UNIVERSITY OF OSLO
Department of informatics

How much empirical evidence is there to support claims made by software engineering tool providers about the benefits of their tools, and how valid is this evidence?

Master thesis

60 credits

Espen Brunsvig

1. May 2008

Abstract

Software engineering (SE) practitioners should consider the empirical validity of claims set forth by SE tool providers in correspondence with an Evidence-based Software Engineering approach, but to do so, empirical evidence needs to be available. As far as I have managed to determine, I conduct the first study looking into claims made by SE tool providers and subsequently the amount and validity of empirical evidence available directly from the tool providers. The study indicates that 61% of the tools have associated claims with no valid empirical material to support them, while only 22% can be said to have a certain minimum amount of valid empirical evidence to back up some of the linked claims. Furthermore, in cases where independent empirical evidence is available, there are examples of that material giving a false impression unless thoroughly scrutinized. Additionally, for the vast majority of cases, data intended as empirical evidence is considered generally invalid and is largely based on case studies lacking in necessary detail. Half of the 23 organizations contacted responded and of these only two thirds sent material. The results are derived from critical appraisal of that material, as well as material found on SE tool providers' respective product websites, and are discussed with relation both to SE tool adoption and marketing. The study contributes by concluding that SE practitioners need to employ their power as consumers to enforce a paradigm change so that more valid independent empirical data will surface, to the definite benefit of the SE practitioners and the potential benefit of the tool providers. Further research is suggested, in particular looking into the selection process of reference clients and the measurements behind the stated benefits of case study companies.

Contents

1. Introduction

1.1 Problem Description

Implementing inapt technology may have dire consequences for the system development process, potentially leading to increased costs, decreased productivity, or in some cases even ending the process completely. As Pfleeger et al. (2000) write, “Even though the absolute cost of a particular technology might be small, the business risk in choosing the wrong innovation might be very high.” Failure is a very possible outcome when a purchasing decision is for example based on how the tool has worked for others (without knowing how their specific situation relates to one’s own) or on a list of features and benefits. Those in charge must make decisions based on how a tool will work for them in particular.

When in the process of acquiring a new SE tool for one’s organization, one will inevitably come across several claims regarding the various tools in question. These claims will state the benefits one will supposedly experience by implementing the tools. In order to uncover how a software engineering tool (SE) will affect software development in one’s own organization specifically, valid empirical evidence regarding the claims are necessary – not just to see if there is support for the claims, but also to identify whether or not the benefits are applicable to the situation of one’s own organization. As far as I have been able to determine through software engineering related sources, there has been no study into the amount and validity of empirical evidence available from the providers of SE tools. As an SE practitioner cannot be expected to manage to find all available such material (if any) from external sources, it ought to be possible to get the necessary information from the tool providers themselves.

1.2 Objective

1.2.1 Definitions

How much empirical evidence is there to support claims made by software engineering tool providers about the benefits of their tools, and how valid is said evidence?

- **Empirical evidence** – any data that is based on observation or experience and can function as support for the claims presented by the tool providers. Data in the form of metrics or data for which the acquirement is described in detail are in this context regarded especially meaningful.

- **Claims** – any statements made by the tool providers and which proclaim that the tool in question will affect a user in a specified positive manner, potentially in comparison with other tools.

- **Software engineering tool providers** – any creators of tools used in software engineering, whether open source or closed source, “free” or at a cost.

- **Benefits** – these may include the user experiencing anything from stronger performance to lower costs. In short, anything considered an improvement and resulting from the use of the tool.

- **Tools** – in general any type of tool that is designed for use in an SE environment. The method section explains in further detail.

- **Valid** – what constitutes as empirical, as evidence or as empirical evidence may vary based on the viewpoint. In the context of this research, evidence is considered valid to a certain degree if it fits the definition of empirical evidence given above and gives support to at least one benefit mentioned in at least one claim from the respective tool provider. Additionally, a key to strongly valid evidence is that the details surrounding the empirical study in question are thoroughly explained. That is to say, who conducted the study,

how the study was conducted, why the study was conducted, how the data was measured, and both what brought about the results and what they imply.

1.3 Area of Research and Relevance

Previous research has discussed that software engineering professionals, when faced with deciding which SE approach to adopt, have primarily based their beliefs on "anecdotes, gut feelings, expert opinions, and flawed research," as opposed to carefully designed experimentation (Fenton et al. 1994). That study looks into claims regarding SE techniques, standards and tools, and describes faulty experiments and contradicting results as to the efficacy of examples of these. Noted is also the lack of knowledge as to how to "establish and evaluate the design of experiments", due to this generally not being an emphasized part of SE curriculums. The authors conclude that "very little empirical evidence" exists to substantiate major improvements by adopting certain new technologies. It goes on to suggest that software managers should demand well-designed empirical research, including quantitative measurements, to back up claims regarding "new or changed practices". Furthermore, both they and software developers need to willingly participate in such studies, according to the authors.

While the study described above focuses largely on new technology, it also mentions changes in technology. Additionally, the study looked predominantly into external sources of experimental evidence. My research, on the other hand, will focus on the amount of evidence available from the tool providers themselves. As the study mentions that standards incorporate unsubstantiated methods and techniques, it follows that some poorly based standards may end up in the SE tools on offer. Thus it will be interesting to see what evidence exists to back up the efficacy claims surrounding the tools. Also, the study mentioned is from the mid-90s. One would hope changes in attitudes and procedures have occurred since, something my research may shed some light on.

The subject area of this thesis relates first and foremost to Evidence-based Software Engineering (EBSE), which in turn takes ideas from Evidence-based Medicine (EBM) (2005 Dybå et al.). EBM is about combining individual expertise with the "current best evidence" based on systematic research, with regards to individual patient care (1996 Sackett et al.). EBSE, then, aims to combine "current best evidence" and practical experience with regards to the decision-making processes involved in software engineering, according to Dybå et al. A short breakdown of the EBSE is as follows:

"

1. Converting a relevant problem or information need into an answerable question.
2. Searching the literature for the best available evidence to answer the question.
3. Critically appraising the evidence for its validity, impact, and applicability.
4. Integrating the appraised evidence with practical experience and the values and circumstances of the customer to make decisions about practice.
5. Evaluating performance and seeking ways to improve it.

"

By employing EBSE, the users discover which technologies are backed up by evidence of their efficacy, how valid that evidence is, and how the evidence and practical experience combined culminate in the best possible choice and use of technologies. Although EBSE appears logical, it appears it is not a process followed by many. One cause may be the difficulty in acquiring evidence due to the limited research within software engineering and the lack of scrutiny of the evidence that does exist. The question that thus appears is that of my thesis; how much evidence is there to support claims and how valid is it? Without the existence of the necessary evidence, EBSE will only take us so far. If my work reveals that little valid evidence exists, affecting step 2 and indirectly steps 3 and 4, an implication may be that the EBSE process will be difficult to adopt, which in turn may mean the scenario of companies investing potentially large sums in tools that fail to deliver continues.

The research will also discuss marketing factors to some extent, which may enlighten decision-makers as to the techniques used by tool providers to convince potential

customers that their tool is superior. By having some awareness as to what the producers do and how some marketing principles generally work, it may be easier to be more objective in the decision-making process. Of interest will be discussing the correlations between the empirical evidence availability (or lack thereof) and marketing strategies.

1.4 Structure

This thesis is split up into five chapters, including the introduction, and two appendixes. Chapter two explains the method used for the study. In chapter three, the results of the survey are described briefly, before each of the tools included in the survey are discussed individually. Finally the findings are summarized. Chapter four discusses the results over four main points, while chapter five draws conclusions and suggests further research. In appendix A there is an example of the information request e-mails sent. Appendix B includes all correspondence with the SE tool providers, except for the original information requests and the request reminders.

2. Method

To accommodate the main objectives of this thesis there were three main tasks to approach. Firstly, claims made by software engineering tools about the efficacy of their tools had to be identified. Secondly, information corroborating those claims was to be acquired. Finally, the acquired information would need to be critically appraised (as would the lack of information or the lack of response). However, before identifying claims it was necessary to devise a method for the sampling of products to research.

Ensuring a large degree of generalizability and statistical significance would require a random selection of a large sample of SE tools. If the intent was to come to a conclusion for all SE tools in general (which, ultimately, would be of major interest), the sampling would become far too widespread and complex for the scope of this thesis. One would have to take into account a range of variables, such as “size” of the tool (e.g. installed base or number of downloads/purchased licenses), type of tool, open versus closed source, and so on, when making generalizations. Therefore I decided to look at certain tool types in particular.

The approach selected was to send a very specific query[1] through the IEEE Software search engine, including terms such as IDE and tool in order to increase the chances of SE tools being mentioned in the results. The first 100 results were selected, in order of relevance (the alternative was date. It was decided that relevance was most appropriate to ensure a maximum of articles mentioning SE tools.) The results were meticulously scanned for names of software engineering tools (not simply names of companies). It was decided to focus on identifying certain popular types of tools, in order to increase the contemporary interest. These were integrated development environments (IDE), AJAX tools/toolkits, version control systems (VCS), software

[1] The query: (((programming tools)<in> metadata) <or> ((ide) <in> metadata) <or> ((development environment) <in> metadata) <or> ((editor) <in> metadata) <or> ((tool) <in> metadata) <or> ((tools) <in> metadata)) <and> (pyr >= 2003 <and> pyr <= 2008) <and> (52 <in> punumber)

configuration management systems (SCM) or other semi-related systems, and modelling (UML) tools. The first 18 tools (originally three sets of 5, then changed to add 3 more tools, simply to generate more results) encountered and falling into any of the above categories were selected. To make sure contemporary tools were identified, the search query was limited to include 2003 through early 2008. In all cases, the newest version of a tool was researched.

Once the tools had been selected, there were mainly two methods considered for identifying claims. One possibility was to scour print media (e.g. magazines, leaflets, information pamphlets); the other, chosen one was to inspect the product websites of the selected tools. If no claims fitting the given definition were found for a tool, a new tool would have to be selected, following the procedure above. Choosing to focus on the product websites was done primarily for two reasons: 1) To save time on acquiring material and thus allow more time for analysis; 2) To more easily allow for a similar study to be repeated at a later date, by me or someone else.

For acquiring information about a software engineering tool, several approaches were available. These included, but were not necessarily limited to questionnaires, information requests, and searching external databases. However, utilizing external sources would go far beyond the scope of this thesis. If searching one specific database, why not another specific one? The second one might contain empirical information on a tool for which no information was found in the first one. Additionally, as explained in the introduction, the interest of this thesis lies in discovering how much empirical evidence the SE tool providers themselves have, as they ought to be the primary source of independent, empirical data should a potential client wish to acquire such information, for instance as part of the EBSE or similar process.

With the above in mind, the chosen method of information acquisition was contacting the producing companies via information requests. Upon having identified claims for a tool, contact information was gathered to the best of my ability, and requests for empirical evidence were sent by e-mail (in some cases by filling out contact forms,

either in addition to or instead of by e-mail, depending on the availability of contact information). The requests were all very similar in structure, an example of which is available in Appendix A. Again, the reasoning behind choosing this mode of research was the potential for future repetition, for anyone, anywhere.

