Are external dependencies, between your product and external components, managed explicitly (in a computer readable format, e.g., "Our Product requires MySQL")?
Your answer: Yes

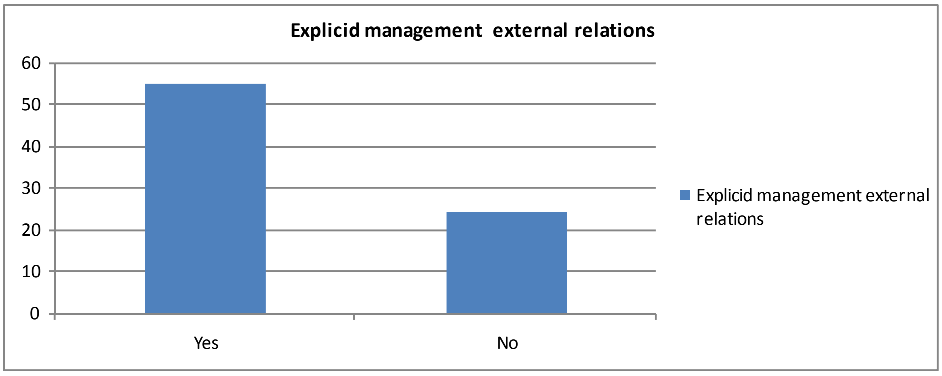


Figure 48: Explicit managment external relations.

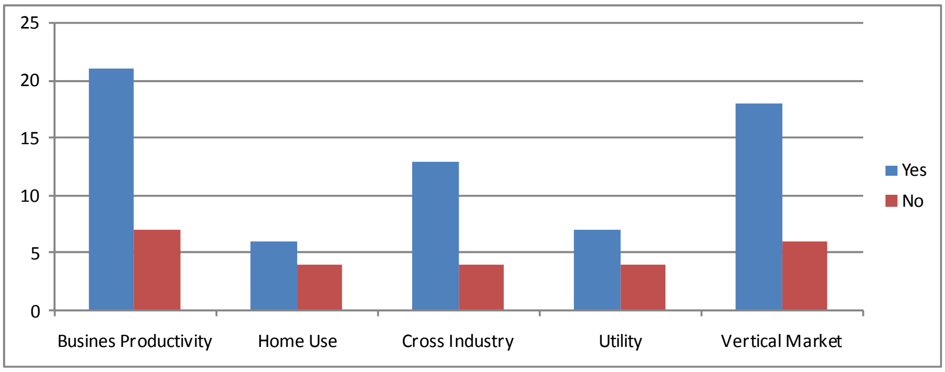


Figure 49: Explicit managment external relations per industry.

Question 8: Does your product contain off–the–shelf components that are delivered with the product?
Your answer: No

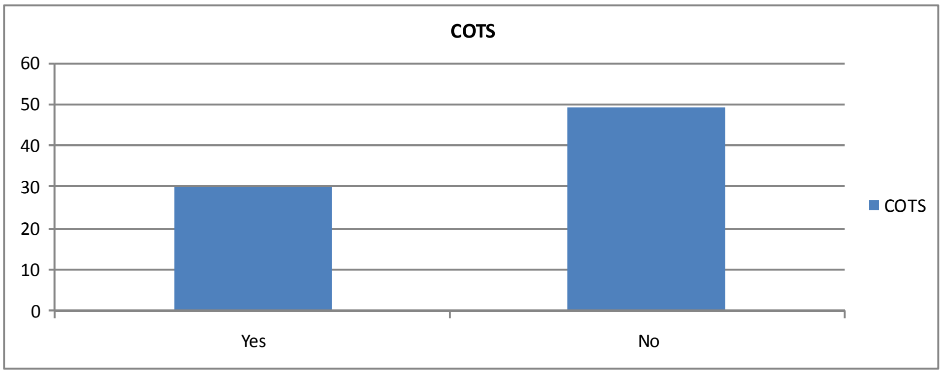


Figure 50: Use of Components of The Shelf.

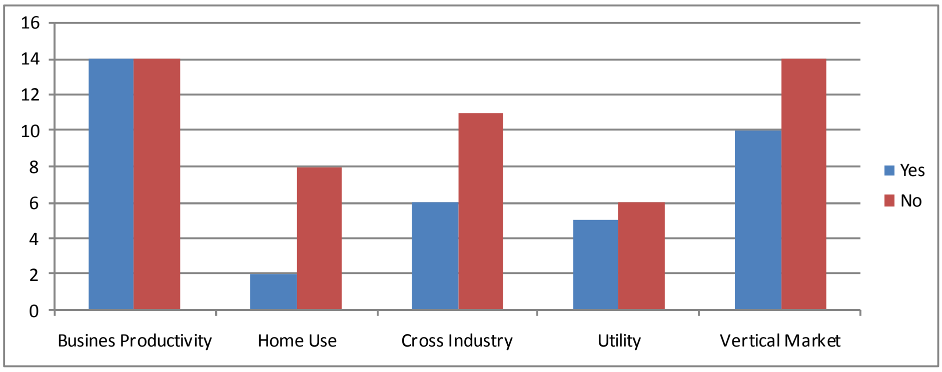


Figure 51: Use of Components of The Shelf per industry.

[Conditional] Question 9: Are these components saved in a repository (at the company side), such that version compatibilty is preserved?
Your answer: No Answer

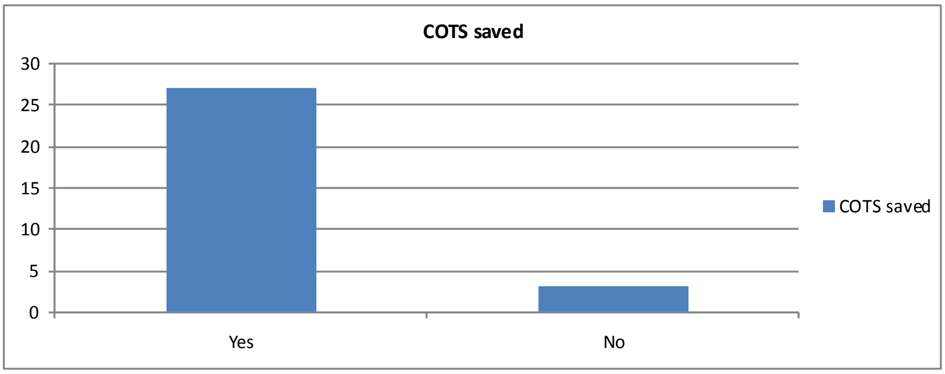


Figure 52: Components of the shelve saved in repository.

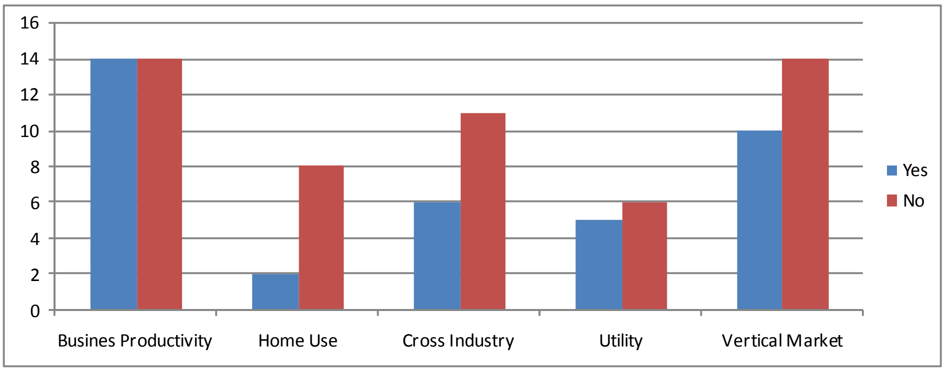


Figure 53: Components of the shelve saved in repository per industry.

4.7 Customer & Delivery

Question 1: You inform your customers through
Your answer:

Domain-specific channels (conferences for instance)	No
The product itself (pop-ups)	No
Paper newsletter	No
A website	Yes
Individual E-Mail	Yes
Phone	No
We push our updates automatically to the customer	No
Online newsletter	No
General announcement list	No
Advertisements in magazines	No
Other	No

Table 26: Customer inform channel.

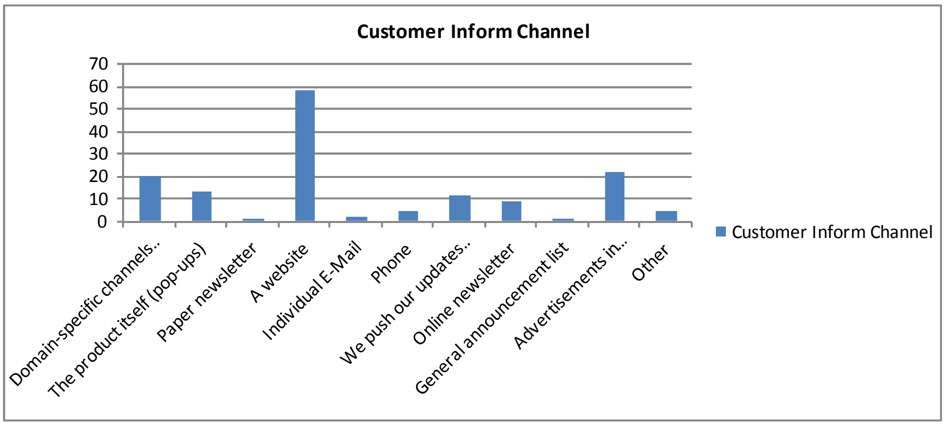


Figure 54: Customer inform channel.

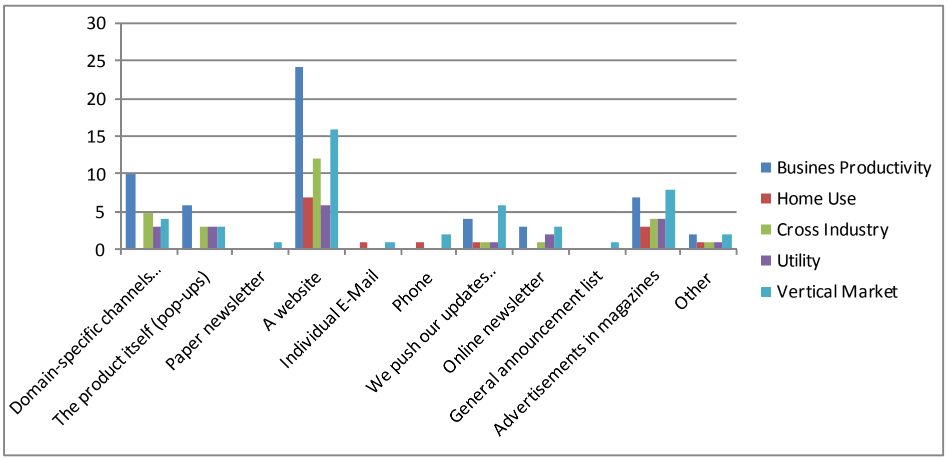


Figure 55: Customer inform channel per industry.

Question 2: We inform our customers, about the product (e.g.; new features, updates, tips, help) at least:
Your answer: Once per three months

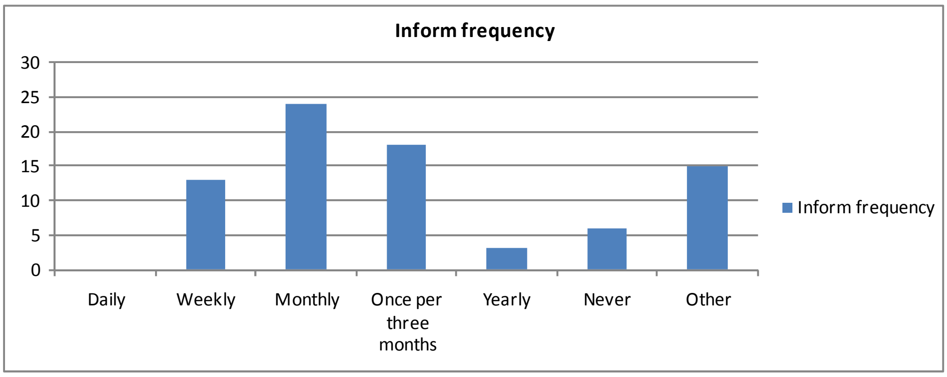


Figure 56: Inform frequency.

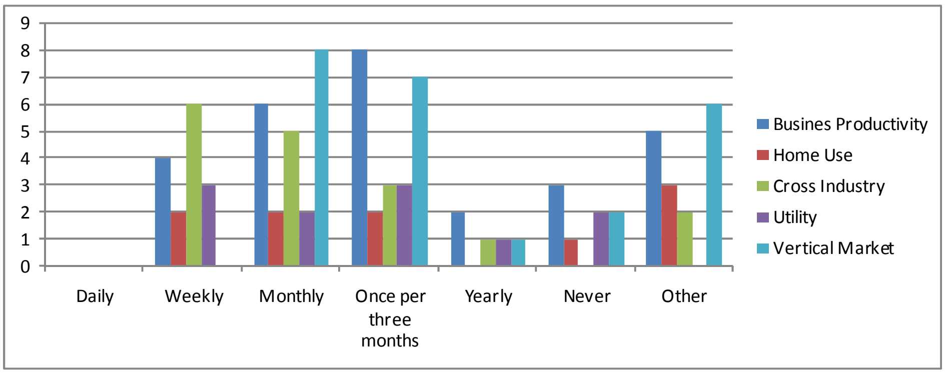


Figure 57: Inform frequency per industry.

Question 3: Customers report bugs through
Your answer:

a online bug system	Yes
E-mail	Yes
Phone	Yes
Fax	No
The product sends automatic error reports	No
Other	No

Table 27: Bug reporting channels.

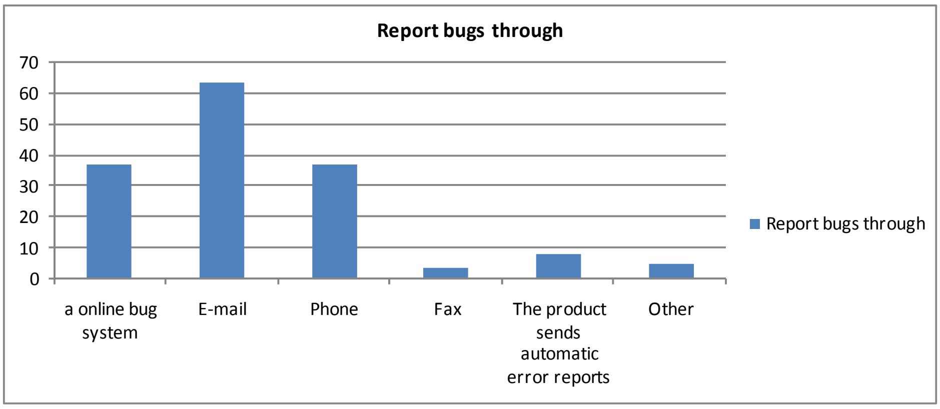


Figure 58: Bug reporting channels.

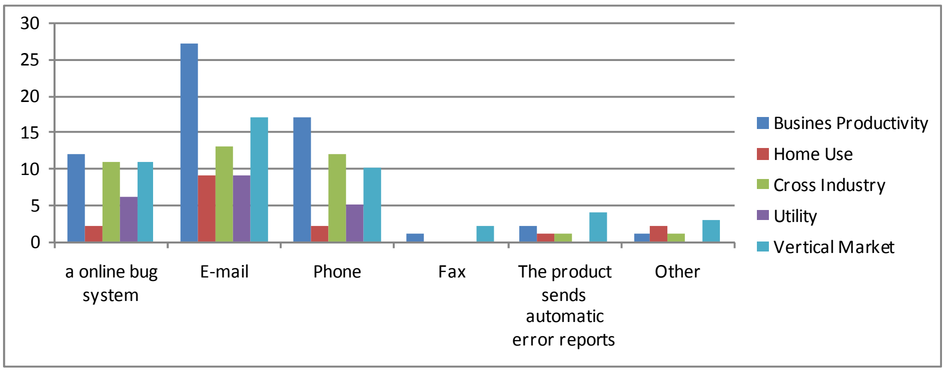


Figure 59: Bug reporting channels per industry.

Question 4: In which package format(s) is your product delivered (select all that apply)?
Your answer:

RPM	No
Exe (wise install)	No
Exe (Installshield)	No
Exe (Powerupdate)	No
Portage	No
MSI	Yes
MSI WIX	No
Zip/Rar archive	No
APT-GET	No
Our corporate installer format	No
Source bundle	No
We don't use an explicit release format	No
Other	No

Table 28: Package format(s).

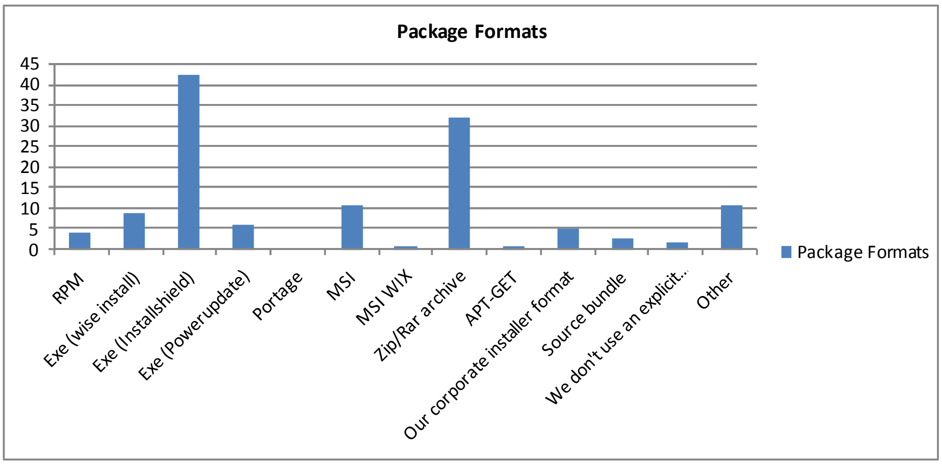


Figure 60: Package format(s).

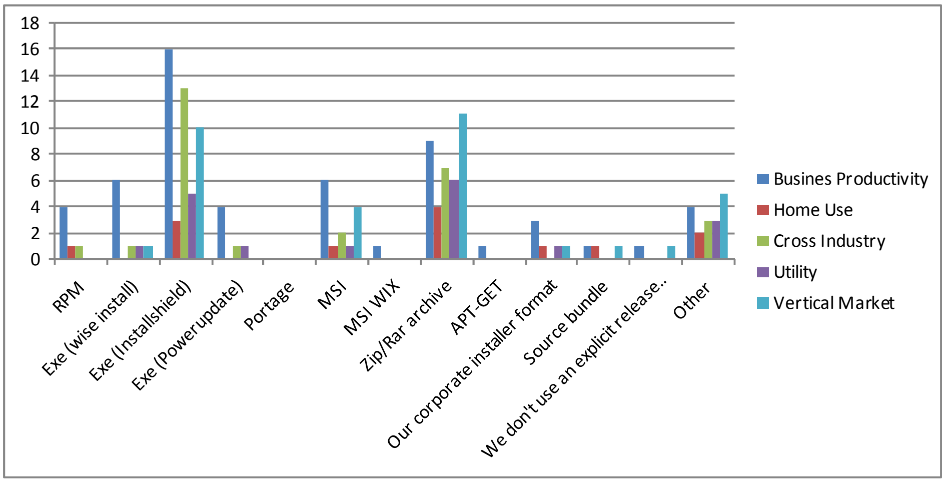


Figure 61: Package format(s) per industry.

Question 5: How long does it take on average for a customer to install a release after its release date?
Your answer:

Major	1 day
Minor	1 day
Bugfix	1 day

Table 29: Release installation lagg.

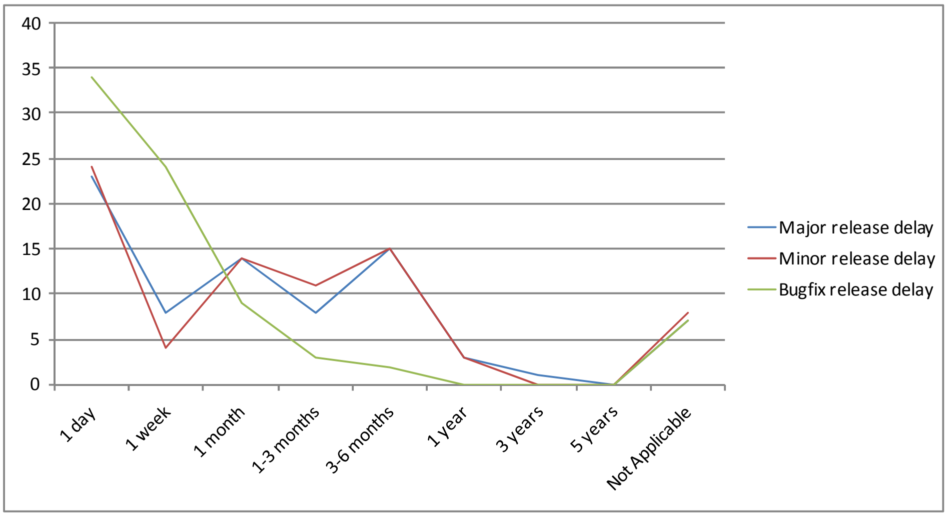


Figure 62: Release installation lagg.

Question 6: Are you able to remotely deploy your product at the customer platform?
Your answer: No

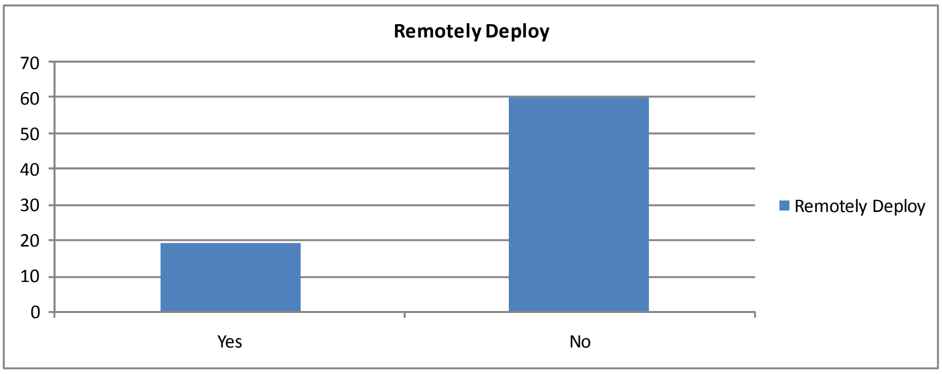


Figure 63: Remote deployment.

4.8 Delivery Methods

Question 1: Your product can be delivered as follows:
Your answer:

Floppy	No
CD-ROM	No
DVD	No
E-mail	No
Our website	Yes
Secure phoneline or internet connection	No
USB stick	No
Our product is web based which we can update ourself and does not otherwise need to be delivered	No
FTP	Yes
Other	No

Table 30: Delivery method.

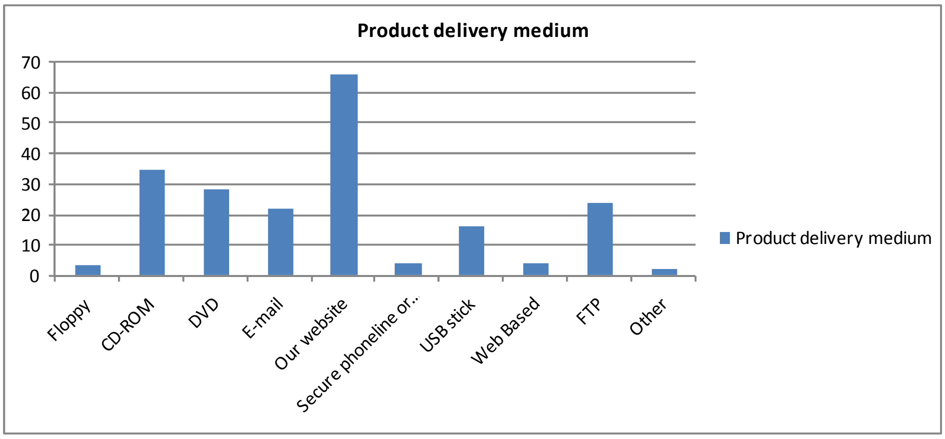


Figure 64: Delivery method.

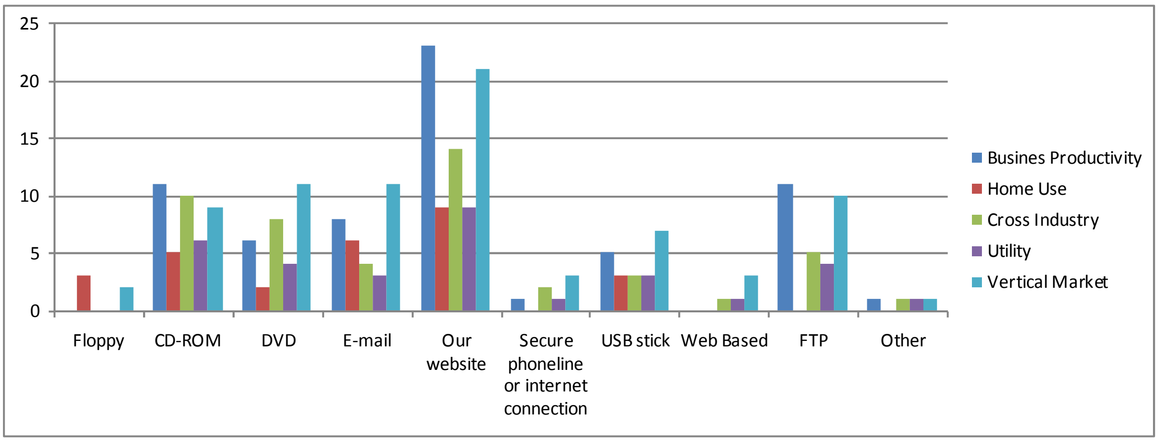


Figure 65: Delivery method per industry.

Question 2: Your product can be pushed/pulled as follows
Your answer:

Manual pull	Yes
Automatic pull	No
Manual push	No
Automatic push	No
Our product is web based	No
Other	No

Table 31: Product delivery construct.

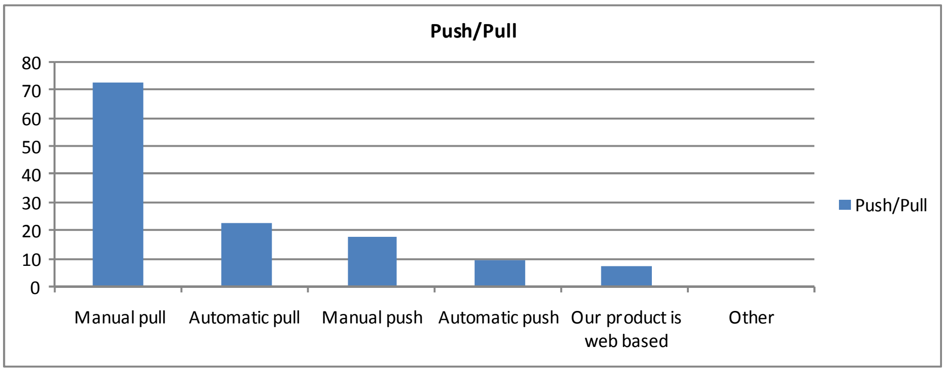


Figure 66: Product delivery construct.

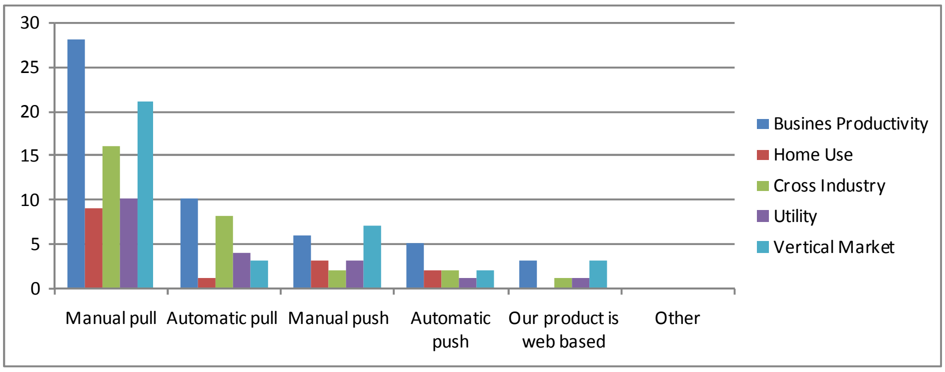


Figure 67: Product delivery construct per industry.

Question 3: Your product update tool makes it possible to download the product from every location, not only from your release repository.
Your answer: Yes

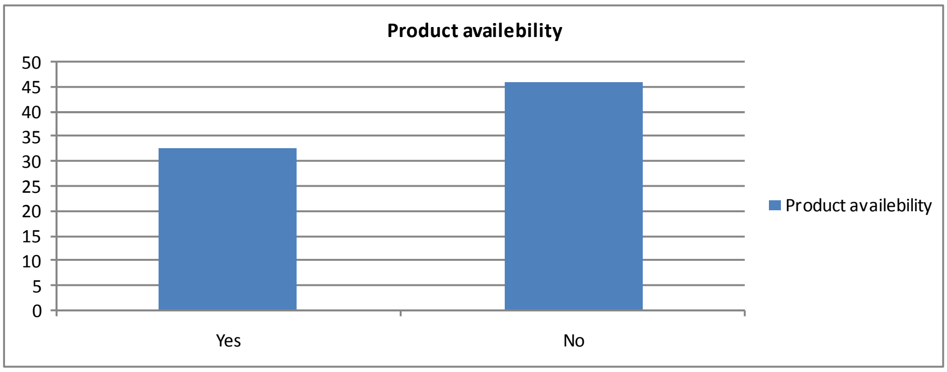


Figure 68: Product availability.

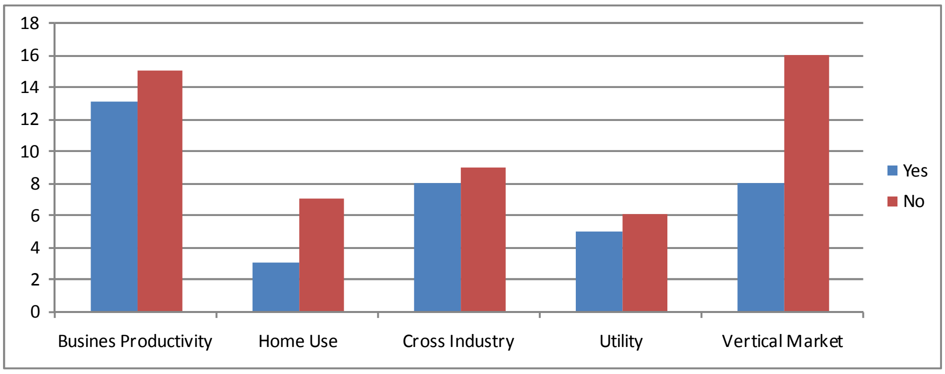


Figure 69: Product availability per industry.

4.9 Installation Methods

Question 1: Is it possible to de−install the previous release of your product without executing complicated manual operations?
Your answer: Yes

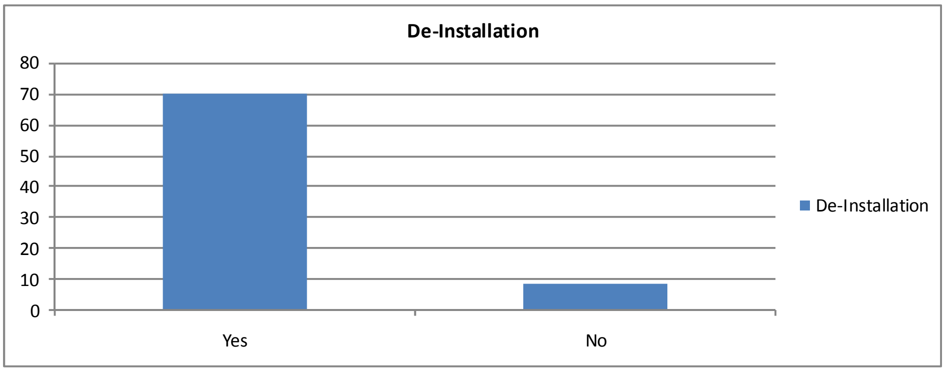


Figure 70: Release de-installment.

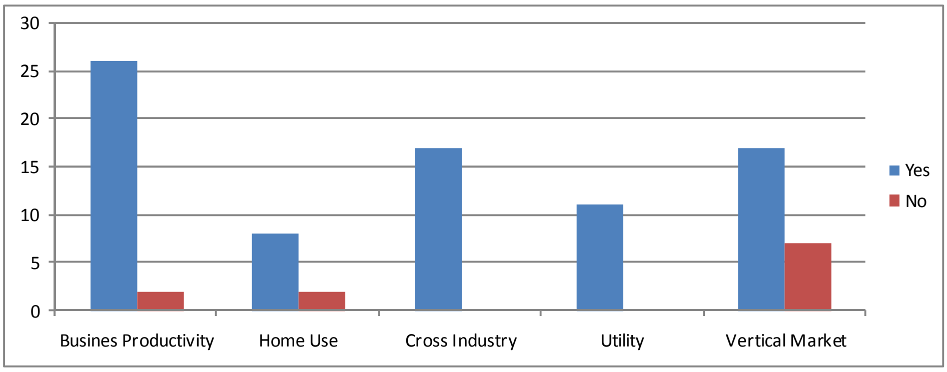


Figure 71: Release de-installment per industry.

Question 2: Are external relationships between your and other products managed?
Your answer: Yes

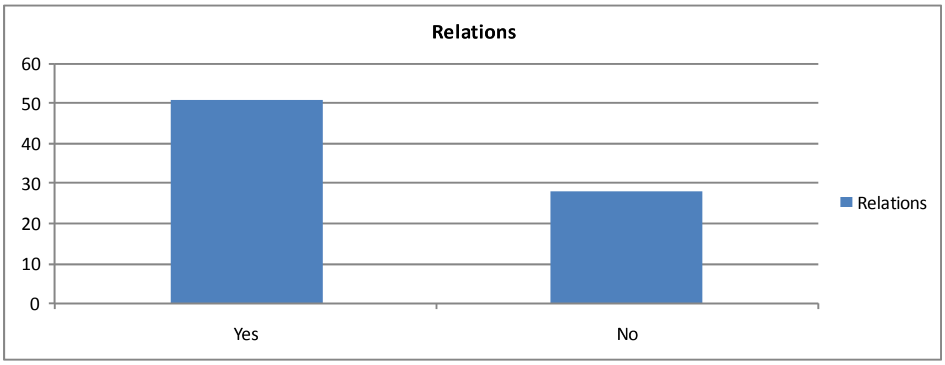


Figure 72: Relations.

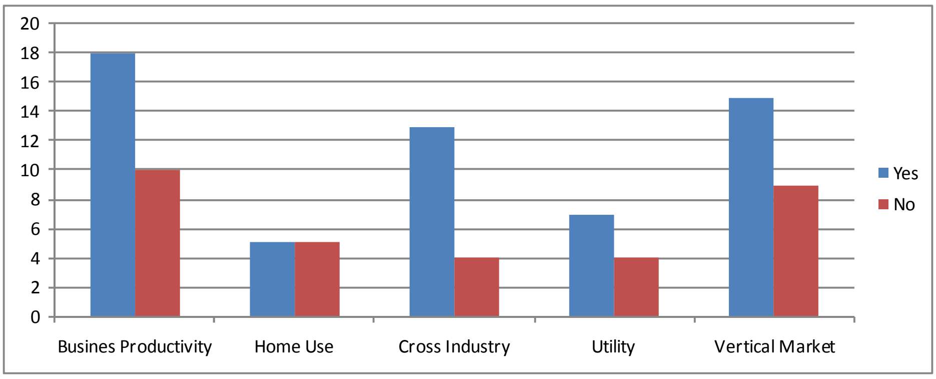


Figure 73: Relations per industry.

Question 3: Does your product check the local customer configuration before installation (such as the needed amount of disk space)?
Your answer: Yes

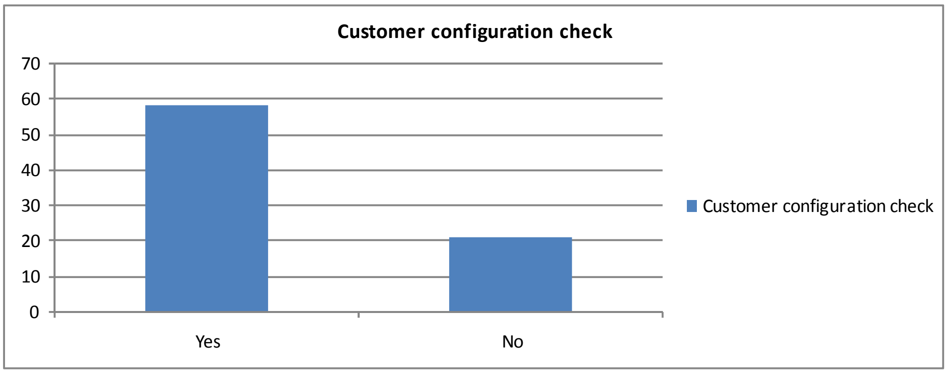


Figure 74: Customer configuration check.

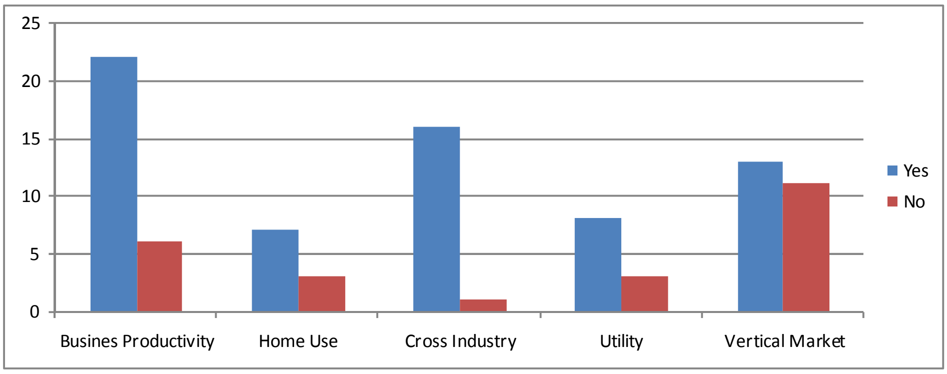


Figure 75: Customer configuration check per industry.