Relying only on responses from the tool providers would paint only a partial picture, in particular because there could be various reasons why an organization did not respond besides not wishing to. It was therefore deemed necessary for the purpose of the study to search the product websites for material that could potentially verify or somewhat support the claims. In many cases, information received in response to the requests included the same material as found on the websites. As explained above, external sources could not be used, which also implied that any external references given by the companies could not be used unless referring to a specific document or a web area containing specific documents.

As the response rate was very low to begin with, five new tools were researched, following the same procedure as for the first 18. The only difference was that it was not decided that these tools had to fall into a certain software category.

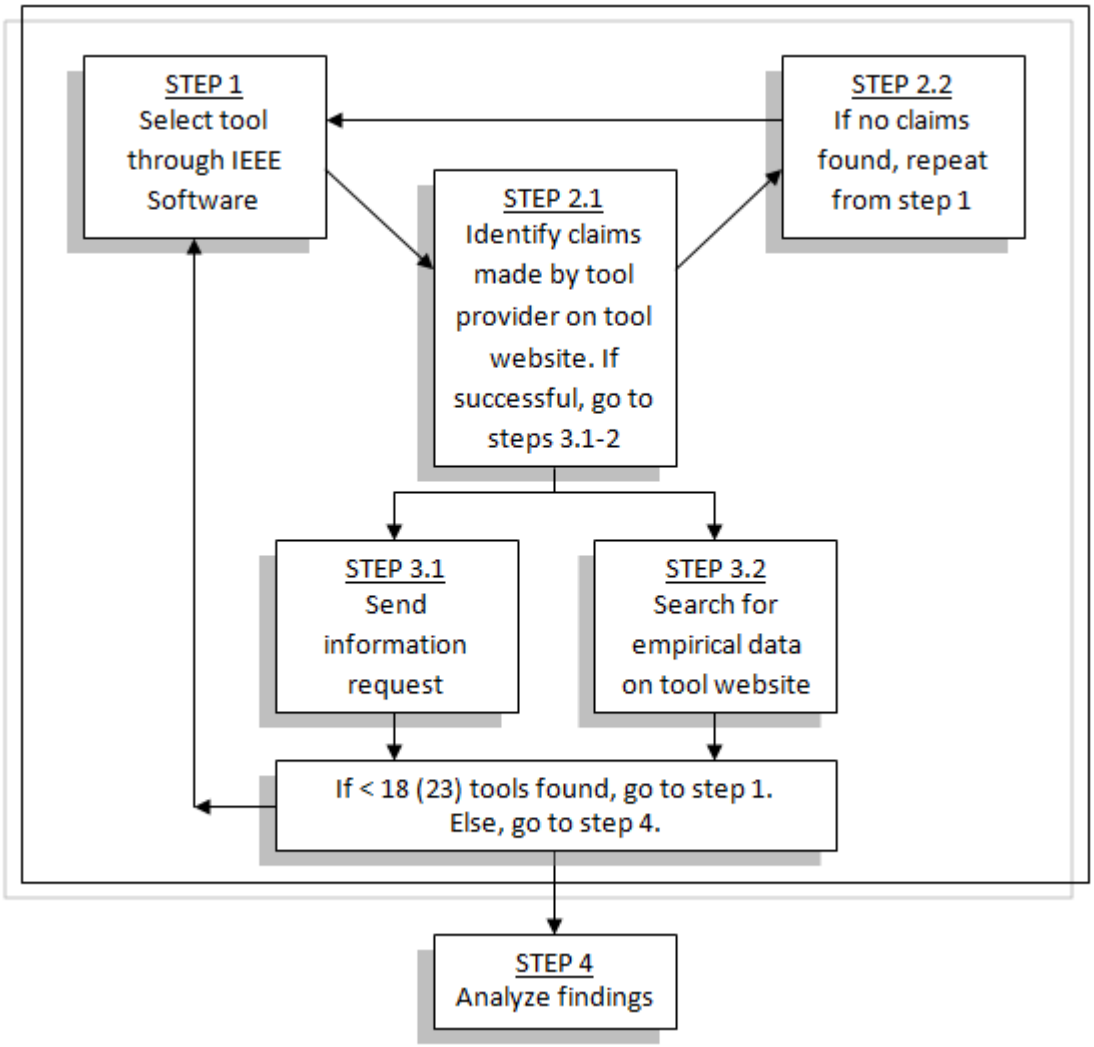
Finally, the material received and collected was analyzed and summarized.

While there is quasi-randomness to the selection of products, my own discretion is indeed also involved, rendering no statistical basis for the degree of which the products chosen represent all software engineering products, as Fowler (2002) writes. The results will, however, give a potentially strong indication as to the current status, and may spark further research into associated areas.

While I have strived to keep the analysis as close as possible in time to the gathering of data, there may of course have been published material on the associated websites while the analysis was undertaken. All claims and material gathered were up-to-date and read in March/April 2008.

Figure 2.1 below summarizes the research method in a flow chart.

Figure 2.1 – Flow Chart of Study Procedure



3. Research Results and Discussion

3.1 Results

There was initially a very low rate of response. To accommodate this, the companies that had not responded were sent a friendly reminder, while a new, smaller wave of five companies were contacted. Table 3.1.1 below shows which companies were contacted, which products they were contacted about, the type of product, and the address of the product or company website. Also shown is whether the company responded or not, regardless of the negativity or positivity of the response. Cases where only an auto-response was received, such as "out of office" or an automatic support response, were seen as non-responsive.

Table 3.1.1 – Overview of companies contacted, products researched and responses

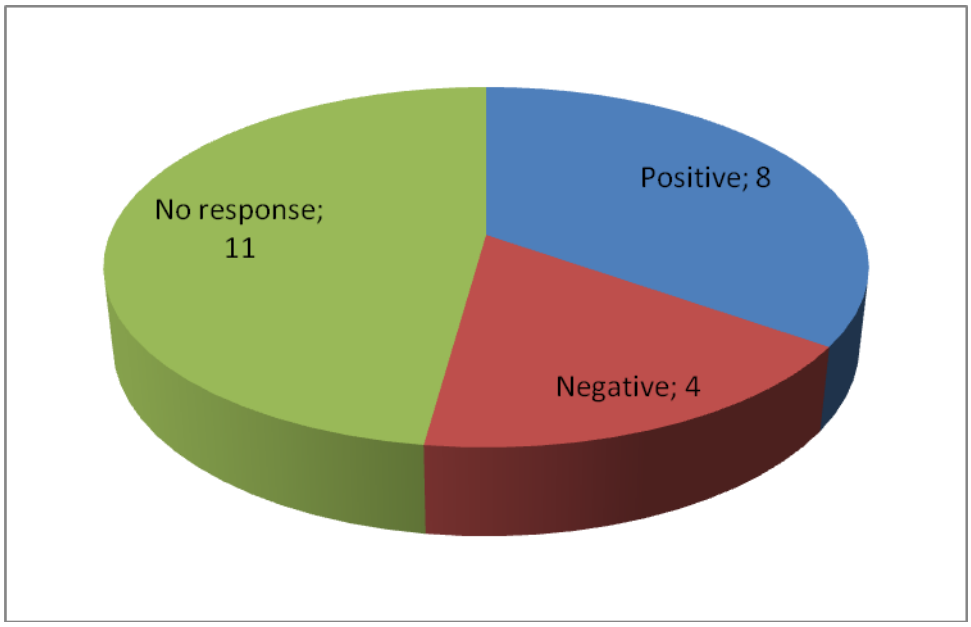
Company/ Organization	Tool	Type/ Genre	Response (Yes/No)	Website
Sun Microsystems	NetBeans IDE	IDE	Yes	www.netbeans.org
Eclipse Foundation	Eclipse IDE for Java Developers	IDE	No	www.eclipse.org
JetBrains	IntelliJ IDEA	IDE	Yes	www.jetbrains.com/idea
Microsoft	Visual Studio 2008	IDE	Yes	msdn2.microsoft.com/en-us/vs2008/default.aspx
Adobe Systems	Adobe Flex Builder 3	IDE	No	www.adobe.com/products/flex/features/flex_builder
Google	Google Web Toolkit	Ajax	Yes	code.google.com/webtoolkit
ThinWire	ThinWire	Ajax	No	www.thinwire.com
Backbase	Enterprise Ajax	Ajax	No	www.backbase.com/products/enterprise-ajax/overview
TIBCO Software	TIBCO General Interface	Ajax	Yes	www.tibco.com/software/rich_internet_application/general_interface/default.jsp
Dojo Foundation	Dojo Toolkit	Ajax	Yes	www.dojotoolkit.org

BitMover	BitKeeper	VCS/ SCM	No	www.bitkeeper.com
Microsoft	Visual SourceSafe	VCS/ SCM	Yes	msdn2.microsoft.com/en-us/vstudio/aa700907.aspx
Perforce Software	Perforce	VCS/ SCM	Yes	www.perforce.com/perforce/products.html
IBM	Rational ClearCase	VCS/ SCM	Yes	www-306.ibm.com/software/awdtools/clearcase
Telelogic	Telelogic Synergy	SDM	No	www.telelogic.com/products/synergy/index.cfm
Sparx Systems	Enterprise Architect	UML	Yes	www.sparxsystems.com/products/ea.html
No Magic	MagicDraw UML	UML	No	www.magicdraw.com
Tata Consultancy Services	MasterCraft	UML	No	www.tatamastercraft.com
SpringSource*	Spring Framework	Java frame-work	No	www.springframework.org
Coverity*	Coverity Prevent	Static code analysis	Yes	www.coverity.com
Borland Software*	Borland Together	Visual modeling	No	www.borland.com/us/products/together
CodeGear*	JBuilder	IDE	No	www.codegear.com/products/jbuilder
Gentleware*	Poseidon for UML	UML	Yes	www.gentleware.com

* These companies were the ones contacted at a later date to increase the potential amount of data. Due to the later date of initial contact, there was not time for a second round with those of these companies that did not respond.

A positive response is one where the company replies with either links to documents on their own site, links to documents on specific external sites, attached documents or similar. The validity of said documents is naturally not necessarily at the desired level and will be critically appraised later in this section. With a negative response I mean a response that declares no information is available or the company is unable to help. Simply referring to “sites such as X”, Google or similar is also seen as a negative response in this context. The pie chart in figure 3.1.1 below shows the spread of positive, negative and no responses.

Figure 3.1.1 – Responses to Information Requests



As can be seen from the above figure, just over half the organizations contacted responded. More precisely, 12 responses were received from 11 organizations (Microsoft were sent requests regarding two separate products and responded to both) out of 23 requests. Of those 12 responses, eight were positive while four were negative. The eight positive responses mainly consisted of links to or attached case studies.