Question 4: Is it possible to undo an update?
Your answer: Yes

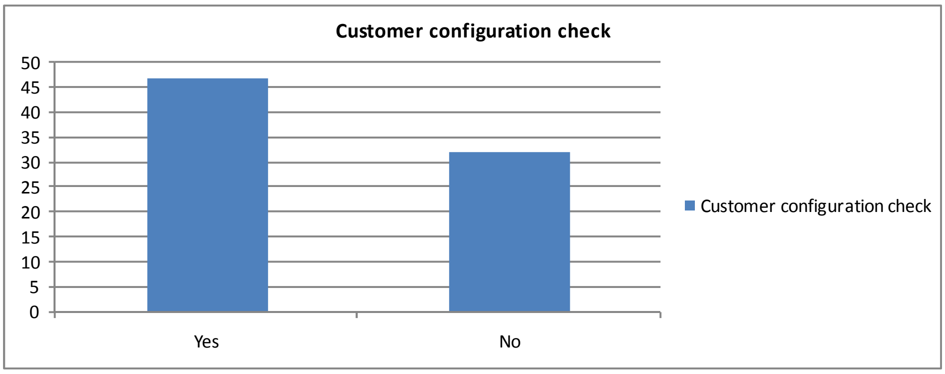


Figure 76: Undo update.

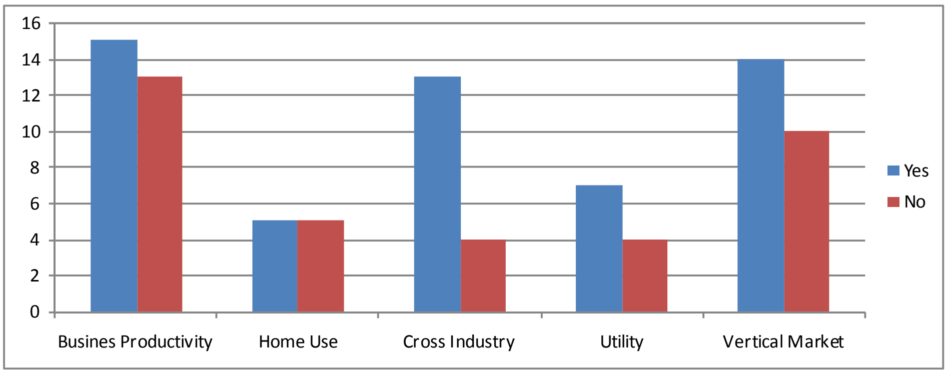


Figure 77: Undo update per industry.

Question 5: Does your product require an update tool, which updates the product on the customer−side?
Your answer: No

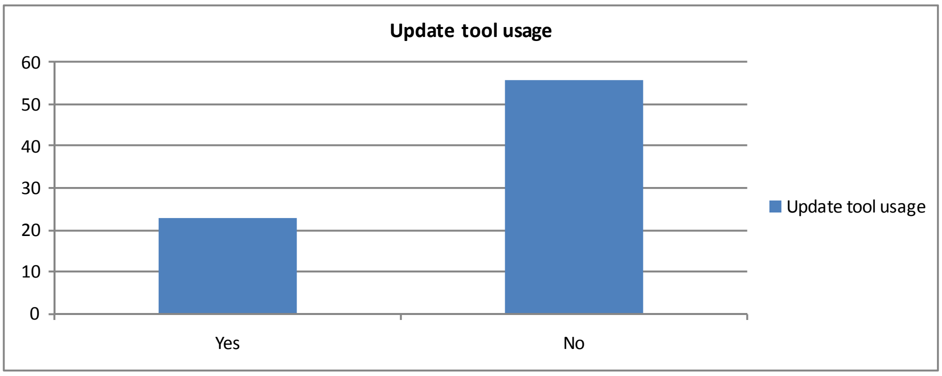


Figure 78: Update tool requirement.

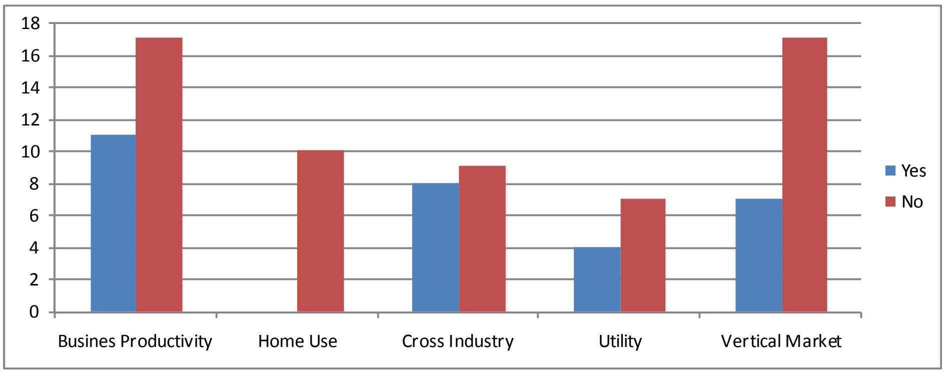


Figure 79: Update tool requirement per industry.

[Conditional] Question 6: When problems occur during installation, which of these are addressed automatically?

Your answer:

Harddisk space	No
Third-party components	No
Availability of (your own) other components	No
Data from earlier product installations	No
Operating systems	No
Hardware	No
There is no check	No
Other	No

Table 32: Automattically addressed problems.

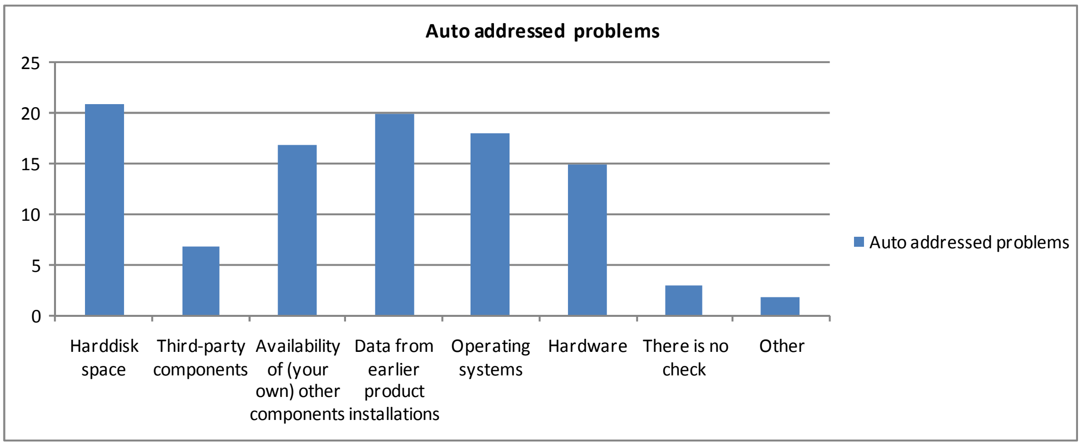


Figure 80: Automattically addressed problems.

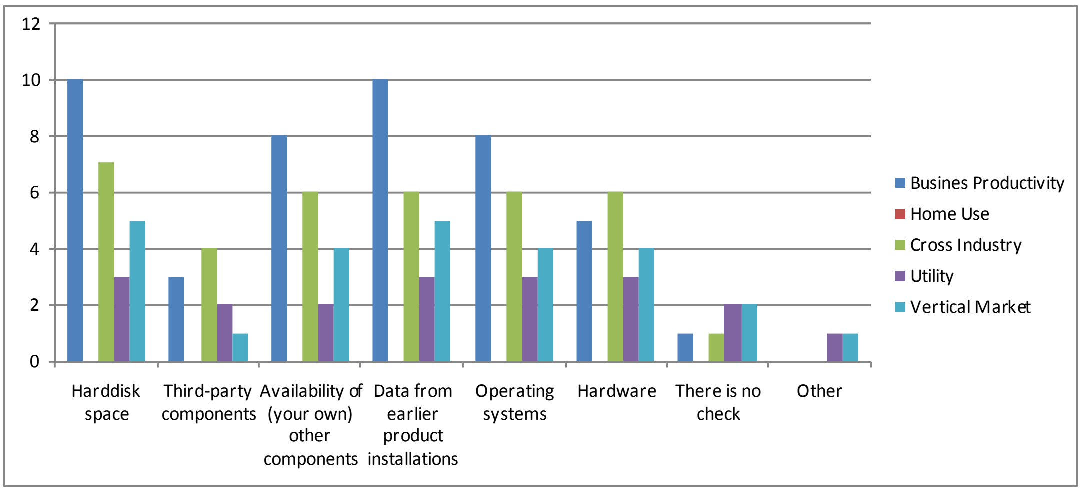


Figure 81: Automattically addressed problems per industry.

[Conditional] Question 7: Is the update tool able to update at runtime?
Your answer: No Answer

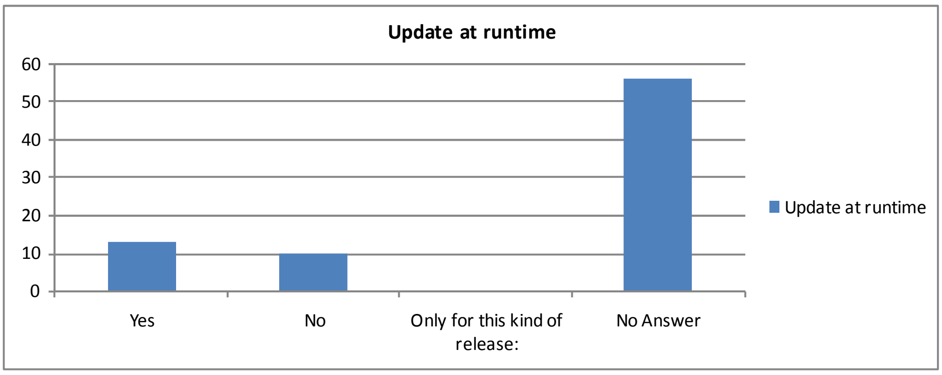


Figure 82: Updating at runtime.

[Conditional] Question 8: Is the update tool still able to deploy the product if the customer implements customizations, extensions and/or customer–specific solutions?
Your answer: No Answer

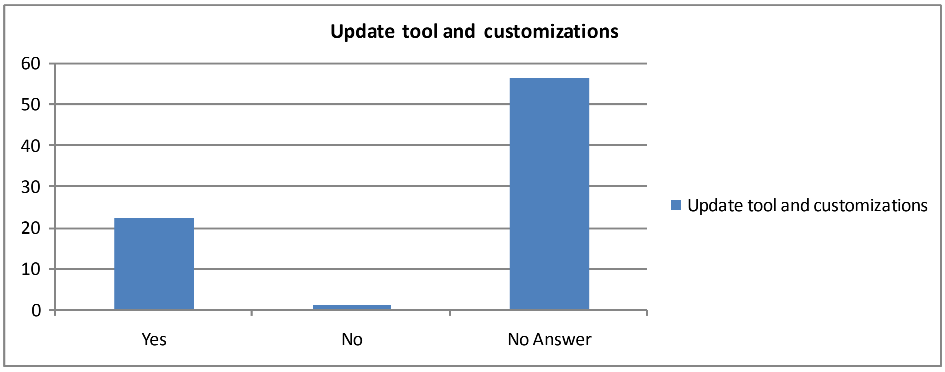


Figure 83: Deployment after customizations.

Question 9: Can your product verify, at any point in time after its been successfully installed, if its critical dependencies are still properly installed and configured?
Your answer: No

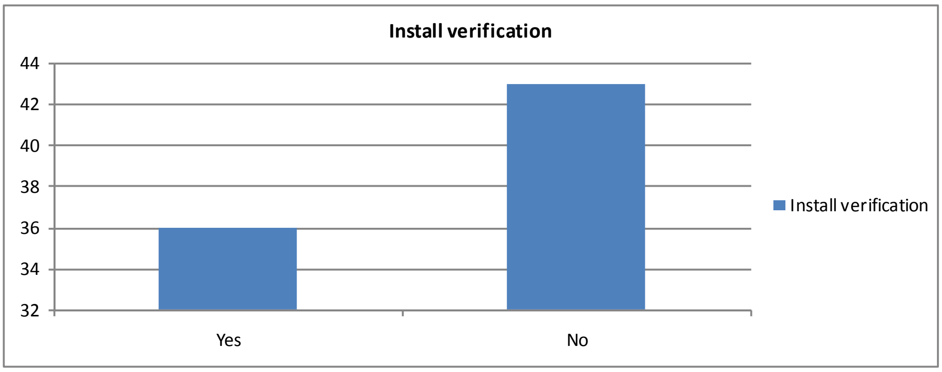


Figure 84: Verification of critical dependencies.

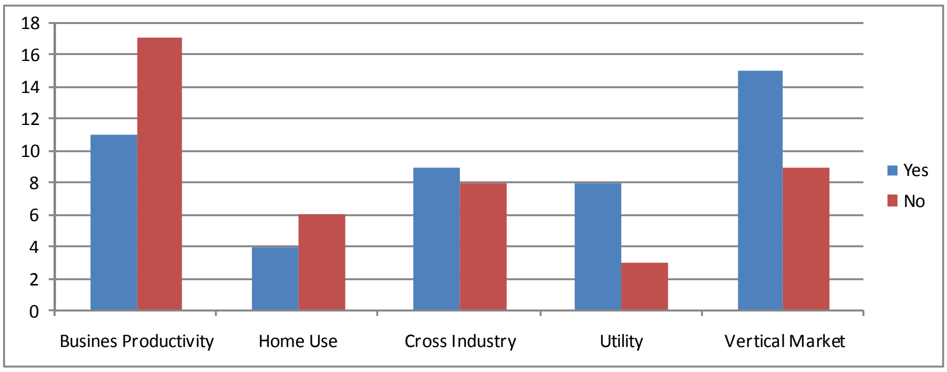


Figure 85: Verification of critical dependencies per industry.

Question 10: Is all the data produced by the user (like configuration settings and documents) stored on a different location from the product (so to make it possible to back up the user data without backing up the product)?
Your answer: Yes

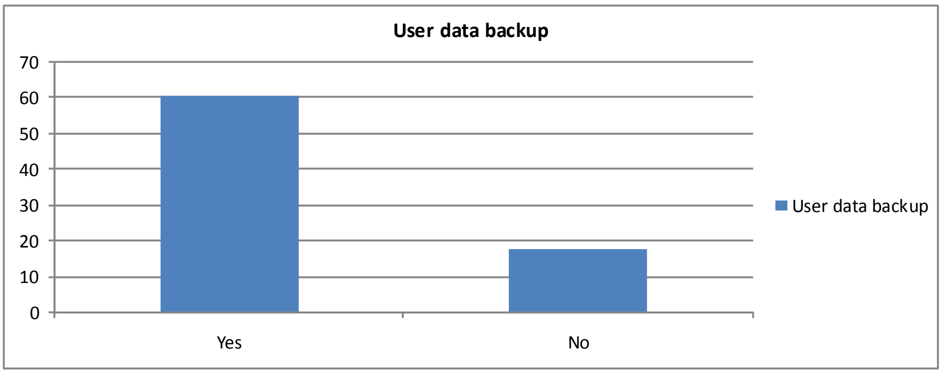


Figure 86: User data seperate from product data.

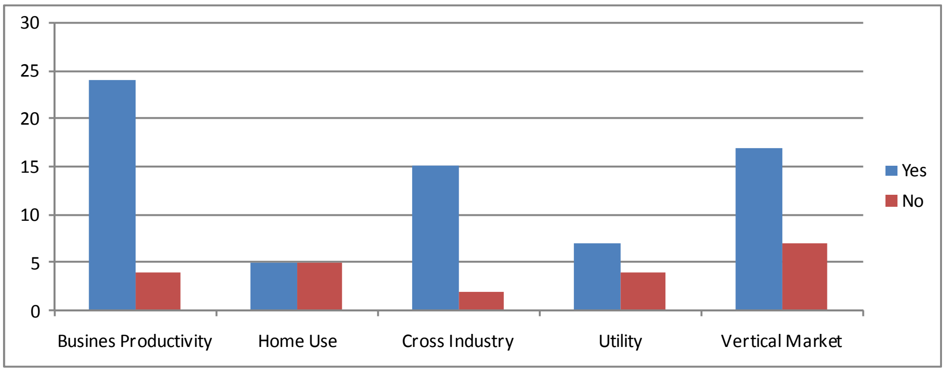


Figure 87: User data seperate from product data per industry.

Question 11: Is it possible to install the product in a DTAP environment, where new versions progress from an unstable to a stable production environment (i.e. a test environment to the real *production environment)?*
Your answer: Yes

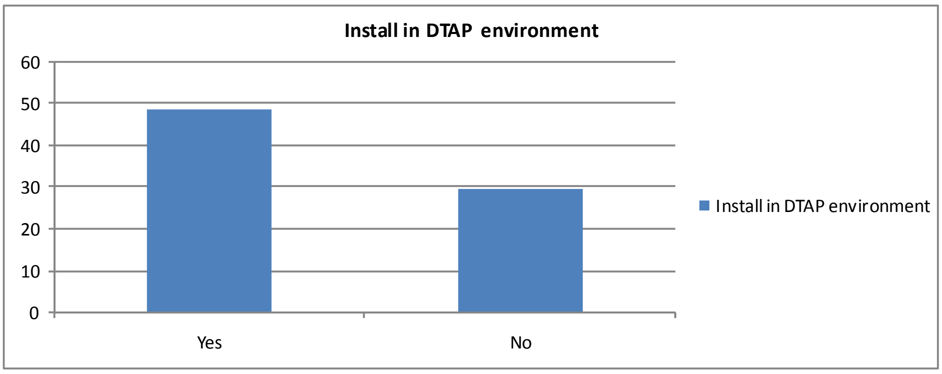


Figure 88: Deployment in DTAP environment.

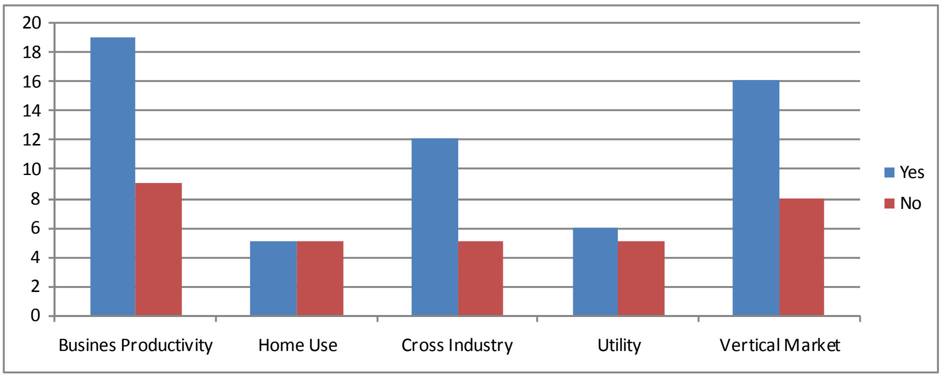


Figure 89: Deployment in DTAP environment per industry.