In order to be able to discuss how much evidence exists to support claims it was, as mentioned in the Method section, necessary to actively roam, within reason, the respective product websites in a search for data. By within reason I mean to suggest that data to support the claims should, if at all existent, be readily available on the websites; it should not be necessary to spend a great deal of time and effort to ingeniously traverse the site. The reality of the matter may of course be different, thus the product websites may contain data not found by me. While the companies may not explicitly state that information found on the website will support any specific claims, the data potentially will do so anyway, and therefore ought to be looked into.

3.2 Claims and Validity of Acquired Data

This section presents the claims identified about the various products researched as well as discusses the validity of the data potentially supporting said claims, both data sent in response to the requests and data found through the previously discussed methods. Unless stated otherwise, all emphasis (bold formatting) in the claims has been added by me and was also added in the e-mail requests. The review is done alphabetically.

3.2.1 Adobe Flex Builder 3

Claims

1. "Adobe Flex Builder 3 software is a **highly productive** Eclipse™ based development tool […]."[2]
2. "Flex Builder 3 provides **the fastest way** to create applications for the Adobe AIR™ runtime […]."[2]
3. "However, **most developers will find it more productive to use Flex Builder 3** to design and create their applications."[3]

Discussion

The claims made by Adobe regarding Flex Builder 3 are good examples of the seemingly common claims about software engineering tools, namely increased productivity and increased speed. These are the types of claims that ought to be relatively simple to create metrics for, using, for instance, control groups. Case studies may also help support such claims, although without a control group several other factors might affect productivity and speed of development (e.g. improvement in coding skills over time).

[2] http://www.adobe.com/products/flex/features/flex_builder/
[3] http://www.adobe.com/products/flex/faq/

In the case of Adobe Flex Builder 3, no response was received, nor was any supporting material found via the product website. When it comes to the lack of response the reasons may, as in the subsequent non-responsive cases below, vary. It may be due to the request not being routed to the appropriate recipient in time or simply reflect a lack of belief in the importance of having empirical evidence to support one's claims. A third possibility is that they may have chosen to rely on the power of their company name, Adobe, as it indeed is one of the better known names in the software business. A similar thought may be behind claim #2: it would likely not be surprising for a potential customer that an Adobe development tool "provides the fastest way to create applications" for an Adobe-created runtime, even though no empirical data is presented to back up the claim.

Unlike several of the companies researched, Adobe's claims are not very bombastic. For instance, a claim stating a tool is "highly productive" is more open to a subjective point of view than the statement "is the most productive of all tools of type Z". Perhaps phrasing claims in that way gives rise to the notion that empirical evidence is less crucial. However, not comparing to other tools, neither specifically nor generally, suggests it should be easier to create empirical data to back up the claims. Instead of testing one's own tool and various rival tools, one can simply test one's own tool and present the metrics. Adobe quite possibly may have done such testing (including comparing to other tools, as indicated by claim #3), but no metrics were found on the website.

3.2.2 Backbase Enterprise Ajax

Claims

1. "For peace-of-mind you need Backbase Enterprise Ajax 4, **the only proven enterprise Ajax framework** with over 5 million runtime deployed."[4]
2. "[Backbase Enterprise Ajax] makes Ajax development **fast and easy**."[5]

[4] http://www.backbase.com/products/enterprise-ajax/why-a-framework/

3. "Backbase Enterprise Ajax includes of full set of development tools designed to **increase developers' productivity**."[5]
4. "Code reuse, agile development methodologies, integrated testing (including user interface), and application guidelines **speed up development** and promote maintainable code. This approach **lowers total cost of ownership (TCO)** for RIAs by eliminating the fragmentation and poor coding practices found in today's JavaScript development."[6]
5. "There are **no quicker ways** to build an Ajax application."[7]

Discussion

Backbase's claims regarding Enterprise Ajax (EAj) continue on from Adobe's Flex Builder claims in that they focus largely on productivity and speed. In addition, total cost of ownership (TCO) is introduced as an argument. They also make the rather bold (at least if no evidence is presented) claim that EAj is the "only proven enterprise Ajax framework", reasoning that with EAj having "over 5 million runtime deployed". With the statement that it is "proven", one would assume there is substantial data to back that up. Such data would be especially interesting considering the word "only" was added.

Claim #4 states that development will speed up while TCO will be lowered when using EAj. As reasoning, several features are mentioned together with the claims. While the reasoning may very well be both logical and true, as it stands the statements are at present still claims. As mentioned earlier, claims related to productivity can be relatively simple to test if the resources are available, even though there are varying opinions as to how productivity itself should be measured. The same goes for the concept mentioned in claim #2, namely ease. One could for instance have Ajax developers create applications using various (unfamiliar) Ajax development tools and survey their opinions as to how easy it was to create the applications for each tool.

[5] http://www.backbase.com/products/enterprise-ajax/overview/
[6] http://docs.backbase.com/docs/Backbase-Ajax-Evaluators-Guide.pdf
[7] http://www.backbase.com/products/enterprise-ajax/10-reasons-to-buy/

Just as in the previous tool's case, there was no response received from the producer, Backbase. In contrast, however, one potentially useful document, a whitepaper entitled "Ajax in the Enterprise", was found on the product website, as well as an excerpt from an InfoWorld review article, a case study from ABN AMRO MoneYou, and three short customer testimonials.

In the whitepaper, which discusses not only EAj, but the "Complete Enterprise Solution", Ajax 360, there is little empirical evidence, the paper mainly consisting of further claims. However, there is a table comparing scores for certain features (such as download size, security, performance, and ease of development) in open source Ajax, commercial Ajax, client-side Java, Flash, and Silverlight tools. The table shows that open source Ajax tools and client-side Java tools score lower than the rest when it comes to performance and ease of development, relating to claims #s 2, 3, and 4. The issue lies in that a reader of the whitepaper cannot know which specific tools are hidden behind the designations "commercial Ajax", "Flash", etc., nor can he know exactly what has been tested and how the testing was conducted. In the light of such issues, the table may be considered as indicative, but has little value as empirical evidence.

The review from InfoWorld (2006 Wayner) gives Backbase near top scores, with 9 out of 10 for ease of development and performance, and 8/10 for value. Interesting to note is that while the review examines four leading enterprise Ajax toolkits, the excerpt reprinted on Backbase's website has excluded the segments about the three other toolkits. Checking InfoWorld's own website one can find the original article, which shows that the three other toolkits received, respectively, the following scores: 8, 9 and 9; 8, 9 and 8; and 8, 9 and 9. In other words, the rival scores are on par with Backbase, diminishing the supportive value of the article as far as the claims "increase developers' productivity", "speed up development", and "lower [TCO]" go, while simultaneously giving some credit to claim #2 ("fast and easy").

In the review it is stated that all four toolkits "represent big leaps forward from the open source toolkits", thus reminding us that one's starting point plays a large role in

determining whether or not the comparing claims of Backbase and all other companies can be fully believed or not. Moving from the worst tool in class to an excellent tool will of course entail several improvements, but one may find moving from a third tool to the same excellent tool gives more mixed results.

The case study from ABN AMRO MoneYou describes the company's search for a tool to help them make the move from a regular HTML website to a "rich client-like application". MoneYou reportedly experienced a productivity increase of 200%, and the case study states that the framework "enabled [MoneYou's] developers to quickly and easily create [applications]". These statements do indeed support claims #s 2, 3 and, in part, 4. However, the usefulness of the report would likely vary depending on the situation of the potential new customer compared to the situation of MoneYou, such as the skill level and area of the developers. Additionally, while the case study paints a positive image, only one company's experiences are reflected. The support would be stronger if more case studies were available, portraying similar experiences.

One of the three rather short customer testimonials declares that EAj allowed the company, KPN, to "go to market faster". There is, however, too little information available through the testimonial for a new customer to learn more than that for one company, in one specific situation or other, EAj created one positive effect. Whether or not EAj allowed KPN to overall develop faster, easier, more productively, and less costly than what was previously the case is not mentioned, and understandably so, as it is simply a short testimonial. However understandable, though, it means the testimonial does not to any significant degree support the claims made by Backbase.

Overall it appears that there is some empirical evidence to support some of the claims regarding Backbase Enterprise Ajax, but that there simultaneously exists empirical evidence refuting some of the claims.

3.2.3 BitMover BitKeeper

Claims

1. "IMPROVE DEVELOPER PRODUCTIVITY"[8]
2. "IMPROVE WORKFLOW & QUALITY"[8]
3. "BitKeeper […] **enables higher software quality**."[8]
4. "BitKeeper's **total cost of ownership is one of the lowest** in the SCM market."[9]
5. "BitKeeper not only **makes developers more productive** […]."[9]
6. "BitKeeper's support level is also **unparalleled** […]."[9]
7. "BitKeeper has unique merging algorithms that **significantly reduce the chance of merge conflicts** when compared to other tools. […] Customers have reported as much as a **18 times reduction in merge time** using these tools."[10]

Discussion

As with Backbase's Enterprise Ajax, productivity and TCO are apparently seen as important marketing factors for BitKeeper. In addition, BitMover emphasizes an SCM specific feature, merging, as well as support. The support claim (#6) would be a relatively difficult one to empirically validate, due in part to the use of the rather ambiguous word "unparalleled". Firstly, what is (or was) measured? Quality of support? Speed of support? How service-minded support is? While speed of support is quite easy to measure, quality and service-mindedness are more complex; the latter is especially subjective. Secondly, how would it be (or was it) measured and how many tools would be/were included? For instance, customers could be asked to fill out a survey about the support system of their chosen VCS/SCM tool. Alternatively, BitMover could themselves test out the support systems of the most popular VCS/SCM tools over a certain period of time.

[8] http://www.bitkeeper.com/
[9] http://www.bitkeeper.com/Home.Managers.html
[10] http://www.bitkeeper.com/Products.Advantages.html

BitMover did not respond to any requests, but have provided via their website data potentially supporting their claims. In particular, the material consists of features comparisons with popular rival tools, customer comments, and two short documents; one explaining productivity, the other more general.

The feature comparisons generally consist of two parts. One part is a feature matrix listing features, whether or not BitKeeper or the rival tool in question includes the feature, and the benefit for BitKeeper users associated with each feature. Of course, this implies, as one might expect, that only features where BitKeeper benefits over the other tool are listed. Such a feature comparison matrix can be quite convincing. Armstrong (2008) writes: “When possible, focus on benefits rather than features.”; what BitMover is doing is focusing on both, in order to show why BitKeeper would be a better choice. However, while it should be simple to confirm that the listed features are indeed not included in the rival tools, this does not satisfy as empirical evidence of the stated benefits BitKeeper customers will get.

The second part of the feature comparisons employ a technique well-known amongst politicians, i.e., negative marketing. More precisely, lists are provided with comments generally in the form of “[Product X] has no [feature A]” – in some cases with a final positive comment at the end of the list. While such an approach may be both effective and factual, again no empirical evidence is presented as to the benefits received by BitKeeper users.