Question 12: Please indicate the rate of deployment failure, for your product, at the first installation attempt, in %.
Your answer: 2

4.10 Licenses

Question 1: Which kinds of purchase/payment methods do you use for your customers?
Your answer:

Pay per usage	No
Pay per user(name)	No
Pay per time unit	No
Pay per floating user	No
Pay for services, no purchase cost	Yes
Lump sum	No
No payment	No
Other	No

Table 33: Purchase/payment methods.

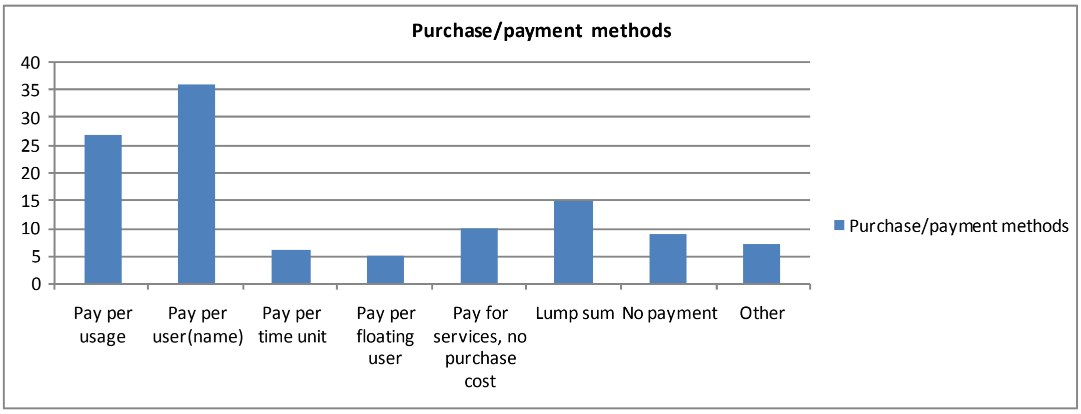


Figure 90: Purchase/payment methods.

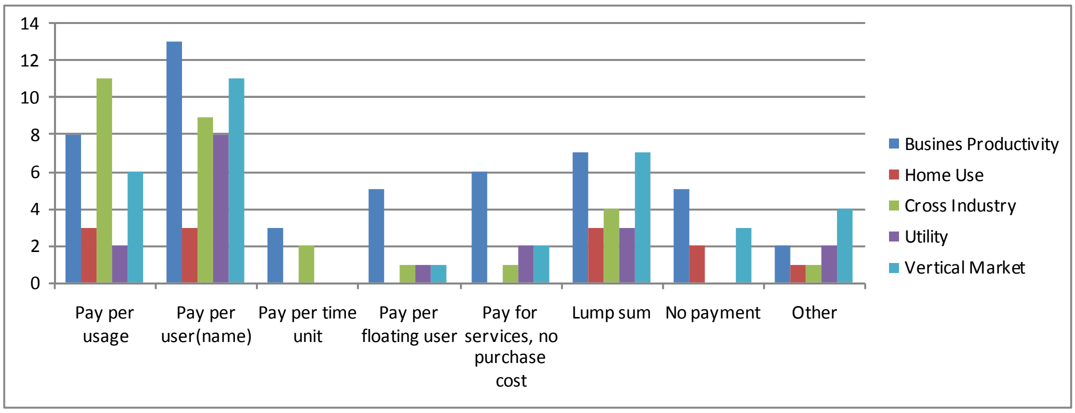


Figure 91: Purchase/payment methods per industry.

Question 2: Do you use a type of license agreement with your customers?
Your answer: Yes

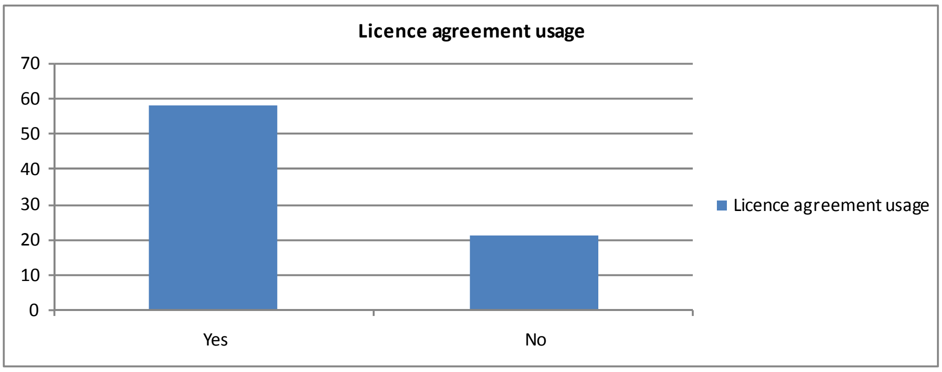


Figure 92: License agreement usage.

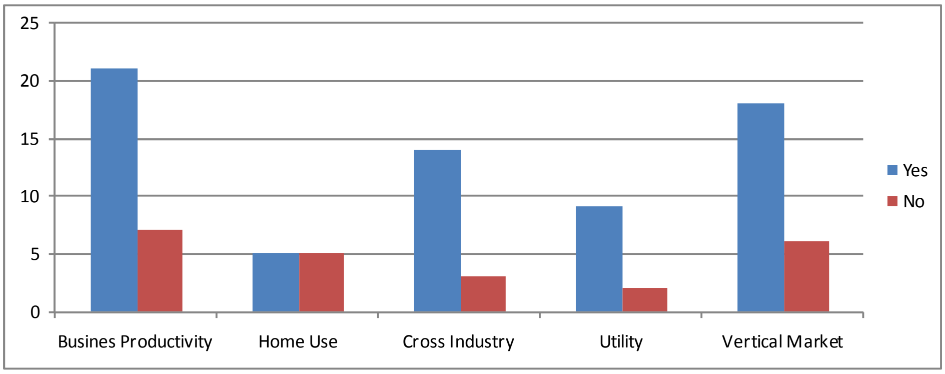


Figure 93: License agreement usage per industry.

[Conditional] Question 3: What data do your license files contain?
Your answer:

The customers name and adress	No
The number of users	No
The purchased modules	No
Other	Yes

Table 34: License file data.

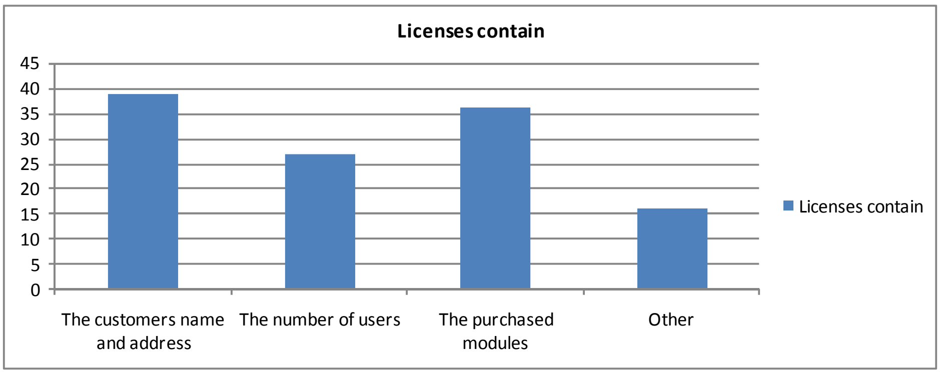


Figure 94: License file data.

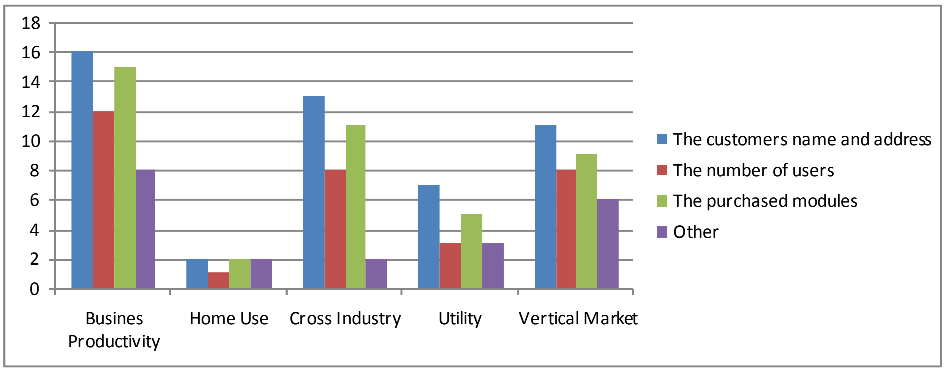


Figure 95: License file data per industry.

[Conditional] Question 4: Is the customer able to renew, extend or expand the license without any actions on your part?
Your answer: Yes

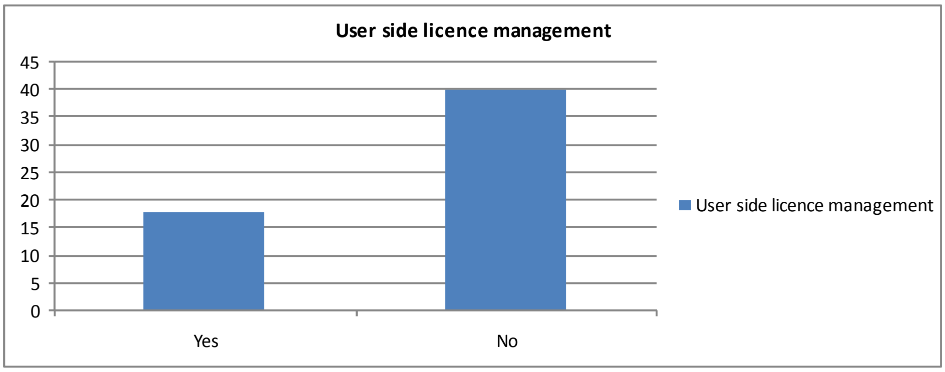


Figure 96: License adaption by user.

[Conditional] Question 5: Do your licenses expire?
Your answer: Yes

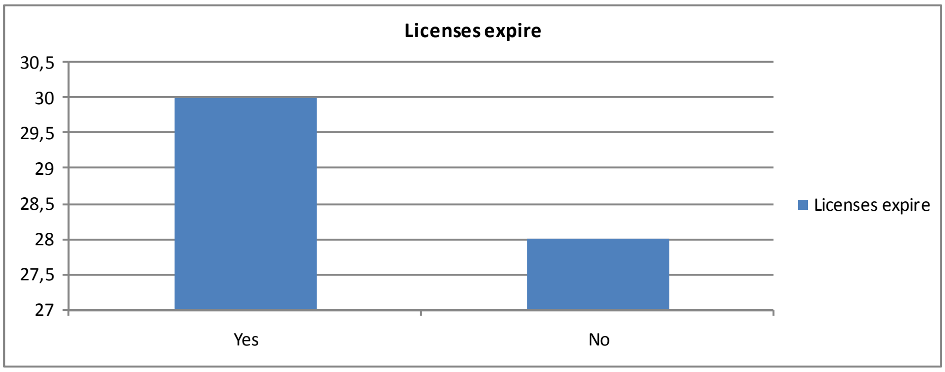


Figure 97: Licenses expire

[Conditional] Question 6: Do you regularly provide temporary licenses?
Your answer: Yes

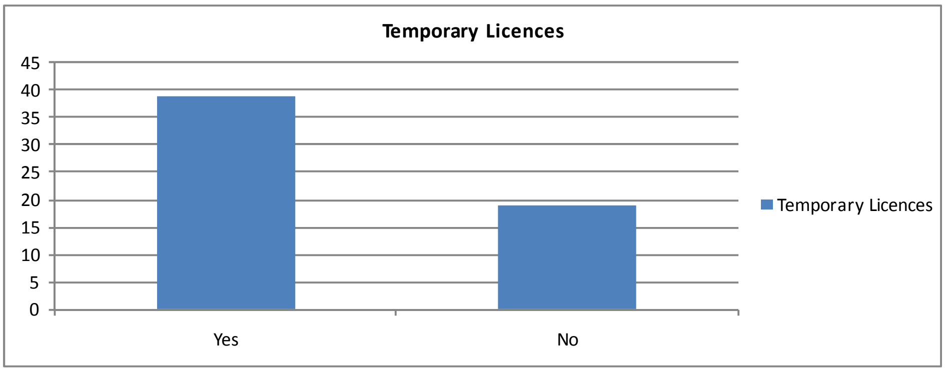


Figure 98: Temporary licenses.

[Conditional] Question 7: Are licenses generated automatically from sales contracts?
Your answer: No

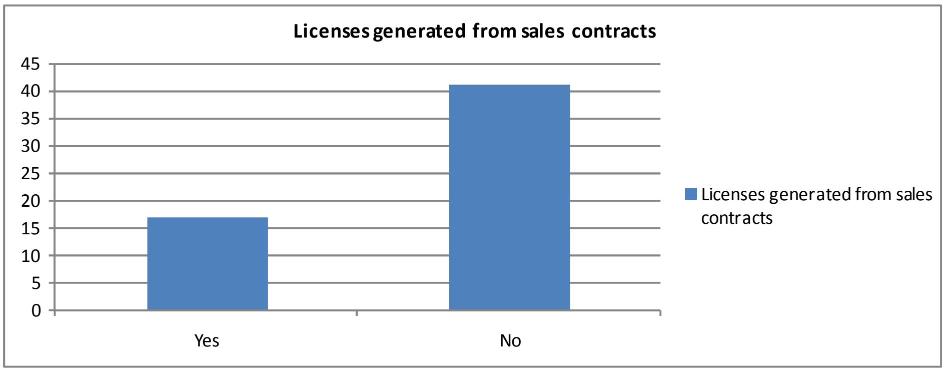


Figure 99: License generation from contracts.

4.11 Tools

Question 1: Compared to the custom−built tools you use now, what custom−built tool would you rather have purchased commercially (if any)?
See next chapter for open questions

Question 2: In the customer configuration updating process, what tools do you believe are missing at this moment?
See next chapter for open questions

Question 3: What tools for the Customer Configuration Updating process would you like to dissuade others from using based on your experiences?
See next chapter for open questions

4.12 User feedback

Question 1: Are you aware of the way in which your customers usually customize the product?
Your answer: Yes

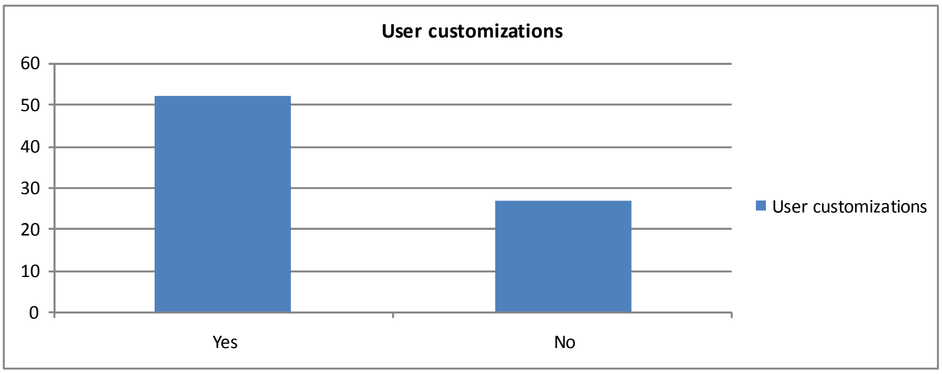


Figure 100: User customizations.

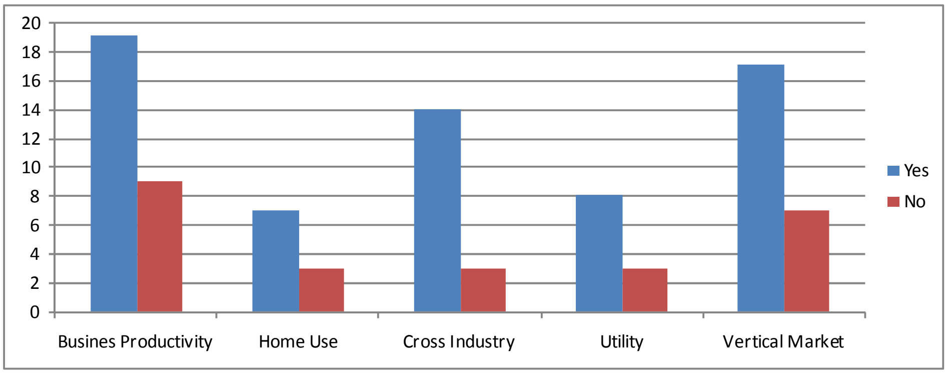


Figure 101: User customizations per industry.

Question 2: Are you aware of the hard– and software platforms that your customers use to operate your product?:
Your answer: Yes

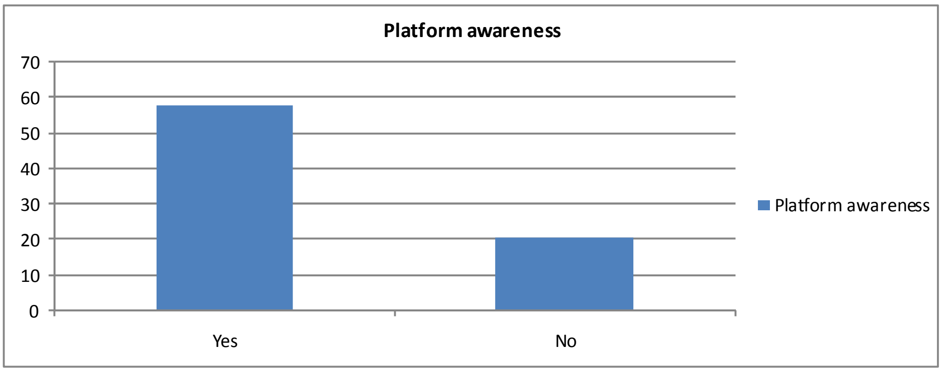


Figure 102: Platform awareness.

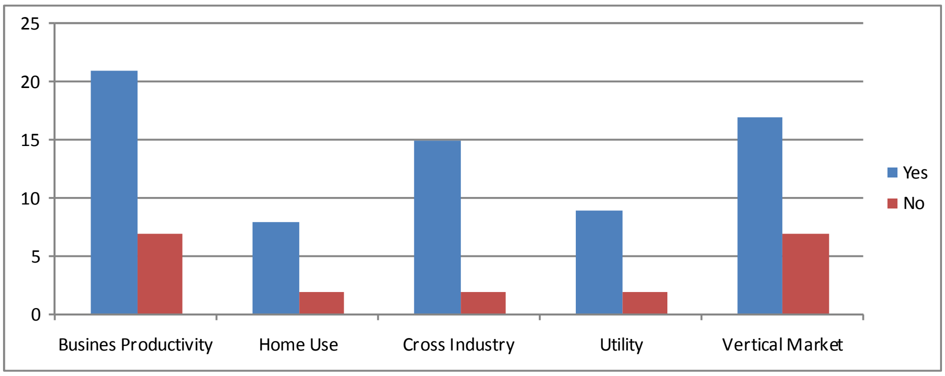


Figure 103: Platform awareness per industry.