BitMover do not have case studies available via the BitKeeper site as far as I was able to discover. Instead they have customer comments and a grid of the logos of BitKeeper customers. While these include individuals such as Linus Torvalds (stating “I took a look at Subversion, and it doesn’t even come close to what I wanted.”[11]) and companies such as Intel, HP, Yahoo and Sony, showing that BitKeeper is used by high-ranking people and companies in the software engineering world, there is no way for a potential customer to know why the current customers chose BitKeeper

(e.g. productivity or TCO), what situation they were in when approaching BitKeeper (i.e., whether or not the situation is comparable to the potential new customer), nor what their experiences with the product have been (how much did they use it? Did they actually note increased productivity or quality, or have they begun looking at other tools?). A few of the comments found on the product website do include statements such as "We migrated from Teamware to BitKeeper and saved thousands of hours by doing so" (Bill Moore of 3pardata)[12]. That statement does indeed support to some extent the claim that BitKeeper will improve productivity, but still begs some of the same questions as above to be asked. In particular, 3pardata went from Teamware to BitKeeper. Unless the new customer has experience with using Teamware, the example may be meaningless in their decision-making process.

As mentioned earlier, there were two short documents available on the website. They included examples such as "BitMover was able to host hundreds of open source projects involving thousands of developers with only a small $2000 PC because the load was distributed across the developers' machines"[13], and "BitKeeper's customers have seen as much as a 2x gain in productivity within the first year."[14] The problem with such examples, given by BitMover themselves, is that while they are intended as empirical evidence, they can just as easily be construed as claims if no more detailed data is available, as in BitKeeper's case.

3.2.4 Borland Together

Claims

1. "**Increase productivity** and **quality** by automating design and code reviews that include audits and metrics at the model and code level"[15]

2. "**Boost productivity** through Model Driven Architecture® (MDA®) features […]"[15]

---

[11] http://www.bitkeeper.com/Comparisons.Subversion.html
[12] http://www.bitkeeper.com/Comparisons.Sun.html
[13] http://www.bitkeeper.com/pdf/BKDatasheet.pdf
[14] http://www.bitkeeper.com/pdf/BKAgileDevelopment.pdf
[15] http://www.borland.com/us/products/together/index.html

3. "Borland Together is a visual modeling platform that enables software teams to **consistently deliver on-time, high value applications** that meet business needs while **improving quality, cost and team communication**."[16]

Discussion

Borland continues the theme of productivity, quality and cost, but also adds in its claims the promise of on-time delivery of applications. With the claim that deliveries will be on time, the product may appeal in particular to companies currently experiencing delays and looking for ways to change that. To ensure that the product, Borland Together, truly does deliver on this claim, real world examples (i.e., case studies) may be deemed especially fit as empirical evidence. However, of importance when discussing case studies and their validity is how the company studied was selected, who wrote it, and so on, something which will be elaborated on in the Discussion chapter. As a brief note it can be mentioned that if, for instance, X number of case studies describe companies managing to deliver their products on time (as opposed to before) using a certain product made by A, but a number Y>X companies have experienced the opposite (or no change), the latter cases most likely will not be presented on A's website, for obvious reasons. By knowing how the case studies were selected in the first place, one can more correctly determine the likelihood of the above scenario.

In Borland's case, no response to my requests was received, but 26 case studies and a few customer profiles (case study summaries) were found via the product website[17]. While it is very positive to have a relatively large number of case studies available, implying a variety of comparable situations, challenges and requirements, there are three issues in particular. One is that most of the case studies involve at least one product in addition to Together, usually two extra products, as part of Borland's complete application lifecycle management (ALM) solution. With several products discussed at once it is difficult to extract exactly how and to what extent Together

[16] http://www.borland.com/resources/en/pdf/products/together/together_faq.pdf
[17] http://www.borland.com/us/customers/profiles/view-by-product.jsp

affected any stated improvements. A minority of the studies do specify that Together improved for instance productivity and time-to-market, though relatively often those gains were described as being a result of using Together in conjunction with other Borland products.

The second issue is that while there indeed are a few case studies which only discuss Borland Together, these are up to several years old, using older versions of the product. Presumably a product does not deteriorate over time, but rather improves as new iterations are published. In the spirit of EBSE though, presumption is not recommendable as basis when purchasing a SE product. Additionally, competing products will likely have changed over the years as well, for better or worse. The older case studies do on the other hand at the very least give an indication of which level the product was on earlier.

The final issue is the lack of metrics and elaboration of testing. Only a small number of companies describe having undergone an exhaustive evaluation of products before purchasing from Borland, just as only a few of the studies include numbers (e.g. a certain percentage increase in productivity). In the cases where numbers are present, there is generally little information about how those numbers were acquired, i.e., what kind of tests and comparisons were performed at the customer company following purchase.

There was one potentially interesting whitepaper, entitled “Successful Implementation of Model Driven Architecture”[18], available via the product website. The paper is in essence an elaborate case study on an anonymous financial institution. While it explains well what was done and some benefits reaped, such as reduced costs, again there is a lack of metrics. As the institution wished to remain anonymous, in itself potentially raising questions, it was, “For the purpose of this paper”, renamed “Prosperous Bank”. In other words, reading through the pages of the report, one repeatedly comes across the word “prosperous” as relating to a company using

Borland Together. While it may be considered humorous by some and perhaps will only be subconsciously noted by others, it may also have an adverse effect on a third group, possibly being seen as somewhat condescending towards the reader.

An upside to the case studies provided by Borland is that all the claimed benefits at the beginning of this section are indeed mentioned in the results. Not all customers describe having encountered all the claimed benefits, but each one is mentioned in at least one case study. Thus some empirical evidence appears to be available, but determining the validity is difficult, as long as the majority of the case studies either describe Together in collaboration with other Borland tools or revolve around older versions of Together.

3.2.5 CodeGear JBuilder

Claims

1. "Experience **unparalleled productivity and code reuse**"[19]
2. "And while Application Factories **will certainly yield higher levels productivity** during the development of an application […]"[19]
3. "ProjectAssist provides simple, single point installation and configuration of a complete developer tool stack […], **saving days or weeks** of manual configuration and integration."[19]
4. "JBuilder 2008 tools **simplify** and **improve** the development process […]."[20]
5. "**Unlike any other product available today**, JBuilder remains the **world's best commercial Java IDE** because it meets this unique blend of needs and allows customers to maintain their own software agenda not change to meet ours."[20]
6. "Introducing Application Factories - game changing technology that **dramatically reduces the complexity** of real world Java development."[21]

[18] http://www.borland.com/resources/en/pdf/products/together/together-successful-implementation-mda.pdf
[19] http://www.codegear.com/article/34448/images/34448/JBuilder2008_RTB_080228.pdf
[20] http://www.codegear.com/article/34448/images/34448/JBuilder2008_FAQ_080311.pdf
[21] http://www.codegear.com/products/jbuilder

Discussion

With snippets such as “unlike any other product available today”, “the world’s best commercial Java IDE”, and “dramatically reduces the complexity”, CodeGear’s claims are rather indicative of market-speak, somewhat more so than the other claims we have so far encountered. As such, one might not have high hopes for unearthing valid empirical evidence to back up the claims. The claim that JBuilder is the best commercial Java IDE in the world is particularly difficult both to prove and accept. One would have to compare all features of all available commercial Java IDEs, a task too vast for reason. Additionally, that which makes an IDE good for one developer may be insignificant for another. That latter point also to some degree applies to the less bombastic claims, such as the classic “increased productivity”. Developers differ in their approach to development, therefore features that make one developer more productive, might not make a second developer more productive – or worse yet, they might even make the second developer less productive.

CodeGear do have something to back up their “best IDE” claim in that InfoWorld selected JBuilder for the Best Java IDE of the Year 2008 award[22,23]. JBuilder, in competition with NetBeans and IBM Rational, gained the highest scores for features and integration, the same as its competitors for ease of use and performance, and the second-to-highest score for value. While JBuilder won overall, it was only competing against two other IDEs, which were selected because they had won the same award each their year the past two years. JBuilder was selected due to having won an InfoWorld IDE comparison in 2005. Therefore the selection method can hardly be seen as highly scientific and random, meaning there may well be other, untested Java IDEs which potentially could have performed better.

CodeGear was yet another company that neglected to respond to the information requests, making the product website the only source of data. Found there was a case

[22] http://www.codegear.com/article/37512
[23] http://www.infoworld.com/article/07/03/26/13FEjavaides_1.html

study on Monterey Bay Aquarium Research Institute (MBARI)[24], an InfoWorld "Java IDE Comparison Strategy Guide"[25] (which includes a version of the review discussed above), and the paper "A Comparative Study of Commercial Eclipse-based IDEs"[26] by Cost Xpert Group.

The MBARI case study describes how MBARI moved to an edition of JBuilder 2007 (mainly from Eclipse) for their development of applications for managing research information. In support of some of CodeGear's claims, the study notes that JBuilder was easy to use for MBARI's developers (due in large part to their developers' familiarity with Eclipse, the framework upon which JBuilder is built), as well as helped increase their productivity. What is not mentioned are any metrics, e.g. on productivity. In addition, we do not know how the case company was selected, something which may be of particular interest considering only one case study was found.

As mentioned, the "Java IDE Comparison Strategy Guide" includes a version of the review already discussed. It also contains some useful empirical data; however it is taken from the Cost Xpert Group's report. Thus I will discuss that instead. The Cost Xpert paper's intention was "to objectively measure the benefits of using" three commercial IDEs based on Eclipse, compared with the freely available, baseline Eclipse IDE. The study used four scenarios: new, large Java projects, new, small Java projects, and enhancing/maintaining large or small Java applications. The results showed JBuilder gave the biggest (and statistically significant) savings in all cases, either alone or together with one of the two other commercial tools. These results were based on a single project at a time. The study was repeated for three imaginary companies of various sizes, developing multiple projects simultaneously. All three commercial IDEs were significantly more efficient than the baseline Eclipse, with JBuilder again giving the highest savings. As the cost savings are explained to be due,

[24] http://www.codegear.com/article/37784/images/37784/MBARI-case-study.pdf
[25] http://www.codegear.com/article/34209/images/34209/CodeGear%20Final.pdf
[26] http://www.codegear.com/article/34209/images/34209/Java-Productivity.pdf

in part, to increased productivity and quality, the report gives support to claim #2 and perhaps also somewhat to claims #s 4 and 6.

There are, naturally, limitations to the study. Firstly, only three Eclipse-based IDEs are tested. While all three are popular IDEs, there exist several other popular Eclipse-based IDEs. Secondly, as the report only sets out to measure Eclipse-based IDEs, there is little to gain for potential JBuilder customers interested in a comparison with IDEs not necessarily based on Eclipse. For its purpose, though, the report provides some valid empirical support for some of CodeGear's claims.

3.2.6 Coverity Prevent

Claims

1. "Prevent **is the industry standard** because **only Coverity understands the strict requirements** of static source code analysis."[27]
2. "Prevent identifies more information about source code than any other product on the market today […]."[27]
3. "[…] **easy-to-use** tools that allow defects to be addressed **within minutes**."[27]
4. "Prevent for Java's technical breakthroughs result in **unmatched improvements to the quality** of critical Java code"[28]
5. "Out of the box, Coverity Prevent has **an average false positive rate of less than 20%**."[29]
6. "Coverity Prevent seamlessly integrates with your existing environment and **can be deployed and configured within hours**. **Other tools can take weeks and even months to set up and configure**, costing you precious time and resources."[29]

[27] http://www.coverity.com/html/prod_prevent.html
[28] http://www.coverity.com/html/prod_map_dna_java.html
[29] http://www.coverity.com/html/prod_benefits_c.html

Discussion

Coverity sent a positive response to my information request, commenting briefly on the claims and linking to two documents. In addition there were case studies available via the product website. Commenting on the first claim, it was admitted in the response that it is "partly market-speak" (2008 Chou), although it was at the same time claimed that Coverity "pioneered many practical techniques" used in static source code analysis. It may be correct, fully or partly, but with no independent source given, the claim still stands as just that – a claim.

When it comes to claim #4, a type of claim that is very difficult, if even possible, to prove, Coverity responds that they have internal measurements showing Prevent performing "better than open source tools such as Findbugs". The immediate implication that then comes to mind is that perhaps they do *not* have internal measurements showing that Prevent performs better than tools not falling in the "open source" category. Additionally, it is stated in the response that the data is not available to the public, yet again making it difficult to consider the validity of the statement. Reasons for the data being publicly unavailable can only be speculated, and may range from the less positive ones, such as the measurements not showing significant advantages for Prevent, to the more understandable ones, such as the measurements including sensitive company data. For claim #5 it was also stated that publicly unavailable internal data supports the claim, as well as customer feedback and trials.

Eight case studies were made available[30], presenting fairly well-defined issues, making it easier for potential customers to find out if the situations are comparable to their own. In addition, the studies discuss only one product, Coverity Prevent. Most of the case studies mention that the company in question has been a customer since a certain year. One would expect a long-term customer to be so due to being satisfied with the product. On the other hand, how likely is it that the long-term customer has throughout the years evaluated and re-evaluated competing products to ensure that

[30] http://www.coverity.com/html/library.php#casestudies

their choice is still the best choice? Evaluation periods are expensive procedures and thus most probably not repeated once a product has been chosen, as long as the satisfaction is present. Having satisfied customers counts as a positive, of course, but it does not entail that there are no better products available at present.

All the case studies combined seem to support the claims that Prevent increases productivity, is easy and fast to deploy and integrate, has a low rate of false positives (in one case 5%), and improves quality (even finding bugs in code believed to be bug-free). Furthermore, as is not the case for some of the other products researched, several of the case studies describe extensive evaluation of tools before selecting Prevent. On the other hand, besides a few customers stating low rates of false positives, detail is seldom given as to how big the improvements have been.

All case studies, for Prevent and other products, tend to include a bit of market-speak. When identified, such market-speak can sometimes cast a somewhat negative shadow over the case study. For Prevent, the AudioCodes case study[31] states that "Software quality has always been a primary goal for AudioCodes' team of over 200 developers." The implication is of course that AudioCodes is focused on software quality and they use Coverity Prevent, so if you are focused on software quality, you should use Coverity Prevent as well. However, it would be surprising if there exists any company dealing with software engineering *not* being focused on software quality, and not all SE companies use Prevent. As mentioned earlier, the validity of case studies comes in part down to the selection process and who has written/conducted the study. The quotation above may diminish the credibility of the case study for some.

The two documents previously mentioned, linked to in the Coverity response, are entitled "A Comparative Study of Industrial Static Analysis Tool (Extended Version" (2008 Emanuelsson and Nilsson) and "Analysis Tool Evaluation: Coverity Prevent" (2006 Almossawi, Lim, Sinha), the former being a study conducted by a Linköping

University professor and Ericsson, a customer of Coverity, the latter a Carnegie Mellon University student report..

The study compares Prevent and two other static analysis tools, but notes the fact that there are several other competing tools and “the list of competitors is steadily increasing”. The study supports the claims that Prevent is easy to install, finds defects not otherwise found and analyzes large code bases in a relatively short time, based on experiences at Ericsson. Ease-of-use has not been studied. Also confirmed was Prevent’s claimed low rate of false positives, however, an interesting point was made in that respect; while returning few false positives, Prevent neglected to find some defects known to be present in analyzed code and found fewer defects overall than some other tools. While the severity of the neglected defects can be discussed and probably vary, it is a good example to remind decision-makers to not only consider what is being claimed, but also what is not being claimed.

The student report set a complete version of Prevent up to be tested on a varied selection of real projects. According to the report, “Coverity claims that Prevent’s average false positive rate is around 20%”, with the results being consistent with that. The claim I identified regarding false positives states that the average rate is *less* than 20%, which is not immediately supported by the study, as it estimates the false positive rate to be “somewhere between 12.7% and 35.7%”. Claim #3 is supported, as Prevent is described as relatively easy to use, and able to analyze large amounts of code in minutes.

During my research, the ever popular cost claim was not identified for Prevent. This may be another example of the importance of noting what is not claimed, as the student analysis describes Prevent’s “high monetary cost” as “Perhaps the biggest limitation”. Another limitation mentioned is that Prevent, when tested by these students, only worked with C/C++ code. At present, Coverity also supports Java. Thus the question is whether or not the positive results obtained in the study are true

[31] http://www.coverity.com/library/pdf/coverity_case_study_audiocode.pdf

for Java as well. Additionally, the report being two years old, other factors may have changed too, such as the cost and the ease-of-use.

Coverity definitely has some proper empirical evidence to back up some of its claims, but there are indeed limitations, as well as indications of less positive aspects to the tool.

3.2.7 Dojo Toolkit

Claims

1. "Dojo allows you to **easily** build dynamic capabilities into web pages […]."[32]
2. "You can use the components that Dojo provides to make your web sites **more usable, responsive, and functional**."[32]
3. "The result is a small, tight toolkit **that is blazing fast**. Dojo's performance alone makes it an ideal platform to extend and build on."[33]
4. "Dojo is **used on high-profile, high-traffic sites every day** and Dojo's build tools are a key reason why."[33]
5. "That's why the Dojo Core gives you **one of the most highly acclaimed** Ajax interfaces around."[34]

Discussion

Looking at the claims surrounding the Dojo toolkit, one notices that unlike many other product claims, no comparisons are made against generic competing tools, apart from to some degree in claim #5. This coincides with a remark made in the response received, namely that "comparisons are often difficult, since the feature sets can differ so much across toolkits" (2008 Peller). It appears then that Dojo have decided to focus on telling what they see as especially good about their product, rather than what it potentially does better than others.

---

[32] http://dojotoolkit.org/about
[33] http://dojotoolkit.org/book/dojo-book-0-9/introduction/why-dojo
[34] http://dojotoolkit.org/projects/core

With the above in mind, one might expect Dojo to have an easier job in acquiring material supporting their claims. However, as was stated in the response, they have no case studies or similar due to a lack of budget and resources. It was suggested that data could be found by perusing specific and non-specific external resources, though that, as mentioned earlier in this thesis, would take my research beyond a manageable scope, since doing it for one tool would require doing it for all tools.

Interestingly, while suggesting looking at external sources, it was also mentioned that "frankly there just aren't enough objective, accurate and unbiased studies." Although one might be inclined to agree, the question that then arises, if external sources are generally subjective, inaccurate and biased, and internally there are no resources for performing own studies, is there really any data existent at all to back up the claims, or are they based entirely on market-speak and/or personal impressions from usage?

Claim #4 is perhaps the simplest one to give examples of. The response mentions both IBM and AOL using Dojo, and the site has a "spotlight" section[35] showcasing a few websites using or having used Dojo, including why they chose Dojo. In essence the spotlight section consists of a few miniature case studies with short statements.

As empirical research is stifled by a lack of budget and they, according to the response, "don't have very much time to spend on PR", Dojo sees the claims as exercises for the user, by employing the demos and tests packaged with the toolkit. Several of the tool producers researched have demos available and, for the tools that come at a price, trial versions for download, in order to let the user find out for his- or herself if the tool lives up to its claims. A positive opportunity as that surely is, empirical support for claims would help a potential user decide which tools to concentrate testing on, as there rarely are resources available to test all possible tools.

[35] http://www.dojotoolkit.org/spotlight

3.2.8 Eclipse IDE for Java Developers

Claims

1. "**Considered by many to be the best Java development tool available**, the Eclipse IDE for Java Developers **provides superior Java editing** with validation, incremental compilation, cross-referencing, code assist; an XML Editor; Mylyn; and much more."[36]

Discussion

The first noticeable feature in Eclipse's case was the near complete lack of claims. This may be due either to a belief that purely factual information "sells" better, or perhaps the fact that Eclipse has become a much larger operation than simply a Java IDE, potentially making it either difficult or unnecessary to focus claims on a single component of Eclipse. Whatever the case, a claim was indeed found eventually, a rather strong one at that.

Eclipse did not respond to the requests nor was there any data found on the product website. There were several links to other sites where some information apparently should be available. However, finding pertinent information proved difficult. In fact, some links were even non-functional. Such a complete lack of easily accessible empirical information may indicate that Eclipse simply does not consider it at all necessary. That in turn may have various reasons. Perhaps Eclipse has experienced no significant demand for empirical experiments data and that other elements, such as e.g. feature lists, are of higher importance. It may be that Eclipse believes their status in the SE environment and the widespread use of their framework as evidence enough (even other tools researched for this thesis, such as JBuilder, are based on Eclipse). Without a response it is difficult to deduce the thought process – all we do know is that empirical evidence, as defined in this thesis, is lacking.

[36] http://www.eclipse.org/downloads/moreinfo/java.php

3.2.9 Gentleware Poseidon for UML

Claims

1. "Nearly every UML tool producer claims to have an "intuitive interface", but **only Poseidon for UML incorporates inherently simple functionality**."[37]
2. "**Amplify quality** while fostering a **shorter time-to-market**."[38]
3. "**Constrain costs and development hours**."[38]
4. "**Lower resistance** to new tools with the **easy and intuitive interface**."[39]

Discussion

Gentleware's claims return to the trend that we had seen before Eclipse and Dojo, namely quality, costs, productivity, time-to-market, and ease-of-use. Included is also the relatively standard claim stating that "only our product [does or has] something", in this case "inherently simple functionality", which, as elaborated on in previous cases, is difficult to provide evidence for. The other claims though, should be possible to back up.