Question 3: Does your product send automatic error reports when an error occurs in your product?:
Your answer: No

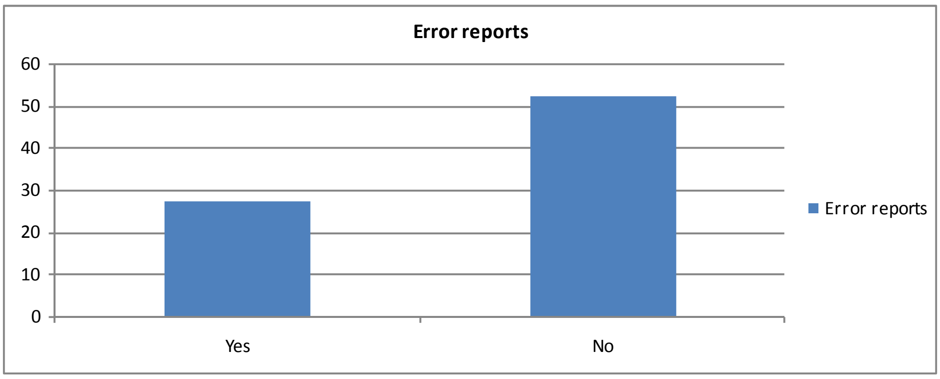


Figure 104: Error reports.

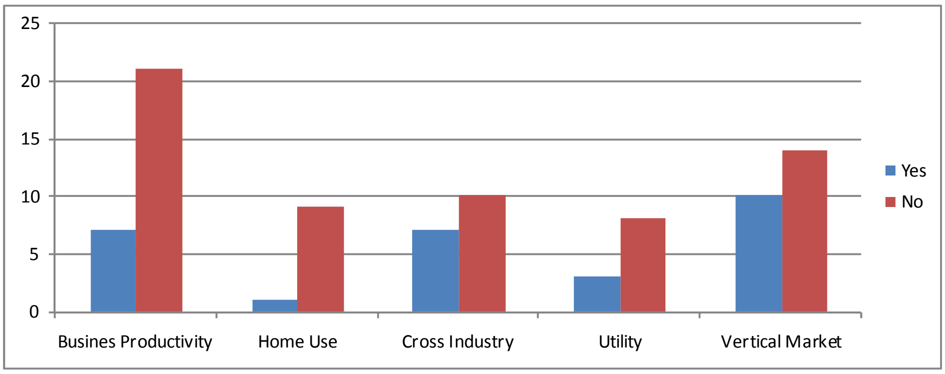


Figure 105: Error reports per industry.

[Conditional] Question 4: Are these automatic error reports analyzed?:
Your answer: No Answer

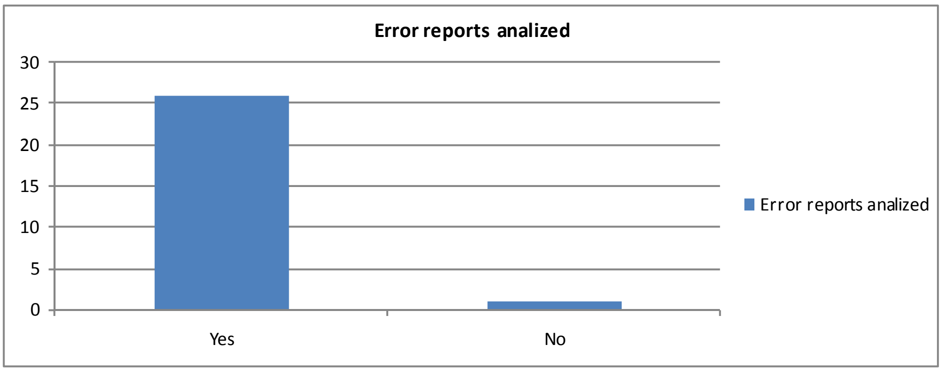


Figure 106: Error reports analized.

Question 5: Does your product generate usage reports?
Your answer: No

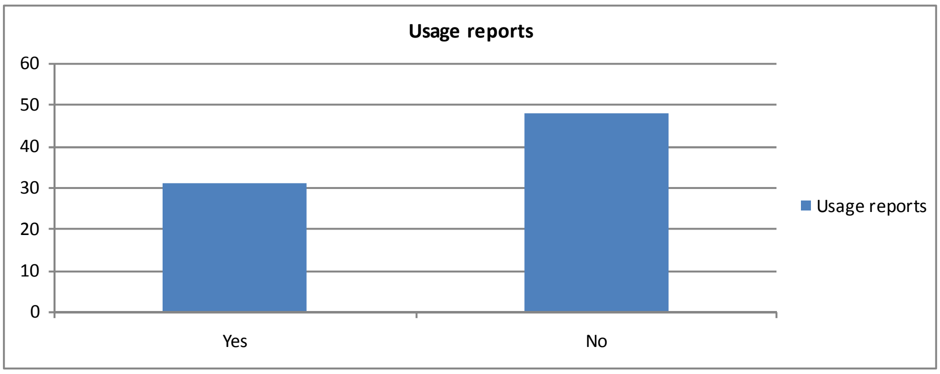


Figure 107: Usage reports.

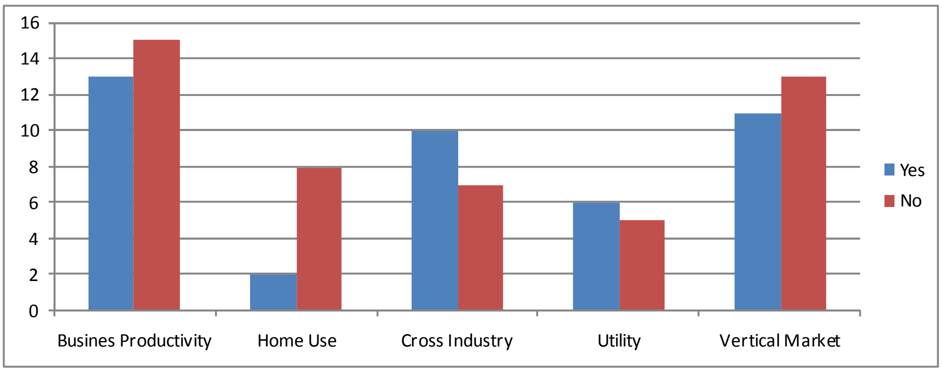


Figure 108: Usage reports per industry.

[Conditional] Question 6: Are these usage reports analyzed?
Your answer: No Answer

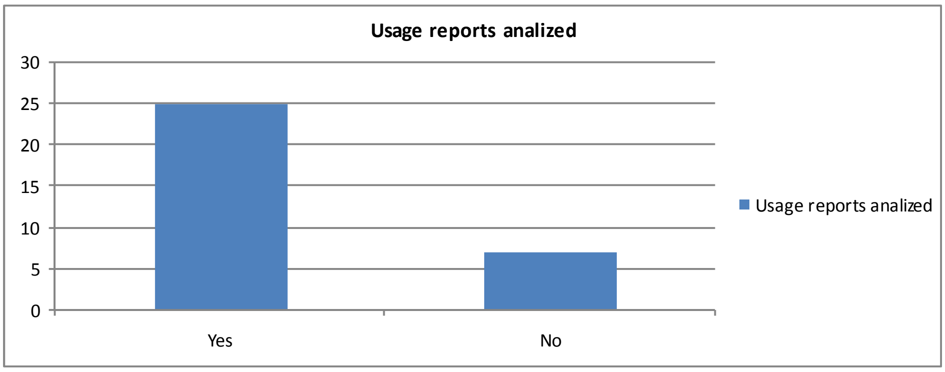


Figure 109: Usage reports analized.

4.13 Maturity & success

Question 1: Please indicate how your Customer Configuration Updating process has evolved over the last two years.
Your answer: Large improvements

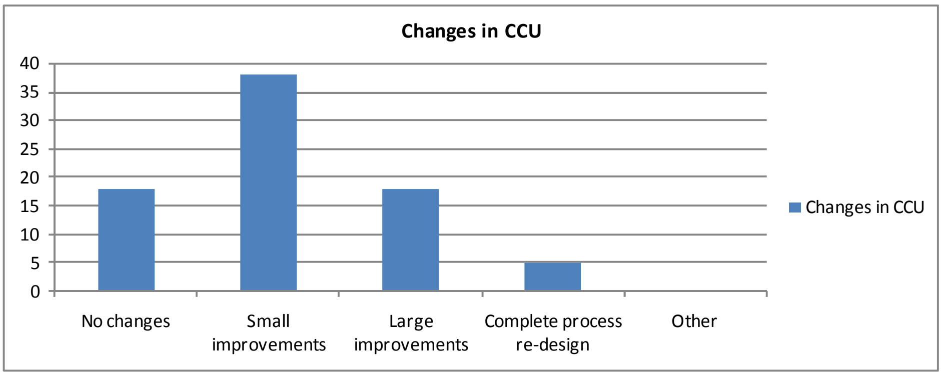


Figure 110: Changes in CCU.

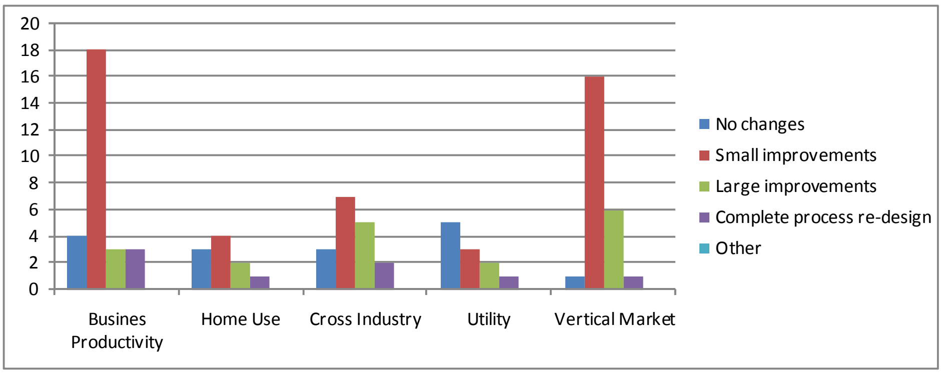


Figure 111: Changes in CCU per industry.

Question 2: These developments have influenced your organisation in the following way.
Your answer:

Lower release & deployment costs	Strongly agree
Higher product quality	Strongly agree
Shorter release cycle(s)	Strongly agree
Fewer installation problems	Strongly agree
Finding and resolving bugs takes less time	Neither agree nor disagree
More stable product	Agree
More customer knowledge	Agree

Table 35: Impact of change.

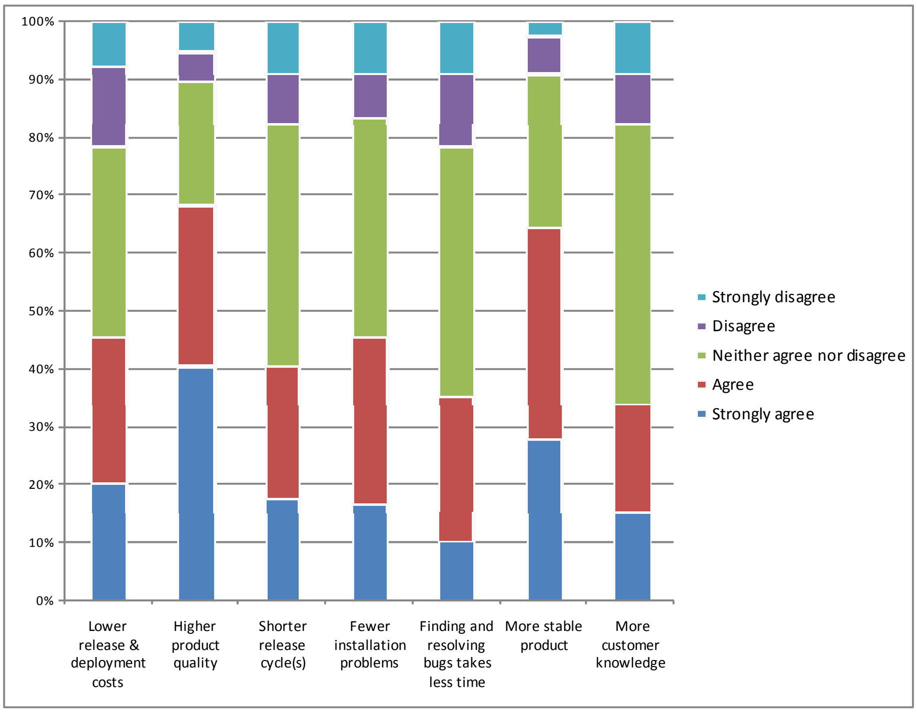


Figure 112: Impact of change.

Question 3: In comparison with your competitors, do you feel that your release, delivery and installation processes are.
Your answer: More advanced

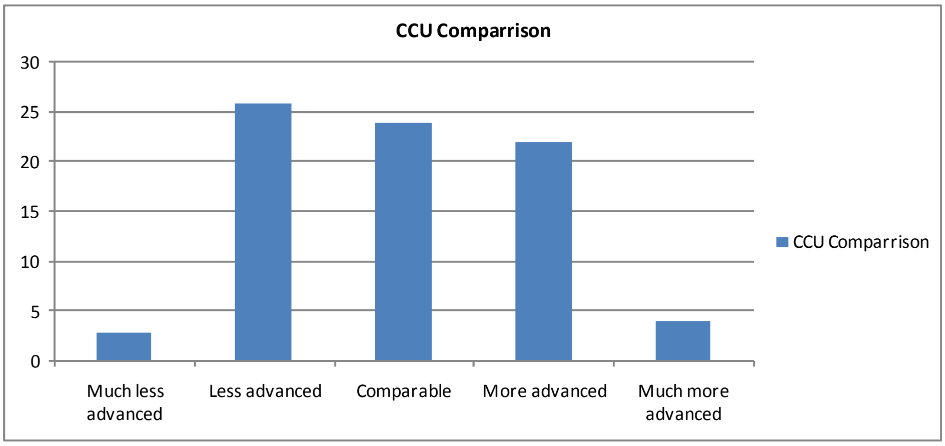


Figure 113: CCU Comparrison.

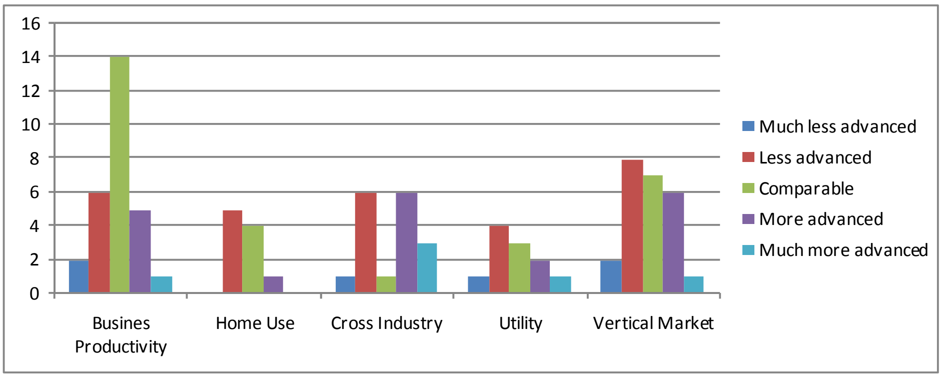


Figure 114: CCU Comparrison per industry.

Question 4: Please check the reasons why you would improve, or have improved, your Customer Configuration Updating processes.

Your answer:

Serve more customers	No
Shorten the release cycle	Yes
Shorten the time in which bugs are found	No
Serve customers more cost	Yes
Reduce the number of installation	Yes
Administer a more flexible licencing	Yes
Other	No

Table 36: Reasons for improvement.

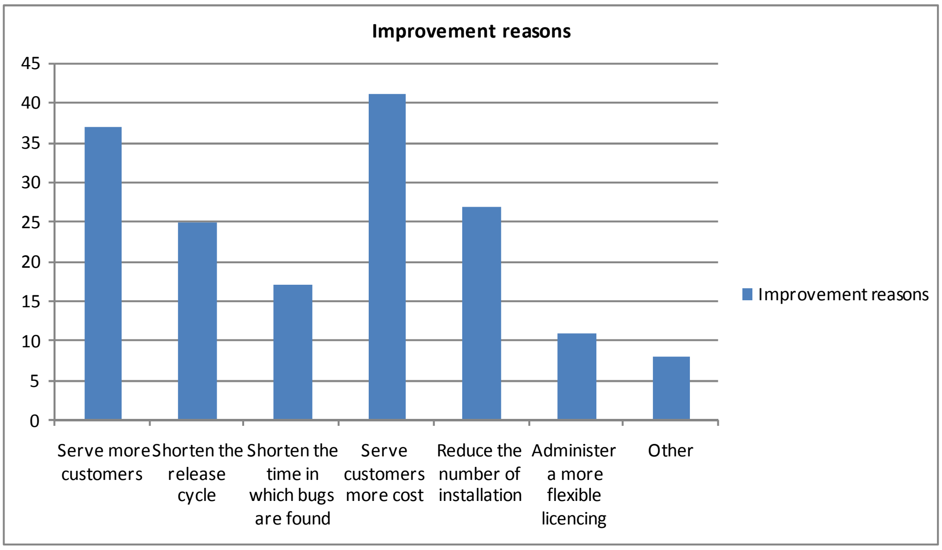


Figure 115: Reasons for improvement.

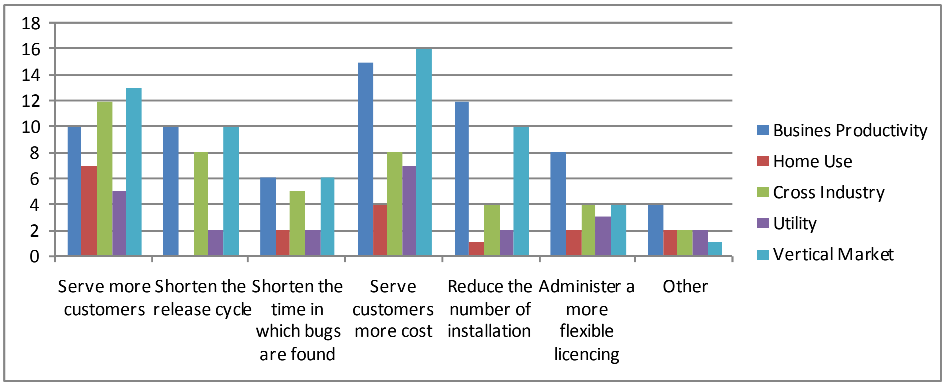


Figure 116: Reasons for improvement per industry.

Question 5: Please indicate how your product developed over the last two years.
Your answer: The product is much more succesfull than two years ago

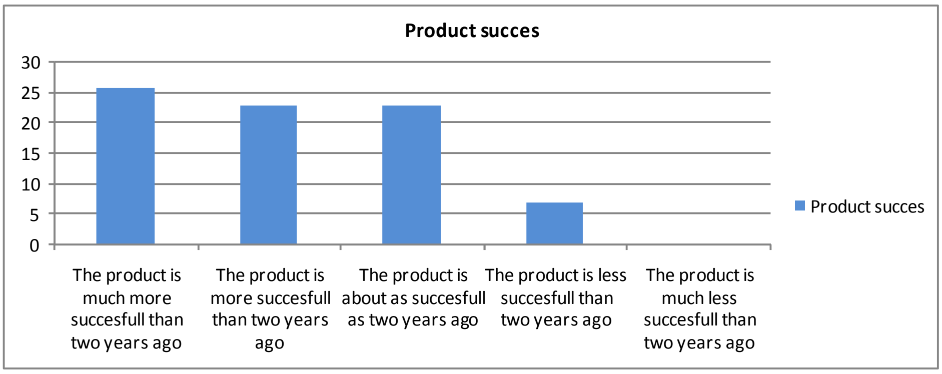


Figure 117: Product success.

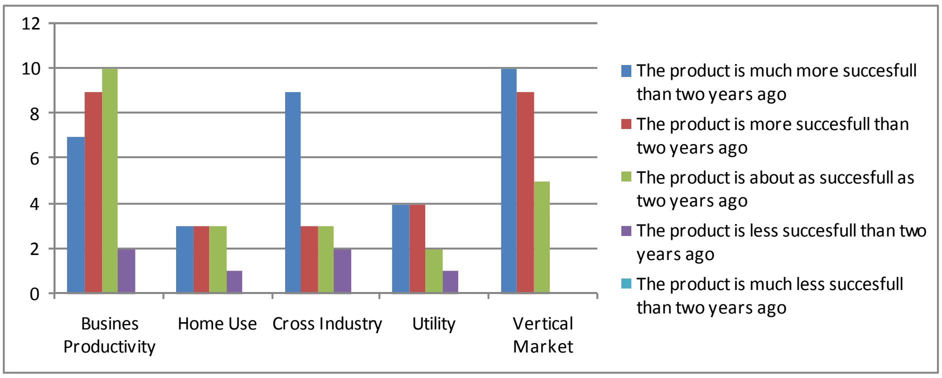


Figure 118: Product success per industry.

Question 6: Please indicate how the product itself has been influenced by changes in the release, deliver and installation process?
Your answer: The developement of the product is partly influenced by the changes

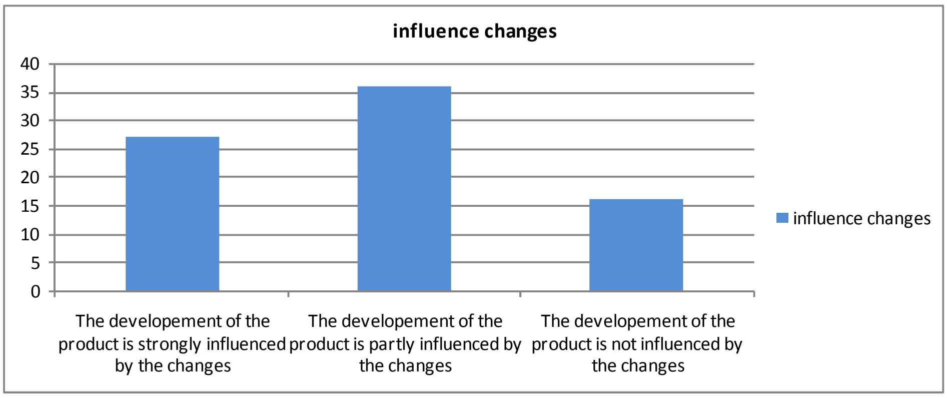


Figure 119: Influence changes.

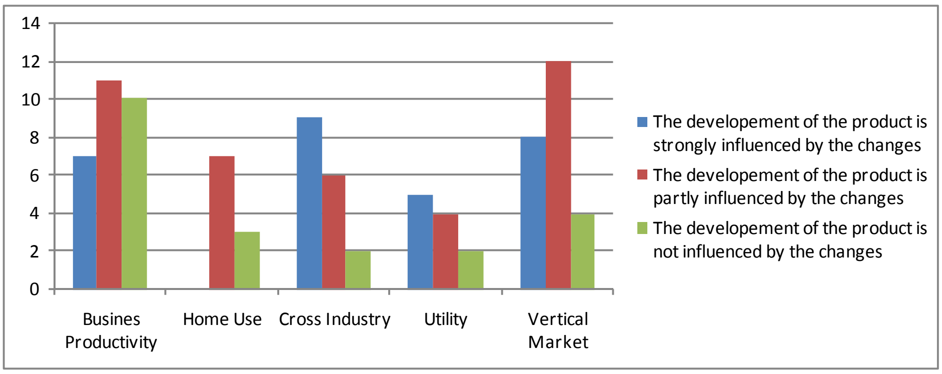


Figure 120: Influence changes per industry.

Question 7: We see the CCU process as a:
Your answer: Neither high or low priority process

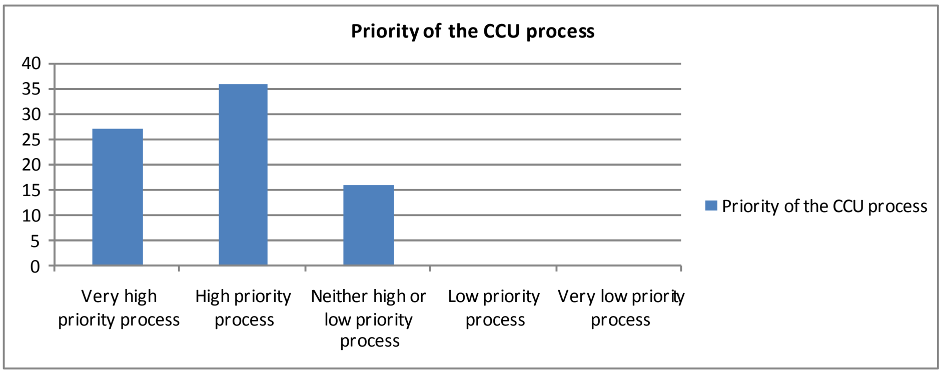


Figure 121: CCU process priority.

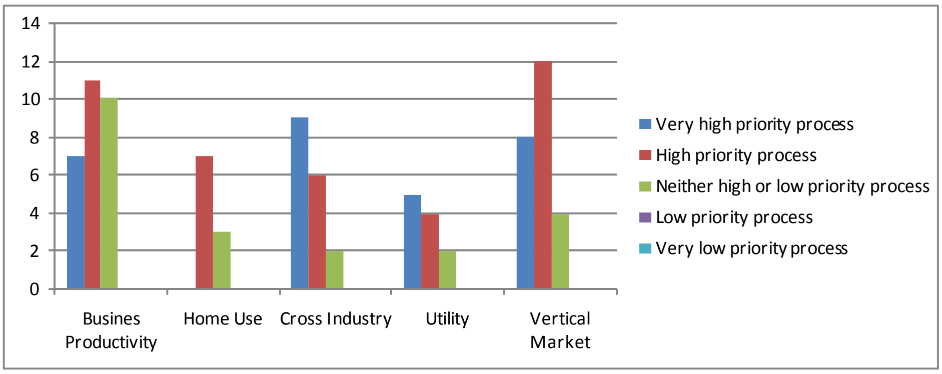


Figure 122: CCU process priority per industry.

4.14 CCU process future

Question 1: What Customer Configuration Updating best practices would you recommend to others?
See next chapter for open questions

Question 2: What are your major concerns for the next 5 years, with regard to the CCU process?
See next chapter for open questions

Question 3: Where do you see release processes evolve to?
See next chapter for open questions

Question 4: Please provide the top three problems you would like to see solved in your CCU processes.
See next chapter for open questions

Question 5: What is the highest priority problem to solve for your product (please indicate why)?
See next chapter for open questions

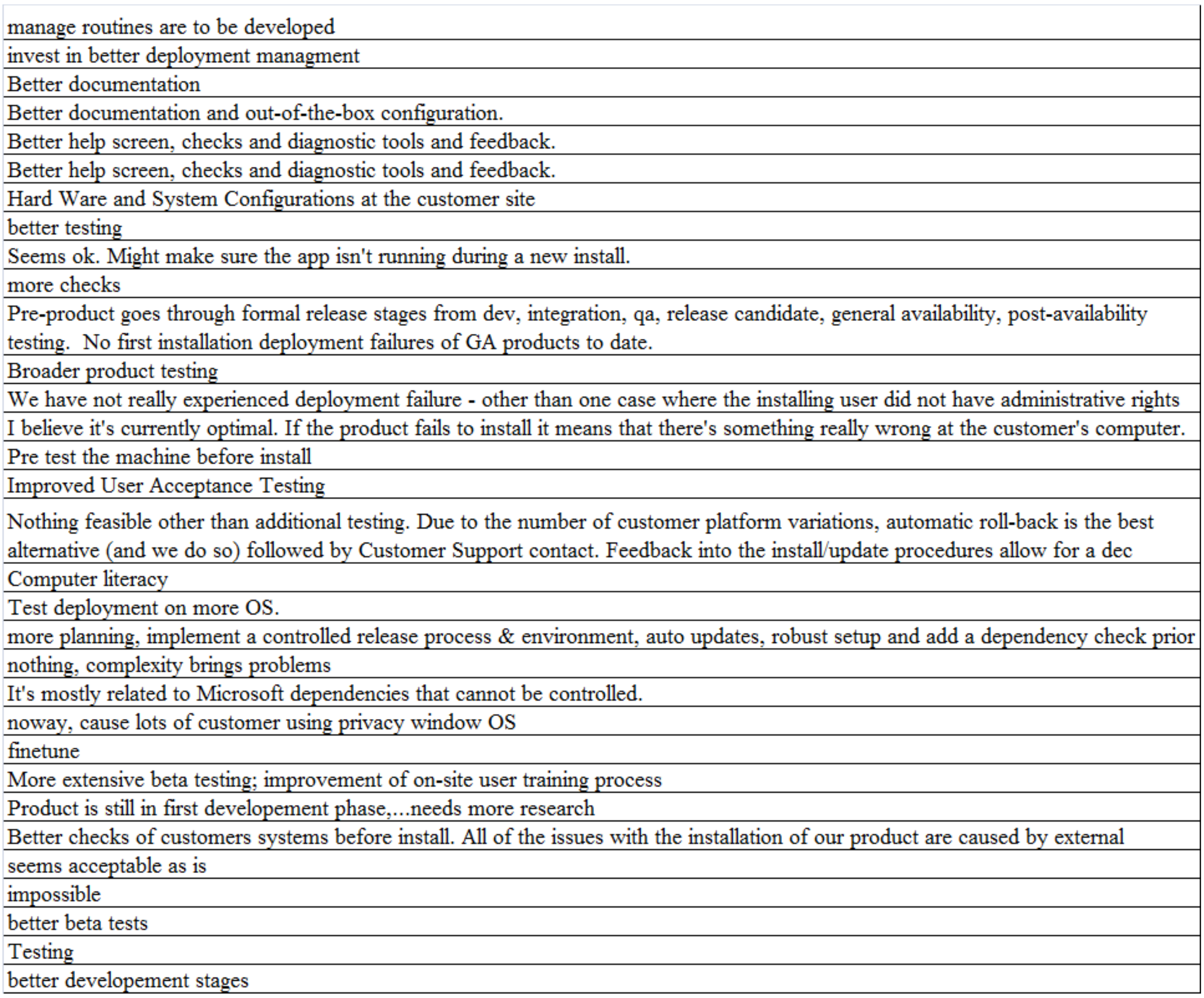
Question 6: What tools that you use for supporting the Customer Configuration Updating process would you recommend to others?
See next chapter for open questions

Question 7: What would your ultimate solution look like?
See next chapter for open questions

Chapter 5. Answers to open questions.

5.1Deployment failure

*InstallationMethods/*Question 13: Please indicate the rate of deployment failure, for your product – What can be done to reduce this figure?
Your answer: Not sure. We only had a few installation problems at customer sites.
Other Answers:



manage routines are to be developed
invest in better deployment managment
Better documentation
Better documentation and out-of-the-box configuration.
Better help screen, checks and diagnostic tools and feedback.
Better help screen, checks and diagnostic tools and feedback.
Hard Ware and System Configurations at the customer site
better testing
Seems ok. Might make sure the app isn't running during a new install.
more checks
Pre-product goes through formal release stages from dev, integration, qa, release candidate, general availability, post-availability testing. No first installation deployment failures of GA products to date.
Broader product testing
We have not really experienced deployment failure - other than one case where the installing user did not have administrative rights
I believe it's currently optimal. If the product fails to install it means that there's something really wrong at the customer's computer.
Pre test the machine before install
Improved User Acceptance Testing
Nothing feasible other than additional testing. Due to the number of customer platform variations, automatic roll-back is the best alternative (and we do so) followed by Customer Support contact. Feedback into the install/update procedures allow for a dec
Computer literacy
Test deployment on more OS.
more planning, implement a controlled release process & environment, auto updates, robust setup and add a dependency check prior
nothing, complexity brings problems
It's mostly related to Microsoft dependencies that cannot be controlled.
noway, cause lots of customer using privacy window OS
finetune
More extensive beta testing; improvement of on-site user training process
Product is still in first developement phase,...needs more research
Better checks of customers systems before install. All of the issues with the installation of our product are caused by external
seems acceptable as is
impossible
better beta tests
Testing
better developement stages

Product rarely fails at deployment.
Better segregation of client-specific data/configuration parameters from common ones (in both our and 3rd party components)
To complicated to explain
Better research
better installation support
not much, there are to much variables to be 100% effective
Installation tool that does better checks
It seems out of our hands and up to Microsoft
streamline hardware
classified
move to open source tools
I fix all issues as they arise with new releases.
Better developement tests
change the deployment procedure
Better help screen, checks and diagnostic tools and feedback.
No idea. The 1% number is a guesstimate - interested parties will contact us in case of problems and we always solve them.
I do not plan any improvements.
Not sure. We only had a few installation problems at customer sites. These were solved by the customer simply rebooting and re-installing. We are still not sure the cause.
There is nothing to be done, there is no failure on deployment.
Better installation checks, simplify the installation process.
not applicable
To think and to work
The deployment it is very easy, it is just an installer
more research
The product is an IDE for the Z++ language, like Visual C++ of Microsoft. There is no need for undoing anything, etc. It does not interfere with user previous written

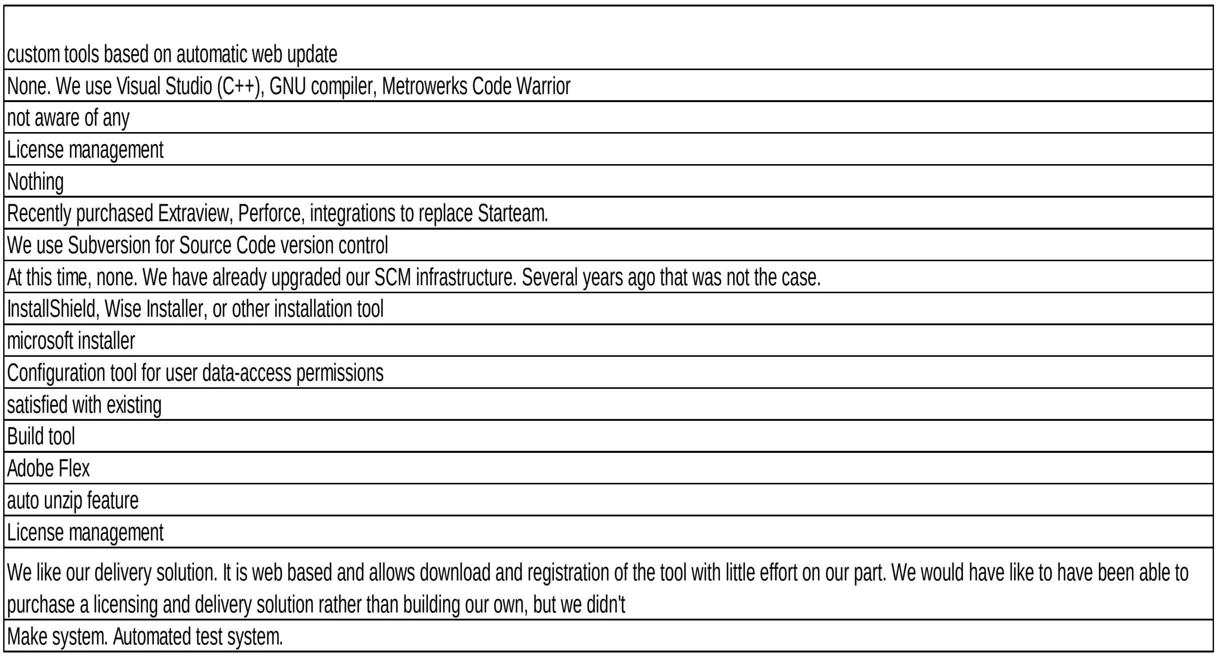
Table 37: Reduce rate of deployment failure

Deployment failure solutions

With an average of about eight percent of deployment failure product software vendors will have to start finding ways to improve the stability of their products. When products managers (a.o.) are asked how to reduce these problems some general trends can be distinguished. First of all the experts indicate that testing procedures during the development phase should be improved. Companies will have to invest in deployment management. Other areas that deserve extra attentions are user acceptance testing and in depth code review before release. Different experts indicate that they use a automatic roll-back procedure as the best alternative for installation problems due to the number of customer platform variations. After a rollback customers are able to contact support. This shows that more comprehensive beta testing should be incorporated by more vendors. At the same time vendors should produce high quality documentation and better support. Experts agree that more planning, a controlled release process & environment should be implemented, auto updates should be used, a more robust setup process should be developed and dependency checks should be added prior to installation.

5.2 Custom buid tools/tools rather purchases commercially.

*Tools/*Question 1: Compared to the custom–built tools you use now, what custom–built tool would you rather have purchased commercially (if any)?
Your answer: Our delivery tool is integrated in all company services.
Other Answers:



custom tools based on automatic web update
None. We use Visual Studio (C++), GNU compiler, Metrowerks Code Warrior
not aware of any
License management
Nothing
Recently purchased Extraview, Perforce, integrations to replace Starteam.
We use Subversion for Source Code version control
At this time, none. We have already upgraded our SCM infrastructure. Several years ago that was not the case.
InstallShield, Wise Installer, or other installation tool
microsoft installer
Configuration tool for user data-access permissions
satisfied with existing
Build tool
Adobe Flex
auto unzip feature
License management
We like our delivery solution. It is web based and allows download and registration of the tool with little effort on our part. We would have like to have been able to purchase a licensing and delivery solution rather than building our own, but we didn't
Make system. Automated test system.

Table 38: Custom build tools, rather have purchased commercially.

Custom made and commercially purchased tools

When investigating which custom made tools the company developed, and which of these they would rather have purchased commercially, around 50% of the experts indicate to be happy with the tools they developed. The other half would have made some changes if possible. A small group says they would rather have bought a solid commercial build tool to avoid a big part of current problems and have acces to support. Others state that custom tools based on automatic web updating are hard to check for correct behavior and ask a lot of maintenance. But the most heard problem by far is license management. A big part of the response group would like to be able to buy a CRM module accessible from clients-side (i.e. customers can recover their licenses themselves). A frequent response is: "We would have liked to have been able to purchase a licensing and delivery solution rather than building our own", but for most companies there are no licensing tools available that are customizable enough to integrate in their organization.