The response from Gentleware included two whitepapers found on their website, as well as a few links to pages on their website. Also available on the website was a list of customers[40] and a few customer quotations[41]. Again as described earlier, customer lists, especially when including well-known names, as in Gentleware's case, add some credibility to a product, but does not include information as to the level of satisfaction, the choices for selecting the tool, which other tools were considered, and so on. Similarly, the short quotations include no data, simply opinions and claims. For instance, one customer writes "Poseidon already provides a price to feature ratio that is unmatched in the market."[42] Even though it is a customer making the statement, it

[37] http://www.gentleware.com/products.html
[38] http://www.gentleware.com/fileadmin/media/pdfs/products/poseidon_datasheet_se.pdf
[39] http://www.gentleware.com/fileadmin/media/pdfs/whitepapers/Introduction_to_Poseidon_for_UML-Gentleware_Whitepaper.pdf
[40] http://www.gentleware.com/103.html
[41] http://www.gentleware.com/102.html
[42] http://www.gentleware.com/customers.html

cannot be leveraged any more than claim #3 (which allures to the same idea) without further empirical evidence.

The two whitepapers[43] were “Selecting a UML tool” and “Introduction to Poseidon for UML”. The former explains tests (e.g. usage) to run on tools under consideration when in the process of selecting a UML tool, noting that the reader should keep in mind that the paper was “written by a tool vendor”. Whether intended as a preemptive strategy or not, it must indeed be kept in mind and lessens the validity of the paper. However, in the context of this thesis, neither paper appears especially valid, as they describe features and associated benefits, not including empirical data. Admittedly the selection whitepaper describes tests run on Poseidon (and three competing tools), but the tests are generally of the form “can you do this or that”, as opposed to more scientific tests producing metrics. One area where the introduction whitepaper may be useful is with regards to claim #4 (“easy and intuitive interface”) as the report contains screenshots (as does the website) with explanations. It is also, as with other products, possible to try an evaluation copy of the tool.

To sum up the above, Gentleware lacks proper empirical data to back up their claims, accentuated by a note in their response: “We do not provide any scientific researches” (2008 Derevenskykh). The information that is available still simply consists of statements. Stating that feature A uses technology B and C to automate D, thus increasing E (e.g. productivity) by F%, which is good, may be as logical and as acceptable as they come, but without actual proof that E increases, by F% at that, through rigorous testing of the feature in various situations and against other tools or older versions of the same tool (due to the “increasing” claim), the statement will continue to stand as a claim.

[43] http://www.gentleware.com/553.html

3.2.10 Google Web Toolkit (GWT)

Claims

1. "GWT applications are **almost always as fast** as hand-written JavaScript."[44]
2. "Google Web Toolkit (GWT) is an open source Java software development framework that **makes writing AJAX applications easy**."[45]
3. "Writing dynamic web applications today is **a tedious and error-prone process**; **you spend 90% of your time working around subtle incompatibilities** between web browsers and platforms, and JavaScript's lack of modularity makes sharing, testing, and reusing AJAX components **difficult and fragile**. **GWT lets you avoid many of these headaches** while offering your users the same dynamic, standards-compliant experience."[46]

Discussion

Google describe GWT as able to produce fast applications in an easy way. In addition, Google point out benefits they claim GWT will get you as a user by first describing the negative situation one will find oneself in if one does not use GWT.

The response from Google was a helpful one, referring to pages already found on the GWT website, as well as a few external, customer websites for testimonials. It was stated that the best approach would possibly be to contact developers who had gone through the decision-making process and chosen GWT, to hear what they had to say. While in truth a possible (and hopefully not uncommon) approach for potential new users of GWT or other applications, it was, as explained earlier, not within the scope of this thesis. However, some of the contact information included a couple of useful links directly to case studies/testimonials.

The testimonials[47,48] as well as a tech review[49] available via the GWT website largely agree with the claims that making applications with GWT is fast and relatively easy.

---

[44] http://code.google.com/webtoolkit/overview.html
[45] http://code.google.com/support/bin/answer.py?answer=54830&topic=10208
[46] http://code.google.com/webtoolkit/
[47] http://googlewebtoolkit.blogspot.com/2007/10/lombardi-blueprint-built-with-gwt.html
[48] http://www.queplix.com/solutions/google-gwt-technology/
[49] http://devblog.glowday.com/2007/05/tech-review-google-web-toolkit.html

Supporting the claims as they do, they do not contain much detailed information, especially the testimonials, other than "we experienced this" (which in itself is merely indicative), nor are they plentiful (as in only three references in total).

As mentioned above, the response from Google also included links to areas of the GWT website, more precisely areas where one could find presentations, articles (where the tech review above was found) and books. A large amount of information is available through these pages, but the majority is simply descriptive (with, for instance, code examples). A few of the presentations did contain a graph (the same graph in each presentation) aimed at showing how much faster a GWT-produced Ajax application (both first run and subsequent runs) is compared to regular HTML. Additionally a few bar charts are included in some of the presentations, showing how the (at the time of writing) latest version of GWT, 1.4, has improved on 1.0 as far as performance of the applications goes. While these graphs and charts also support the notion of GWT applications being fast, they give little, if any, evidence with regards to the claims above.

3.2.11 IBM Rational ClearCase

Claims

1. "An **industry-leading solution** that provides sophisticated version control, workspace management, parallel development support and build auditing to **improve productivity**."[50]
2. "Deliver high-quality code with fewer bugs through secure version management and reliable builds".[51]
3. "Rational ClearCase can help you **get more done in a shorter period of time**."[51]
4. "Rational ClearCase software provides extensive support for parallel development, enabling developers to […] **more easily resolve conflicts** and **reduce confusion**."[51]

[50] http://www-306.ibm.com/software/awdtools/clearcase/
[51] ftp://ftp.software.ibm.com/software/rational/web/datasheets/clearcase.pdf

5. "[…] Rational ClearCase software helps to **prevent mistakes**, **reduce bugs** and **identify errors earlier** in the delivery cycle to **resolve them more quickly**."[51]

Discussion

Combining the fact that IBM's claims are of the same type as many already encountered above and seemingly not overly difficult to measure with IBM's massive size gives reason to believe that IBM performs scientific testing of their products to one degree or another. Whether they do or not, no information to support that hypothesis was found, nor was any other potentially useful empirical information.

IBM did respond, but alas, no information came from that. One response referred to another point of contact which was subsequently contacted, but did not respond. Another response explained that IBM receives a massive amount of student requests and therefore cannot respond to such requests. Links to areas of the website where information might be available were given, but nothing of use was found there.

IBM being unable to help with regards to the request due to the amount of student contact they receive both seems highly likely and comes across as fair enough. However, a thought that then comes to mind is that IBM and perhaps other companies might have responded quite differently if requests similar to mine were sent from a potential customer. One can only speculate, of course, but it seems a rather fair assumption as long as a for-profit organization is involved.

Having found no useful information and having contemplated the likelihood of IBM performing scientific tests to gather empirical data, one wonders why the information appears to be lacking from the (albeit very extensive) website. It may come down to the power of company and brand names. IBM, being the huge company that it is, has a very recognizable company name in the computer technology world. Armstrong (2008) describes recognizable brands in the modern, impersonal era akin to how people long ago tended to only trade "with local shopkeepers who they knew and trusted". Armstrong also states that "when new claims are made, they are more convincing if made by a credible firm – so emphasize the brand and company." As

can be seen above, three of the five identified claims begin with the brand name, Rational ClearCase. It is therefore conceivable that IBM does not regard empirical data as important enough in this context, relying instead on the power of their company and brand names.

3.2.12 JetBrains IntelliJ IDEA

Claims

1. "IntelliJ IDEA has been **consistently called the best Java IDE in the industry**."[52]
2. "**Greatly improved performance**, combined with **way better usability** is a key highlight of ver.7."[52]
3. "In version 7 you will find **even better debugger** […]."[52]
4. "Starting up is **plain faster** (we mean it, you'll notice). […] IntelliJ IDEA now fully leverages multi-core processor capabilities to bring you **significant improvements in performance**. […] Thus, **it feels naturally faster**. […] **You will notice this as soon as you try the new version, even on slower machines.**"[53]
5. "With all this, your work is going to get **easier, faster**, and a lot more enjoyable!"[53, 54]
6. "Our IDE has **always** made the work of developers **easier and more efficient**."[53]

Discussion

JetBrains seem to focus largely on performance and speed, as well as ease of use. All of these, the former two in particular, can be tested relatively simply. Conversely, though, according to the response from JetBrains, they have not done so.

The first response from JetBrains stated simply that they had no such information as I requested. Upon asking for a clarification as to whether that meant they truly had no such information or that they had it, but were not willing to send it, the response was that no studies or researched was performed regarding the claims, stating that some of

[52] http://www.jetbrains.com/idea/
[53] http://www.jetbrains.com/idea/buy/top_reasons.html
[54] "all this" being features for "web service oriented developers".

the claims were obvious and easily observable while others could "be googled for" (2008 Baranov). While it is true that it for instance can be found that IDEA has won some awards as best Java IDE (as has at least one of the previous cases, JBuilder) by using Google and while it is possibly quite "obvious" that fully leveraging the capabilities of multi-core processors will "bring you significant improvements in performance", neither are satisfying as evidence for the claims. As mentioned when discussing Gentleware Poseidon for UML, no matter how logical and acceptable a claim is it is still just a claim until empirical measurements can support its validity.

Besides the negative response from JetBrains, no significant empirical information was found on the website either. Considering the last sentence in the second response, this does not come as much of a surprise: "I don't even know why such studies would be needed." It is certainly both acceptable and expected that there are differing views regarding the necessity of empirical research when it comes to software engineering. JetBrains' response merely underlines the idea that the reality of the matter is that other factors are either just as or even more important than empirical evidence when it comes to purchasing SE tools.

3.2.13 Microsoft Visual SourceSafe

Claims

1. "**Increase Developer Productivity**"[55]
2. "**Improve Software Quality** Through Effective Source Code Control"[55]
3. "Developers can use Visual SourceSafe **quickly and effectively without extensive training or maintenance**."[55]
4. "Visual SourceSafe 6.0c is the **easiest, most productive** source code management and version control system for development teams using Microsoft Visual Studio .NET."[56]
5. "[…] and **application performance scales with hardware**."[56]

[55] http://msdn2.microsoft.com/en-us/vstudio/aa700907.aspx
[56] http://msdn2.microsoft.com/en-us/vstudio/aa700906.aspx

Discussion

The claims identified regarding Visual SourceSafe (VSS) include the common productivity, quality and ease-of-use statements, as well as scalability and non-extensive learning curves – all allowing for measurable metrics of one kind or another.

Microsoft was contacted regarding two products, VSS and Visual Studio 2008, and responded in a positive manner to both, stating that they would gladly help in acquiring data. Material regarding Visual Studio 2008 (consisting mostly of case studies) was received later, but information about VSS would take a while longer. In the end, data for VSS was not received in time. The material received for Visual Studio 2008 will be discussed separately below. Even though there are two different products, it is not entirely unlikely that had there been time enough to receive data regarding VSS it would be similar in empirical validity and strength to that received for Visual Studio 2008, though naturally, that remains an assumption.