5.3 Missing tools

*Tools/*Question 2: In the customer configuration updating process, what tools do you believe are missing at this moment?
Your answer: Integration of tools
Other Answers:

custom tools based on automatic web update
Tier upgrade
we just use emacs on xml files, so a nice gui might help beginners
Update tools.
Automatic ties between contracts, licenses and web-accessability (security).
A proper CRM/license integration.
MDAC, Memory, online facilities
Our customer problem tracking system is a separate system from our installation system. They should be integrated. Our problem tracking system requires us to add users ourselves and the user identification of problem tracking should be integrated with the
Not an issue
check for update
Automatic builds (i.e. "nightly builds")
we are fine...
good installing tools
We don't have the model of updating at customer site since we host the customers online. Many of your questions are irrelevant for us.
automation (auto-update process)
Automated RS-232 interfaces to user-owned medical equipment
I feel as though we need to signifigantly improve our update success/failure reporting tools to be able to assist our customers in the best manner possible.
Platform specific installers
all of them
The ability to revoke a license
hardware census
A proper CRM/license integration.
A proper (web-based) full-cycle planning and project-management tool integrating the full lifecycle management process of a product.

Table 39: Missing tools.

Missing tools

When prompting experts with questions what tools they think that are missing at the moment, a lot of responses are generated. There is some demand for customizable tools that will facilitate the web updating. On a higher scale experts are talking about a proper (web-based) full-cycle planning and project-management tools, integrating the full lifecycle management process of a product. But the main focus with regard to missing tools is on two aspects. Firstly, there is a need for proper installment tools. Experts report that their customer problem tracking system is a separate system from their installation system. They should be integrated to enable a efficient CCU process. New tools should improve on this by offering improved update success/failure reporting. Secondly, extra attention should be given to the licensing process. A lot of experts would like to see a proper CRM/license integration. Tools should offer automatic ties between contracts, licenses and web-accessibility (also with regard to security) and should incorporate the ability to revoke a license.

5.4: Tools to avoid

*Tools/*Question 3: What tools for the Customer Configuration Updating process would you like to dissuade others from using based on your experiences?
Your answer: Our tool is Eclipse based
Other Answers:

auto reprt sending
non-textual configuration files are hard to manage
Online Facilities
Starteam
Installshield
ones not taking into account user permissions.
We found CVS use to be problematic
cvs for java development.
manual update feature
InstallShield; it is very expensive, very crappy, and completely unnecessary.
Do not know.
Our tool is Eclipse based and we originally used the Eclipse plug-in installation solution for our delivery and this was a disaster. We shifted over to an .msi based installation solution with a web site to deliver the .msi and this was much better.
Not an issue
We only use a installer that does this process.
None for our type of product.

Table 40: Dissuade tools.

5.5: CCU best practices

*ProcessFuture/*Question 1: What Customer Configuration Updating best practices would you recommend to others?
Your answer: No Answer
Other Answers:

Using a version control system!
Online Supports
I've been happy with NSIS.
Internal custom database, strong revision controls.
Listen to feedback from customers. They can really help to improve the product.
Let customers drive changes and dont push it to them.
it depends on software type
Release what you test (i.e. do not use a separate release build if at all possible). Integrate Version Control (VC), Defect-Issue-Enhancement Tracking (DIET), Build and Release systems. Use Automated Life Cycle Automation (ALA) tools where possible. KE
Develop a formal CCU process, used automated tools, add logging reporting early in the product development lifecycle
I am still waiting for the next generation scenario, because the current one is not satisfactory.
network setting
as much internal testing before release as possible.
Don't just collect user data. Analyze it within the context of defined goals
Focus on it and make it a core competency
include an auto backup feature to safeguard customer data
Integrate with the platforms you're delivering on; use their native installers, put files where the customer expects them to be, etc.
Do not know.
Inter-version compatability. Currently, when a customer upgrades either our product or its configuration all installed clients must be updated.
None, we only deploy an installer

Table 41: Customer Configuration Updating best practices.

Best practices

When looking for current trends and the way the process will evolve in the future it is useful to look at what experts perceive as CCU best practices. Through this exercise the reader will be able to form an image of what is most interesting to experts in the field, and what might be must-have features in new generation tools.

With regard to the development of new products or choices for new updates experts indicate that one should listen to feedback from customers. This can really help to improve the product. Vendors should let customers drive changes and not push their own vision to them. This implicitly asks for more elaborate feedback tools. As an additional remark experts say that user data should not just be collected, but vendors should analyze it within the context of defined goals.

Furthermore, vendors should actually release what they test. In other words they should not use a separate release build if at all possible. An addition to this is that an version control system is a must for every software developer. This version control system should be integrated with Defect-Issue-Enhancement Tracking (DIET), Build and Release systems.

Updating tools should use Automated Life Cycle Automation (ALA) tools where possible, have the ability to track release levels of product in use in the field to minimize support issues, as well as the ability to support special variant releases and international builds. Also, it should support the implementation of release postmortem analysis with feedback into the development and CM processes and offer enhanced metrics analysis throughout the life cycle. And of course a tool should include an auto backup feature to safeguard customer data.

Experts want to simplify and automate the upgrade process as much as possible but increase the number of features at the same time. With regard to the product / update:

- There should be done as much (internal) testing before release as possible.
- A formal CCU process should be realized,
- Automated tools should be used,

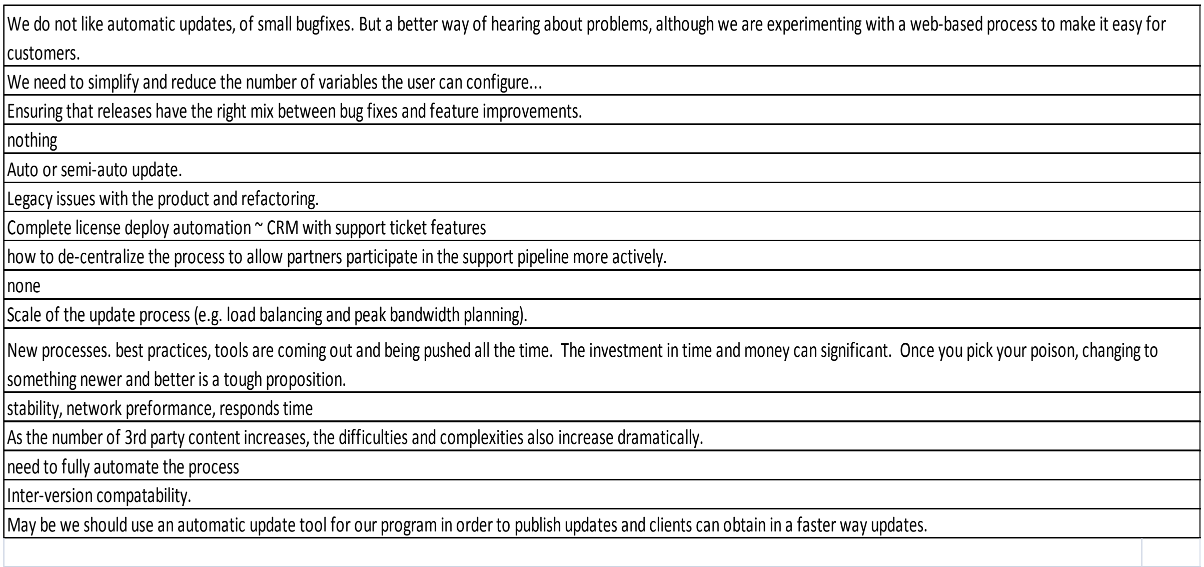
Feedback reporting should be taken seriously.

5.6: Major concerns for the next 5 years.

*ProcessFuture/*Question 2: What are your major concerns for the next 5 years, with regard to the CCU process?

Your answer: No Answer

Other Answers:



We do not like automatic updates, of small bugfixes. But a better way of hearing about problems, although we are experimenting with a web-based process to make it easy for customers.
We need to simplify and reduce the number of variables the user can configure…
Ensuring that releases have the right mix between bug fixes and feature improvements.
nothing
Auto or semi-auto update.
Legacy issues with the product and refactoring.
Complete license deploy automation ~ CRM with support ticket features
how to de-centralize the process to allow partners participate in the support pipeline more actively.
none
Scale of the update process (e.g. load balancing and peak bandwidth planning).
New processes. best practices, tools are coming out and being pushed all the time. The investment in time and money can significant. Once you pick your poison, changing to something newer and better is a tough proposition.
stability, network preformance, responds time
As the number of 3rd party content increases, the difficulties and complexities also increase dramatically.
need to fully automate the process
Inter-version compatability.
May be we should use an automatic update tool for our program in order to publish updates and clients can obtain in a faster way updates.

Table 42: Major concerns for the next 5 years

Concerns for the next five years

The three main pillars of concerns are stability, automation and upcoming new processes. Stability is a concern because vendors are forced to create more efficient processes. This forces them to simplify the process and reduce the number of variables the user can/should configure. Experts indicate that they need to clean up their frameworks, so it is becoming easier for them to keep them stable. Current problems and software crashes create a huge customer dissatisfaction, which is exactly what one does not want in the current customer focused trend. Simplifying the CCU process is a noble goal but as the number of 3rd party content increases, the difficulties and complexities also increase dramatically.

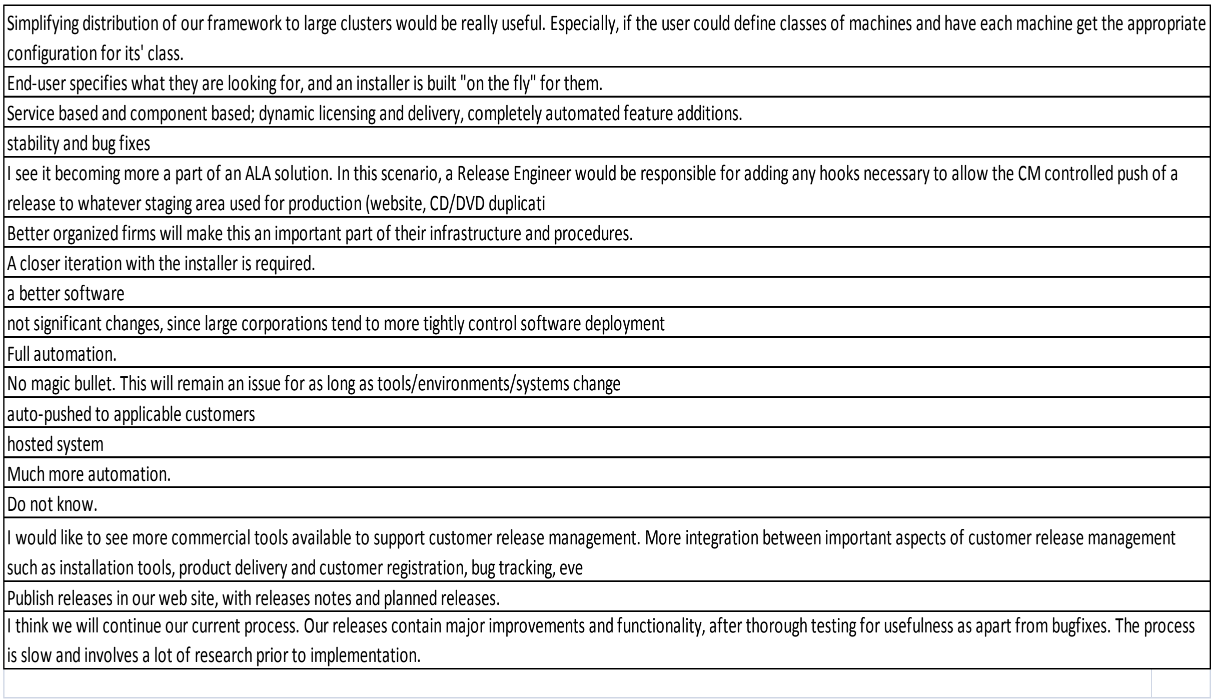
The second concern is about automation. The majority of experts has implemented, is busy with (or is thinking about) implementing complete license deployment automation. They see the need to fully automate the process, but do not know how to approach the problem.

The final main concern is that of new processes. New ideas, processes, best practices, and tools are coming out and are being pushed at a fast pass. The time and money investment with regard to keeping up with these trends can significant. Once one picks his poison, changing to something newer (and better) is a tough proposition.

Some other concerns to keep in mind are, ensuring that releases have the right mix between bug fixes and feature improvements, legacy issues with the product and refactoring and how to de-centralize the process to allow partners participate in the support pipeline more actively.

5.7 Evolving CCU process.

*ProcessFuture/*Question 3: Where do you see release processes evolve to?
Your answer: No Answer
Other Answers:



Simplifying distribution of our framework to large clusters would be really useful. Especially, if the user could define classes of machines and have each machine get the appropriate configuration for its' class.
End-user specifies what they are looking for, and an installer is built "on the fly" for them.
Service based and component based; dynamic licensing and delivery, completely automated feature additions.
stability and bug fixes
I see it becoming more a part of an ALA solution. In this scenario, a Release Engineer would be responsible for adding any hooks necessary to allow the CM controlled push of a release to whatever staging area used for production (website, CD/DVD duplicati
Better organized firms will make this an important part of their infrastructure and procedures.
A closer iteration with the installer is required.
a better software
not significant changes, since large corporations tend to more tightly control software deployment
Full automation.
No magic bullet. This will remain an issue for as long as tools/environments/systems change
auto-pushed to applicable customers
hosted system
Much more automation.
Do not know.
I would like to see more commercial tools available to support customer release management. More integration between important aspects of customer release management such as installation tools, product delivery and customer registration, bug tracking, eve
Publish releases in our web site, with releases notes and planned releases.
I think we will continue our current process. Our releases contain major improvements and functionality, after thorough testing for usefulness as apart from bugfixes. The process is slow and involves a lot of research prior to implementation.

Table 43: Evolving release processes.

Evolving CCU process

It is established that there are some concerns for the upcoming years. How then do these experts think the CCU process will evolve over the upcoming years. Most experts believe that better organized firms will make CCU a more important part of their infrastructure and procedures. And the only way to do this is to automate a big part of the processes.

This will also involve support automation; most common questions from customers (there's a FAQ though). Other thoughts are proactive automated feedback from the Customer Support and Field Engineering groups to CM.

Another idea is to connect functionalities of for instance AnthillPro, AccuRev, Bugzilla, SalesForce (or a FOSS equivalent CRM). Vendors should implement all of the bridges/interfaces between the tools (like the ones mentioned) necessary to provide full traceability.

Experts think that more commercial tools will become available to support customer release management. More integration between important aspects of customer release management such as installation tools, product delivery and customer registration and bug tracking is needed. Others think that there will be no magic bullet and no significant changes, since large corporations tend to more tightly control software deployment. These problems will remain issues for as long as tools/environments/systems change.

5.8 Problems

*ProcessFuture/*Question 4: Please provide the top three problems you would like to see solved in your CCU processes.
Your answer: No Answer
Other Answers:

need a way to show new configuration options
The biggest challenge is ensuring continuity of the release process as individuals come and go within the team.
Understanding the issues
Automating authoring of product updates; transition of QA automation tools to customer facing.
Automate in some way the most common questions from customers (there's a FAQ though)
Making it easy to distribute builds to the customer.
Proactive automated feedback from the Customer Support and Field Engineering groups to CM. Implementation of release postmortem analysis whith feedback into the development and CM processes. Enhanced metrics analysis throughout the life cycle.
Reduce complexity, Increase value-add, Reduce install/config/update failure rates
Installation progress indicator.
Installation, software crash, network responds
no problems.
Lack of company-wide understanding of process. Developer resistance of adhering to policy.
How to distribute add-ons or extensions to the customer."
Registration, planning and tracking of bugfixes/feedback, requests and regular development .
Do not know.
[I'm assuming CCU is the step that comes after release, deliver, install], Simplify the upgrade process.
We try to wait for a few months (to make it almost a year) prior to each release. This is more for ensuring that we are improving the product in useful ways other than minor features. Most of the work is on the linguistic abstractions. So, we are not sure

Table 44: Top three CCU problems.

Problems

While working towards the goals mentioned in the sub paragraphs above, experts indicate to bump into problems. They have to re-design and re-implement the internal modules, which have become obsolete. Many of the feature requests by customers cannot be implemented at this time due to limitations in current architecture, which is the base of the old internal modules, etc. A lot of companies experience problems improving their processes because they fail to:

- Make it easy to distribute builds to the customer.
- Incorporate more automated testing in the Build/Release stages.
- Create company-wide understanding of the (importance of the) process.
- Integrate with 3rd party tools and devices via automated means.
- Create formalized CCU process and arrange management buy-in.
- Create a better way of controlling the use of our product in accordance to the contracts.

These are examples of problems that can be found in a broad range of companies; new updating tools will have to help overcome them.

5.9 Priority problems

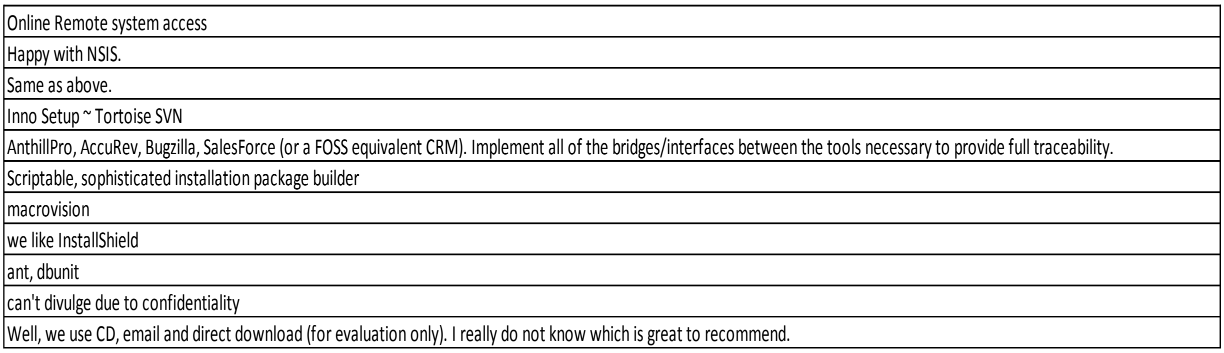
*ProcessFuture/*Question 5: What is the highest priority problem to solve for your product (please indicate why)?
Your answer: No Answer
Other Answers:

Several features unrelated to CCU.
Find a publisher
Re-design and re-implement the internal modules, which are now obsolete. Many of the feature requests from customers cannot be implemented at this time due to limitations in current architecture, which is based in the old internal modules.
Both major and minor releases generally involve significant code refactoring. This makes both bug fix integration and automated regression testing more difficult. Our new VC system is helping address some of this, but not to the level we need from an over .
Need formalized CCU process and require management buy-in
To adjust the better operation with customers, because in this area we have no experts.
none, product is stable
configuration management for metadata repositories, in an environment w/ many source code branches, without imposing the need to hire an army of engineers
instead stability in most environments
marketing
Do not know.
Corporately we give our release processes a very low priority. Within that domain reducing installation failures and improving product quality are high priority.
Improve update tool, in order customer can have bugfixes faster
We need to have a better way of controlling the use of our product in accordance to the contracts. For instance, it is priced per developer, but we cannot be sure if that is honored by customer.

Table 45: Highest priority problem.