Microsoft has a large amount of information available about their products on the respective websites, but as far as empirical data goes for VSS, nothing was identified, bar case studies[57]. These do not specifically support the claims in much length, although for example increased productivity and lower TCO are mentioned. Where the studies do appear rather useful, however, is in explaining in detail the situation that needed resolving and, unlike most other case studies encountered in my research, the actual results for the companies from Microsoft's solutions. That is to say, instead of simply, say, improved quality and lower cost (if that was the case), the studies elaborate on the effects improved quality and lower cost had on the business.

Alas, a downside to the VSS case studies is the same that brought down the helpfulness of the Borland Together case studies: all the VSS studies encountered describe a vast range of tools and services in a complete Microsoft solution, with very little detail as to whether or not any specific elements of the solution had a certain

effect. Therefore it is difficult to determine the value of the case studies when considering to what degree they support the VSS claims.

As a side note it is worth mentioning that the response from Microsoft explained that VSS is in much less use now, as Visual Studio Team System, discussed below, has taken over in many cases, and is thus focused on more.

3.2.14 Microsoft Visual Studio 2008

Claims

1. "Developers of all levels – from hobbyists to enterprise development teams – now have **a consistent, secure and reliable solution** for developing applications for the latest platforms […]."[58]

2. "Visual Studio-branded tools continually deliver **better ways for software developers to do more with less energy wasted** on repetition and drudgery."[59]

3. "Visual Studio is engineered and **tested to be consistently dependable, secure, interoperable, and compatible**."[59]

4. "Use Visual Studio development solutions to give your development team powerful ways to:
   Increase productivity and quality through integrated and familiar tools.
   Reduce costs through better visibility of your development process."[59]

5. "With Visual Studio 2008, organizations will find it **easier than ever before to capture and analyze information** to help them make effective business decisions."[60]

6. "**Ensure application correctness more easily** with integrated unit testing."[61]

Discussion

Although noted as an IDE in the table at the beginning of section 4, Visual Studio 2008 (VS08) comprises both Visual Studio (VS) and Visual Studio Team System

[57] http://www.microsoft.com/casestudies/search.aspx?ProTaxID=1293
[58] http://msdn2.microsoft.com/en-gb/vstudio/default.aspx
[59] http://msdn2.microsoft.com/en-gb/vstudio/products/bb931214.aspx
[60] http://msdn2.microsoft.com/en-us/vstudio/products/bb931331.aspx
[61] http://msdn2.microsoft.com/en-gb/vstudio/products/bb931328.aspx

(VSTS), the latter of which is Visual Studio with several added features, largely to do with working in teams, as the name indicates. The claims above may refer to both VS and VSTS, or either one. Covered in the claims are more or less all types of previously discussed claims; productivity, cost, quality, communication, and reliability. Microsoft goes even further with VS08 by adding claims about consistency, security, compatibility, and more. Finally, the claims are promising when it comes to hopes of finding empirical data, as claim #3 states specifically that VS has been tested.

As mentioned when discussing Microsoft VSS, a positive response was received and quite a bit of material given. Most of the material, as in so many other cases, consisted of case studies. Such data was also found on the website, but as with VSS, the ones on the website to a high degree discuss several products/services at once and will therefore be disregarded in favour of the ones received from Microsoft, which mostly discuss VS/VSTS only and also include ROI (return on investment) case studies.

Claim #3 mentions that VS has been “engineered and tested” for, among other things, compatibility. One case study received tells how the Dutch company Achmea used Visual Studio (albeit 2005, not 2008) to “boost productivity”. Interestingly, the following text is included: “Although a number of suppliers offered tools with [the desired] capabilities, few of them were able to deliver the crucial element: integration with the existing platform. [New paragraph] Achmea has a successful track record using Microsoft® solutions, and it is company policy to consider Microsoft solutions before others.” Further in the text it is stated that the platform used is Microsoft .NET. That may make a reader wonder; were the other tools unable to deliver integration because the Microsoft-based platform is difficult for others to integrate with or because they were inherently bad at integration? If the former case is true, one might begin to question the compatibility claim of VS as well. Furthermore, using a Microsoft platform combined with a company policy of looking at Microsoft products first may make a reader question the objectivity of particularly selection, but also of subsequent testing, of alternative products.

Looking at the case studies in general they largely corroborate several of the claims made. While the aforementioned concerns regarding case studies remain, such as company selection, Microsoft has a few extra positives, especially in the form of rather elaborate studies (generally more detailed and lengthier than other case studies looked at), with much explanation as to situation and requirements. Additionally, there was somewhat more use of measured numbers. However, on the less positive side, several of the case studies regarded Visual Studio 2005, not 2008. As discussed previously, one would generally imagine new versions of tools are released due to them being improvements, but it would still be preferable to view opinions on the current product, the one that is under consideration by a potential customer.

The ROI case studies, three in number, were prepared by Nucleus Research, a global "provider of investigative information technology research and advisory services."[62] As such one can expect their research to be independent and objective. The three ROI case studies focus, naturally, on ROI, but also explain the direct and indirect benefits that lead to the ROI. Additionally, the studies include a large amount of detail, particularly regarding the numbers used to calculate the ROI. These three case studies give valid support to the claims involving improved productivity, quality and communication, as well as reduced costs. However, it must be added that the VSTS version in question is VSTS 2005.

A Microsoft IT Showcase paper was included in the received material, giving details on deployment of an early version of VSTS 2005 within Microsoft. The results presented show that VSTS scaled well and maintained a 99.9 percent uptime. While keeping the fact that this was in-house deployment and an older version of VSTS than the claims above relate to, the results indicate that there may be something to the claims regarding dependability, reliability, and consistency.

[62] http://nucleusresearch.com/about/

3.2.15 NetBeans IDE

Claims

1. "NetBeans provides you with […] a **more productive** work environment than other IDEs."[63]

2. "The NetBeans IDE **can boost your productivity** when you're working with Java SE, Java EE, or Java ME technology as well as Ruby, Ruby on Rails, JavaScript, and C/C++."[64]

3. "Extending the platform and its Swing-based foundation **saves development time and can optimize performance**."[65]

Discussion

All three claims from NetBeans revolve around the same key point: developer productivity. A reason for that may be that NetBeans considers productivity should be the main focus for an IDE, or perhaps they believe potential users will consider for themselves any direct or indirect benefits increased productivity may reap, such as potentially reduced costs.

The response from NetBeans stated that no data was available regarding the identified claims, but nevertheless included links to usability tests reports[66] and statistics[67]. The statistics incorporate information that may be of use to some decision-makers in determining productivity, provided they have similar data from other tools to compare with, or have other means of analyzing the data. For instance, one statistic shows time to failure, while another shows usage of the code completion feature (NetBeans IDE correctly guessing desired code to be entered, alternatively invoked by the user himself). These statistics would likely be even more representative if all NetBeans

[63] http://www.netbeans.org/switch/index.html
[64] http://www.netbeans.org/switch/why.html
[65] http://www.netbeans.org/switch/why.html
[66] http://ui.netbeans.org
[67] http://statistics.netbeans.org/analytics

users were included (a module needs to be installed for the logging to happen), but still provide some valuable information as is.

Two of the usability tests were relevant to productivity, one testing editor usability[68], and the other debugger usability[69]. The studies both included eight participants, with information about these such as years of experience with Java programming and current IDE used. Besides a comparison with an editor usability test performed two years earlier on a previous version of NetBeans IDE, showing improvements, the results were presented as a list of findings and recommendations as to how to deal with the findings. The positive side to this is that the tests show explicitly that NetBeans consider usability seriously and contemplate ways in which to improve it. Of course, the information does not help in determining whether or not NetBeans IDE will boost one's productivity, but it does indicate a focus on the issue – presuming the recommendations given are taken into account when developing each new iteration of the IDE.

3.2.16 No Magic MagicDraw UML

Claims

1. "It provides the **industry's best code engineering mechanism** […]."[70]
2. "With MagicDraw you can complete your tasks with **half the steps** demanded by other tools."[70]
3. "Creates diagrams faster **than any tool on the market**".[70]
4. "The **learning is straightforward** and **learning period is short**."[71]

Discussion

No Magic was another company that did not respond, bar an e-mail auto-response. The product website was also of limited help, containing little empirical information.

[68] http://ui.netbeans.org/usability/editor_2/report.html
[69] http://ui.netbeans.org/usability/debugger_Sep_06/report.html
[70] http://www.magicdraw.com/ → *What is MagicDraw*

What were found were a list of customers, the value of which has been discussed earlier, and a rather long list of customer testimonials[72]. The validity of customer testimonials, with generally little indication of how the customers have come to their conclusions, has also been elaborated on previously. In this case, however, a running theme among the testimonials is that user interface is intuitive and the tool easy to use, making it easy for new users to get started, thus lending some support to claim #4.

The testimonials date as far as about a decade back. How one chooses to see the value of them in light of that fact may vary. On one hand, it shows customer satisfaction ranging over ten years. On the other hand, one cannot base a purchasing decision on statements regarding iterations of the program released years and years ago.

The lack of available data may in large part be due to three of the four claims, #s 1 through 3, in particular 1 and 3, being rather grand in their wording. Claiming "industry's best" and "faster than any tool on the market" puts them in the "quite difficult to prove" category. Thus, with the lack of empirical data in mind, one feels inclined to believe that market-speak more than anything else lies behind the claims.

3.2.17 Perforce

Claims

1. "Perforce gets **its high performance** from an intelligent implementation on top of a strong model […]."[73]

2. "[…]Perforce **responds so quickly** that developers never doubt using it. Simply put, Perforce **never makes users wait**."[74]

3. "[...] Perforce is so fast that **developers often use it for activities they once did manually**."[74]

[71] http://www.magicdraw.com/ → *Key Benefits*
[72] http://www.magicdraw.com/ → *Testimonials*
[73] http://www.perforce.com/pitch/page01.html
[74] http://www.perforce.com/pitch/page02.html

4. "A high-power SCM system, Perforce is still **easy to learn**. […]Perforce users can get up to speed in **less than a day**."[74]
5. "And **when the requests get big** -- hundreds of thousands of files big -- **Perforce really shines**."[75]
6. "One of the big barriers to adopting a new SCM system is the high cost, both in the up-front purchase as well as the ongoing maintenance and administration. **In each category and overall, Perforce is unbeatable**."[76]

Discussion

Perforce went furthest in responding to the claims, which focus on performance, learning curve and cost, by preparing a document with references (mainly to case studies) and personal commentary (2008 Tyler) for each claim.

With regards to claim #2 the response states that performance has been a central focus for Perforce, more so than for rival tools, referring specifically to a Zoran case study[77] and a Google presentation[78] from 2007 for support. The presentation describes in detail the large Perforce environment at Google, indicating high performance, as well as explaining ways in which performance can be improved (or has been improved by Perforce). The Zoran case study does indeed state that better performance than their previous solution was a factor in choosing Perforce, and that they saw improved performance as a result of deployment, as well as increased productivity and lower costs, lending some support both to claim #2 and other claims.