5.10 Tools to reccomend

*ProcessFuture/*Question 6: What tools that you use for supporting the Customer Configuration Updating process would you recommend to others?
Your answer: No Answer
Other Answers:

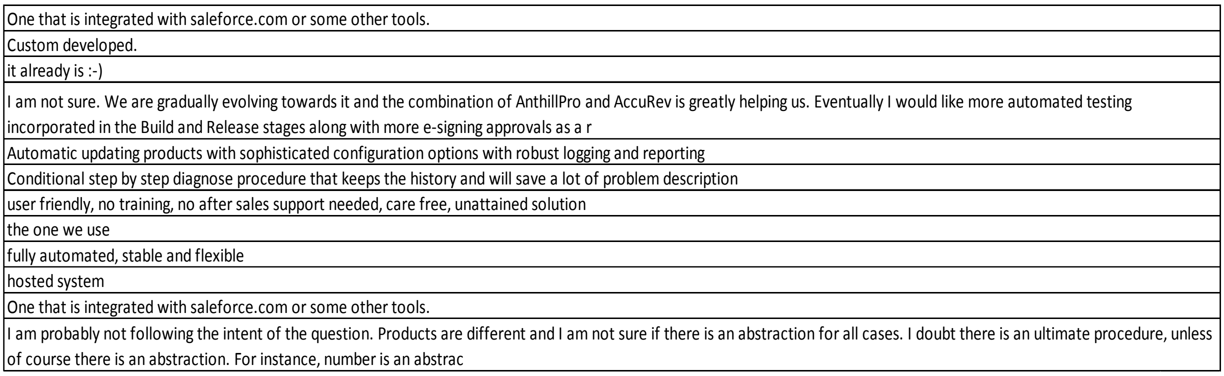


Online Remote system access
Happy with NSIS.
Same as above.
Inno Setup ~ Tortoise SVN
AnthillPro, AccuRev, Bugzilla, SalesForce (or a FOSS equivalent CRM). Implement all of the bridges/interfaces between the tools necessary to provide full traceability.
Scriptable, sophisticated installation package builder
macrovision
we like InstallShield
ant, dbunit
can't divulge due to confidentiality
Well, we use CD, email and direct download (for evaluation only). I really do not know which is great to recommend.

Table 46: Recommended tools.

5.11 Ultimate CCU solution

*ProcessFuture/*Question 7: What would your ultimate solution look like?
Your answer: No Answer
Other Answers:



One that is integrated with saleforce.com or some other tools.
Custom developed.
it already is :-)
I am not sure. We are gradually evolving towards it and the combination of AnthillPro and AccuRev is greatly helping us. Eventually I would like more automated testing incorporated in the Build and Release stages along with more e-signing approvals as a r
Automatic updating products with sophisticated configuration options with robust logging and reporting
Conditional step by step diagnose procedure that keeps the history and will save a lot of problem description
user friendly, no training, no after sales support needed, care free, unattained solution
the one we use
fully automated, stable and flexible
hosted system
One that is integrated with saleforce.com or some other tools.
I am probably not following the intent of the question. Products are different and I am not sure if there is an abstraction for all cases. I doubt there is an ultimate procedure, unless of course there is an abstraction. For instance, number is an abstrac

Table 47: Ultimate CCU solution.

Chapter 6. Detailed scores with questions

The scores below show the bottom line of this benchmark research. The scores are based upon expert interviews and a second survey amongst product software vendors and lead researchers in the release and deployment area. Eight experts indicated for every topic how important this topic is with regard to the practice. This was done to create an practice based view in order to be able to compare questions to each other. This resulted in them, assigning points to the different questions.This enabled the researchers to derive how the differen questions relate to each other.

6.1 Release

1. RELEASE

1.1 Release Frequency.

Proces	Practice	Description of Question	Your Score	Max Score
Release	A	The average time between the last major updates.	0	5
Release	B	The average time between the last minor updates	0	3
Release	C	The average time between the last bugfix updates.	0	3
Release	Total	Do you release your product/updates regularely?	0	11

1.2 Release planning.

Release	A	A formal release planning is utilized, in which specific dates are indicated with regard to the upcoming major, minor, and bug fix releases.	0	4
Release	B	The release planning is published in such a way that all relevant product stakeholders can access this planning at all times.	0	3
Release	C	There is a formal publication policy with regard to the release planning document.	0	2
Release	D	Release of updates and products at times that are convenient with regard to your customers.	4	4
Release	Total	Do you publish a release planning with data within your organisation?	4	13

1.3 Release scenario.

Release	A	A formalized release scenario is present within the organization that describes what happens step by step on release days.	0	4
Release	B	Releases are saved in a specific format/place (like CVS).	5	5
Release	C	All major, minor and bug fix releases can be accessed and downloaded.	4	4
Release	Total	Is there a formalised release scenario that describes all 'to be taken' steps?	9	13

1.4 Release management of (ext-)components.				
Release	A	All custom-built tools that are used by the organization to support the CCU process, like installation programs and scripts, are managed explicitly, just as if they were purchased externally.	4	4
Release	B	All external dependencies between your product and external components are managed explicitly in a computer readable format (e.g. Our Product requires MySQL).	3	3
Release	C	All commercial and open–source tools that are being used for development and support of the CCU process are managed explicitly.	4	4
Release	D	Are the components saved in a repository (at the company side), such that version compatibilty is preserved?	0	4
Release	Total	External products like products of the shelf are managed explicidly.	11	15

Total	**All**	**Release**	24	52

6.2 Delivery

2. DELIVERY

2.1 Delivery distribution channel.				
Delivery	A	The customer is informed through different channels (e.g. website/newsletter etc.).	1,5	4
Delivery	B	All releases can be downloaded by all stakeholders.	3	3
Delivery	C	Customers can report bugs through different channels.	3,5	5
Delivery	D	The product can be delivered in different package format(s) (e.g. MSI, exe, zip).	1	3
Delivery	Total	Is the customer regularry approached?	9	15

2.2 Delivery distribution Method.				
Delivery	A	The customer is frequently informed about the product (at least once a year).	1	4
Delivery	B	Your product can be delivered through automatic push or pull.	1	3
Delivery	C	Products can be delivered in different ways (e.g. Mail, DVD, FTP).	1	4
Delivery	D	The product update tool makes it possible to download the product from every location, not only from your release repository.	2	2
Delivery	Total	Do you use as much chanels as possible to deliver the product?	5	13

Total	**All**	**Delivery**	14	28

6.3 Deployment

3. DEPLOYMENT

3.1 Deployment dependency management.				
Deployment	A	The product can verify at any point in time after its been successfully installed if its critical dependencies are still properly installed and configured.	0	4
Deployment	B	External relationships between your and other products are managed.	0	4
Deployment	Total	Do you use explicit dependency management for correct installation?	0	8

3.2 Deploy removal.				
Deployment	A	It is possible to de-install the product without executing complicated manual operations.	4	4
Deployment	B	It is possible to undo an update.	3	3
Deployment	Total	Is your product removable?	7	7

3.3 Deployment customization management.				
Deployment	A	The product uses an update tool, which updates on the customer-side of the product.	0	4
Deployment	B	It is possible to install the product in a DTAP environment, where new versions progress from an unstable to a stable production environment.	3	3
Deployment	C	The update tool is able to update during runtime.	0	2
Deployment	D	The update tool is able to cope with customizations, extensions and customer-specific solutions.	0	5
Deployment	Total	Do you use an update tool and can it cope with customer specific solutions?	3	14

3.4 Deployment product reliability.				
Deployment	A	When problems occur during installation, they can be addressed automatically.	0	3
Deployment	B	Your product checks the local customer configuration before installation (such as the needed amount of disk space).	4	4
Deployment	C	You are aware of how your customers generally customize the product.	3	3
Deployment	D	You are aware of the hard- and software platforms that your customers use to operate your product.	4	4
Deployment	E	All the data produced by the user (like configuration settings and documents) is stored on a different location from the product, which makes it possible to back up the user data without backing up the product.	4	4
Deployment	F	The company has Pilot customers to test the product.	0	4
Deployment	Total	Update reliability and (semi-) automatic solutions.	15	22

Total	**All**	**Deployment**	25	51

6.4 Usage

4. USAGE

4.1 License usage.				
Usage and Activation	A	License agreements are used for your customers.	4	4
Usage and Activation	B	Your license contains user information and number of users.	4	4
Usage and Activation	C	Your product can be purchased per module.	4	4
Usage and Activation	D	Your licenses expire.	4	4
Usage and Activation	Total	You use licences that activate specific parts of the software.	16	16

4.2 Organizational license management.				
Usage and Activation	A	Licenses are generated automatically form sales contracts.	0	3
Usage and Activation	B	The customer is able to renew, extend or expand the license without any actions on your part.	0	3
Usage and Activation	C	The company regularly provides temporary licenses.	3	3
Usage and Activation	Total	Do you manager licences explicid and semi automatic?	3	9

4.3 Usage feedback/product knowledge.				
Usage and Activation	A	Your product sends automatic error reports when a error occurs in your product.	0	4
Usage and Activation	B	These automatic error reports are analyzed.	0	4
Usage and Activation	C	Your product generates usage reports.	4	4
Usage and Activation	D	These usage reports are analyzed.	3	3
Usage and Activation	E	You are aware of customers specific solutions for the product.	4	4
Usage and Activation	Total	Customer information is constantly being harvested and cleverly analyzed.	11	19

Total	**All**	**Usage**	30	44

Table 48: CCU (sub-) processes and practices.

Chapter 7. Conclusions

The increase in bandwidth, opportunities for software products in global markets, increasing numbers of customers, changing technologies, growing software markets and the increase in computer/hand held use in daily life. These are all external factors which will drive product software vendors to continuously improve their processes in order to keep up with changing environments and trends. With this growth of customer numbers, possibilities on a global market and new technologies to deliver the product globally without effort, it becomes of greater importance for software companies to focus on the interaction with the customer as well as the development of efficient ways to deliver the product.

This paragraph addresses some of the concusions generated by the research based on the survey data.In this research the state of the practice of the Customer Configuration Updating (CCU) process for international product software vendors was investigated. Which process properties need to be measured, in what smaller parts processes can be broken down and which parts of organization processes are part of the CCU cycle are not yet extensively addressed by earlier research. Different scholars present models that include parts of the process and a few try to capture it as a whole, but no standard model for the description and assessment for the CCU process has been appointed. The authors of this thesis see the necessity for such a model. Product software vendors spend to little resources on the CCU process which causes it to be inefficient.

Based on these developments the research questions of the present research is:

> ***Research Question: What is the current state of the practice of the Customer Configuration***
> ***Updating process within international product software companies?***

Context hypotheses with regard to this research question are:

> H1: Recent changes in the CCU process have significant impact on the success of the software product.
>
> H2: The priority that the vendor gives to the CCU process has significant impact on the success of the software product.
>
> H3: The eight general product characteristics have significant impact on the success of the software product.

H4: In their estimations for the future with regard to the state of the practice of CCU, product software vendors will
be looking for updating tools that fully automate the process but which are fully customizable as well.

H5: International product software vendors are more successful than Dutch product software vendors.

Changes in the CCU process have significant impact on the product success

Based on the analysis presented in this research with regard to the question if there is a relation between recent changes in the CCU process and the product success, the hypotheses is proven to be correct. Crosstab and correlation analysis show that there is a significant positive relation between these variables. During the analysis four different groups were established in the change/success crosstab, two of these groups supported the hypothesis where the other two did not. An additional question in the survey stated “do you think the recent changes have influenced the product success”; pre percentages of answering towards “yes” (on a three point scale) was high for the two groups supporting the hypothesis and low for the two groups not supporting the hypothesis. This finding, once more, proves the hypothesis to be true.

The priority that the vendor gives to the CCU process has significant impact on the product success

This second hypothesis tests the relation between the fact that a vendor does or does not give a certain amount of priority to the CCU process with regard to the successfulness of the product. After conducting the correlation test it became clear that this hypothesis was true as well. Interesting finding here is that of the respondents who declared to give a high priority to the CCU process, 85,7% thought their product to be much more successful.

Eight general product characteristics have significant impact on the success

The eight general product characteristics mentioned in this hypothesis are the product’s age, the number of employees, the number of customers, the number of developers, the turnover, the number of end users, the number of Kilo Lines Of Code and the number of natural languages. After computing the correlation values it became apparent that only two of these characteristics showed a significant relation to product success. Relations were found between product success and the number of customers as well as the number of KLOC. Especially the relation with the number of customers was unexpected. This finding turned out to be negative which means that as long as the amount of customers goes down the product success goes up. This result is explained by the fact that smaller startup companies are usually quite successful even though they have a small customer base and bigger companies doe not spend enough resources on the CCU process which has a negative impact on the product success. The other variables do not influence the software product’s success. This means that the hypothesis can be seen as incorrect.

Completely automated updating tools which are fully customizable are the future

The answer to this hypothesis was based on the collection and ordering of data from several open questions in the CCU survey containing expert opinions. Amongst others, interesting finds were; "*Vendors should further develop their updating tools to be able to check the deployment environment and be sure all needed resources are present and dependencies are in place"*and "*There is a need for an updating tool that partly supports the build process, simplifies the automatic updating process and provides a customizable license management module*". Overall, the experts indicated not to buy commercial updating tools because they are not customizable to their specific processes. But at the same time they wished they could buy such an updating tool, because building one from scratch resulted in a lot of bug fix problems and support investment. This confirms the hypothesis.

International product software vendors are more successful than Dutch product software vendors

This final hypothesis draws a comparison between the international data in the present research and the international data from former research by Jansen and Brinkkemper (2008). Execution of a independent T-test resulted in conclusion that Dutch product software vendors are significantly more successful that international software vendors. This result was unexpected and proved the hypothesis to be incorrect. Next to product success a percentage analysis was used to compare the Dutch and International data on a range of other aspects. Most important findings were that Dutch product software vendors work more efficient, Dutch product software vendors use twice as much programming languages, International product software vendors use twice as much pilot testers, international product software vendors are more customer focused, Dutch companies turn out to use components of the shelf in 30% more cases, international software vendors make better use of digital deliver media that are available to them and Dutch product software vendors use an updating tool at the customer site to deploy their product in 53% of the recorded cases where the international product software vendors only use such a tool in 30% of the cases.

The state of the practice of international product software companies

The answers conveyed in the paragraphs above show that CCU is a lively and interesting research field with still a lot of uninvestigated sources of information. Even though these are interesting results it is time to discuss the core of the research. Through the use of the adjusted CCU model by Jansen and Brinkkemper it became possible to build the skeleton of the different sub processes, practices and capabilities. After analyzing the survey data it was possible to check to what extend product software vendors implemented the different practices for every CCU sub process, which makes it possible to say something about the state of the practice of the CCU process withn product software companies.

Every sub process showed different results. International software vendors do not implement their release process very well. The frequency and management of (external) component practices are pretty well implemented in most cases, but the planning and scenario practices are poorly represented. The same can be said for the delivery process; most respondents implement the distribution channel practice, they inform their customer quite frequently and through modern channels but the distribution method is implemented to less extend. Again, only half of the practices are correctly implemented in the deployment process. The dependency management and removal practices are generally implemented quite well but the Customization management and product reliability practices deserve a lot of extra attention. Finally, the usage process is one of the best performing CCU processes. The majority implements the license usage and license management practices and a lot of companies are working on the actual usage practice. We think this is the best developed part of the process because this is where the vendors earn their money.

These survey results show that CCU is an underdeveloped process area and that the state of the practices can be indicated as average. The process area needs more attention, research and tooling to improve this trend. As can be seen in the paragraph above; the release delivery and deployment processes are in need of attention. Most of these processes don't even have half of their practices implemented.
Research and investments should particularly be focused on the use of release planning, release scenarios, update tools, usage reports and error reports. Only 45% of the international software vendors uses a release planning, 55% indicates to implement a release scenario, 30% uses an update tool, 35% use usage and error reports. Spending more attention to these processes will help vendors improve product success, customer retention rate and deployment reliability (Jansen, 2008).

To Conclude

We think that product managers are aware of the problems in the field of CCU and shortcomings in specific processes, but they are unable to prove the importance of these problems to management. The results of the hypotheses presented in this research might serve as leverage to convince management to free resources in order to improve the CCU process. At the moment management focuses on efficiency and cost cutting, but the only way to improve the company (on this area) is to invest. Research in this relatively new and unexplored area is beneficial to industry, customers and scientifically bodies and it therefore is expected that this document will form the foundation for further research.

Chapter 8. Company Participation

Are you interested in participating in CCU improvement projects? Would you like one of your collegues to undertake the survey? Utrecht University has extensive experience in providing consultancy in the area of software release and delivery. This work is frequently funded (in part) by either National Science Foundations or specific technology awards.

Some example projects we did in the past:

Cross analysis of a large software vendor with over 44 products - In the Netherlands a cross analysis was done of 44 products of a large software vendor in the Netherlands. Many process improvements were proposed and initiatives were started to share knowledge across product lines in the areas of release, delivery, deployment and usage and activation.

Licensing Solutions for a Growing Software Product - A software product in the Netherlands that would soon grow from 1000 to 2000 customers (an objective that was achieved a year later) required more automation in its licensing and license renewal process to cut costs. Utrecht University provided a number of process improvements of which most were implemented. Some tool research was done and a toolset was procured in the next year, based on the advice of Utrecht University.

Please contact Slinger Jansen at Utrecht University for more information.

dr. Slinger Remy Lokien Jansen
email : slinger@cs.uu.nl
Universiteit Utrecht
Institute of Information and Computing Sciences
Centrumgebouw Noord, B234
Padualaan 14, De Uithof
3584CH Utrecht

Chapter 9. Related Literature

9.1 Literature

Slinger Jansen and Sjaak Brinkkemper. **Definition and validation of the key process areas of release, delivery and deployment of product software vendors: turning the ugly duckling into a swan**. In proceedings of the International Conference on Software Maintenance (ICSM2006, Research track), September 2006. http://www.cs.uu.nl/people/slinger

Slinger Jansen and Wilfried Rijsemus: **Balancing Total Cost of Ownership and Cost of Maintenance Within a Software Supply Network**, proceedings of the IEEE International Conference on Software Maintenance (ICSM2006, Industrial track), Philadelphia, PA, USA, September, 2006.

Ilja Heitlager, Slinger Jansen, Remko Helms and Sjaak Brinkkemper: **Understanding the dynamics of product software development using the concept of coevolution** ,International Workshop on Software Evolvability, 0-7695-2698-5, pages = 16-22

Slinger Jansen and Sjaak Brinkkemper: **Ten Misconceptions about Product Software Release Management explained using Update Cost/Value Functions** , First International Workshop on Software Product Management

Slinger Jansen, Gerco Ballintijn, Sjaak Brinkkemper and Arco van Nieuwland : **Integrated development and maintenance for the release, delivery, deployment, and customization of product software: a case study in mass-market ERP software** - Journal of Software Maintenance and Evolution: Research and Practice Issue 2, volume 18, 2006

9.2 Final words

This report was created for Endeavors Technologies, and send to ahitomi@endeavors.com only. No rights can be derived from this report. All information related to this research is treated confidentially. For more information you can contact Wouter Buts:

Wouter Franciscus Tonnis Buts

email : wbuts@uci.edu