The commentary regarding claim #3 explains that Perforce considers certain functionality, such as merging, an integral part of the functionality, as opposed to other configuration management (CM) systems (according to the response). By automating such tasks otherwise done manually (again according to the response), Perforce will perform faster. While indeed making logical sense, one would have to investigate other tools to empirically testify to the above. However, it would not be

[75] http://www.perforce.com/pitch/page04.html
[76] http://www.perforce.com/pitch/page06.html
[77] http://www.perforce.com/perforce/success/zoran.html
[78] http://www.perforce.com/perforce/conferences/us/2007/index.html

unreasonable to expect that users of other tools and considering Perforce would already have an empirical basis to use for comparison. In addition to merging, deployment to runtime environments is mentioned as an extra feature not regularly involved in CM tools, and is supported by a Monster presentation from 2003[79].

With claim #4, Perforce, according to the commentary given, means to say that one day is generally enough to learn what is needed "to perform day-to-day parts of their jobs". Some of the case studies provided state that ease-of-use and that Perforce was easy to learn were key reasons for their selection when comparing with other tools. While indicating support for the claim, no information found mentions that one day or less was enough, although the commentary states that personal experience as a Perforce trainer has shown that only one or two days are generally required to get a group started with the tool.

Low (or lower) cost (claim #6) is mentioned by several case studies as a benefit from adopting Perforce. In many cases, reduced administrative overhead is partly the cause. Additionally, many of the case studies substantiate claim #5 by detailing the number of files (up to several million) and the repository sizes Perforce is used for. Perforce also performs benchmark tests which, if similar data were available for competing tools, can give some good ideas as to how the tools compare. While customer comments/testimonials are also available, they are, as in all other cases, far less detailed than the case studies.

In general, the case studies support one or more of the claims to some extent, and include quite a bit of useful detail, although somewhat less so when it comes to elaborating on the tool selection process. At the same time, some of the claims include typical market-speak that is difficult, if at all possible, to prove empirically.

Perforce provide five individual comparisons between their tool and competing tools via the website (also mentioned in the response). While partly focusing on explaining

[79] http://www.perforce.com/perforce/conferences/us/2003/tyler/tyler.pdf

features and associated effects, the majority of these comparisons do in fact contain quite a bit of empirical metrics, especially regarding performance and to some extent cost. The tests performed are explained in detail, including test environment, procedure, data, and results, with Perforce performing best in most cases. While the tests have been performed by the vendor themselves and they appear to be a couple of years old, the results are indeed empirical and support some of the claims made. That, together with the extensive response received, indicates that Perforce has a certain amount of appreciation for being able to empirically backup claims.

3.2.18 Sparx Systems Enterprise Architect

Claims

1. "With a great feature set and **unsurpassed value for money**, EA can outfit your whole team […] for **a fraction of the cost** of some competing products."[80]
2. "Enterprise Architect supports this process in **an easy to use, fast and flexible environment**."[80]
3. "**Intuitive and simple to use**".[80]
4. "One PIM can be used to generate and synchronize multiple PSM's - **providing a significant productivity boost**."[80]

Discussion

Besides the first claim, which in strong terms describes cost and value for money as key benefits, Sparx Systems makes fairly straightforward statements about Enterprise Architect (EA), focusing on ease of use, performance and flexibility. Sparx Systems sent a positive response, including links to case studies[81] and independent analyst reports[82], as well as attaching a list of EA industry citations. Some of these citations were from case studies or implementation reports, while others were brief mentions of EA being used in a project or other, stating for instance that cost and performance

[80] http://www.sparxsystems.com/products/ea.html
[81] http://www.sparxsystems.com/press/#CaseStudies
[82] http://www.sparxsystems.com/press/#AnalystReports

were integral when choosing EA[83] (though with no further information as to having, for example, considered other tools). Testimonials were present on the website, but the value of such has been discussed earlier.

Of the five analyst reports, three of them required purchasing and subsequently had to be disregarded. They would very possibly be informative (from an empirical viewpoint) for potential customers, however. The remaining two reports[84,85], both "Sponsored by Sparx Systems", do not support the claims, as they are simply analyses, as the term "analyst report" implies. They do not contain results from research on usage and comparisons with other tools. Both do, on the other hand, state that EA is in the low end as far as price goes.

A commonality of the case studies and EA implementation reports was that they were lengthy and describe in detail the situation, the implementation and the benefits, though as in most cases the studies rarely include metrics or explanations of how, when applicable, the benefits were measured. As detailed as the studies are, it becomes easier for potential customers to make a verdict as to whether or not the situation is comparable to their own and if they can expect the same benefits. In most cases it is also declared that extensive evaluation was performed before choosing EA, although, as in previous cases, without going into much detail regarding that procedure. These studies and reports seem to corroborate to a relatively large degree the strong (if not "unsurpassed") value for money and the fractional cost. Ease-of-use and intuitiveness is also repeatedly stated as an observed benefit, while performance improvements are mentioned, but somewhat less so.

An additional note on EA is that Sparx Systems are currently conducting a survey, according to the response, the results as of yet not available. What is being surveyed was not mentioned.

[83] http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1524939
[84] http://www.sparxsystems.com/press/articles/pdf/IDC%20Vendor%20Spotlight%20-%20Sparx%20Systems.pdf
[85] http://www.sparxsystems.com/press/articles/pdf/VDC%20Report%20-%20Sparx%20Systems.pdf

3.2.19 SpringSource Spring Framework

Claims

1. "Spring delivers significant benefits for many projects, **increasing development productivity and runtime performance** while **improving test coverage and application quality**."[86]

Discussion

While only one sentence was found for the Spring Framework, it incorporated several claims, most notably productivity and quality – again testable claims. SpringSource, which was one of the five companies contacted in the second wave, did not respond to the request sent. While having papers and articles available explaining how Spring works and how to get the most out of it, no empirical evidence was found.

3.2.20 Tata MasterCraft

Claims

1. "**50% cut** in development times and costs"[87]
2. "processes **simplified**"[87]
3. "**enormous flexibility**"[87]
4. "[MasterCraft Enterprise] enables software developers to rapidly build medium to large-scale applications that are **robust, flexible and scalable**."[88]
5. "This revolutionary suite of software tools helps software developers **drastically reduce (by as much as a half) the time taken** to develop new software solutions or to modify existing ones."[88]

Discussion

Tata has some of the same general statements as most others; decreased cost, and decreased development time (i.e., increased productivity). Other claims are also made,

[86] http://www.springframework.org/
[87] http://www.tatamastercraft.com/
[88] http://www.tatamastercraft.com/mcmastercraft.htm

which at first sight can be seen as difficult to verify due to varying opinions as to what, for instance, constitutes a “robust, flexible” application. One would, however, be inclined to think that when a company makes such claims, they have defined the terms involved, hence making testing and conclusions possible. In particular, the impression that testing *has* been done is made by claim #1, which states a specific percentage of decreased costs and development times. Only through actual monitoring and measuring can such a specific claim be made.

There appears to be a low level of activity around MasterCraft at present (the support pages note that support is closed on certain days – in 2006), which might explain the lack of response from Tata. I have been unable to determine whether or not the tool is in fact still available (whether as MasterCraft or incorporated into another tool). Some data was available, however, in the form of five case studies[89,90]. Quite a bit of detail as to time spent on deployment, lines of code produced, and so on, are presented, though with no comparisons (albeit allowing to some degree for readers to compare with own experiences). How much stock can be put into these case studies is questionable though, as at least one of them is over half a decade old (the organization studied discontinued operations in 2002).

Whether MasterCraft is still available or not, it is worth including in this research as the websites are still available and the reasons for Tata not responding are as unknown as in all other cases lacking a response. It must also be kept in mind that due to the research method, MasterCraft must necessarily have been mentioned in an article in 2007 or early 2008.

[89] http://tata-mastercraft.tcs.com/casestudy.html
[90] http://www.tatamastercraft.com/mcgstpa.htm

3.2.21 Telelogic Synergy

Claims

1. "Telelogic Synergy […] helps **development teams work faster and easier** by increasing communication and collaboration."[91]

2. "Synergy **accelerates** the release management and build management processes, **maximizes the efficiency** of limited development resources, and unites distributed development teams."[91]

3. "Telelogic Synergy […] **improves software development** by increasing communication and collaboration."[92]

4. "Telelogic Synergy **improves** build management by **reducing build times** and **enhancing build quality**."[93]

5. "Telelogic Synergy **increases developer productivity** with an easy-to-use, task-based to-do list. The Synergy for Developers interface **reduces a developer's management overhead**."[94]

Discussion

Telelogic Synergy was yet another product for which the claims revolve around performance, quality and productivity, which yet again gave rise to hopes that empirical evidence would be available. In addition, communication is mentioned as a positive factor. While no response was received from Telelogic, bar an auto-response, case studies were found on the product website, as well as one potentially useful report. Getting access to any whitepapers, case studies, and similar via the Telelogic website requires registration for a free so-called Telelogic passport, demanding a large amount of information from the user. While the severity of such a hindrance is likely to be subjective, it is generally a good idea to make the information gathering process as easy as possible for potential customers.

[91] http://www.telelogic.com/products/synergy/index.cfm
[92] http://www.telelogic.com/products/synergy/overview.cfm
[93] http://www.telelogic.com/products/synergy/reduced-build-time-for-build-managers.cfm
[94] http://www.telelogic.com/products/synergy/reduced-overhead-and-productivity-tools-for-developers.cfm

The case studies[95], while relatively short, consist of the fairly standard situation-solution-benefits scenarios. The studies do mention performance, quality and productivity as benefits gained, however, certain factors reduce their validity. These include a lack of metrics, most case studies including at least one other Telelogic tool besides Synergy, the old age of some of the studies, and a lack of comparison with other products (though some of the studies indicated the customer companies having checked out several tools before selecting Telelogic's, albeit without much detail about the selection processes).

The analyst report found on the website, entitled "Software Development Management"[96] and published in 2007, was written by the independent research company Yphise, who conduct their research in order to help IT executives make the right decisions[97]. The paper includes Synergy and four competing products which are assessed based on a range of Yphise criteria, with the aggregate results compared. It describes in some detail how the five tools were selected, including giving a lengthy explanation of various disciplines similar to, but not the same as Software Development Management (SDM), with examples of tools falling into those categories rather than SDM. Additionally, some SDM tools not included in the shortlist are mentioned, in some cases with short explanations as to why they missed the cut.

Synergy scores overall best of the five products in the Yphise report, as it also did the three previous times Yphise conducted a similar study – in 2000, 2002 and 2004. In particular, the report supports the claims regarding communication and productivity. Despite the lack of detailed metrics in the report, it is quite valid in supporting some of Telelogic's claims, as the study has been performed by an independent IT research organization and explains well the criteria involved and the results. What is less clear, however, is the manner in which the tool is assessed; that is to say, whether Yphise

[95] http://www.telelogic.com/customers/success-stories.cfm
[96] http://www.telelogic.com/download/index.cfm?id=5113
[97] http://yphise.com/Private/Pres/PresYphiseUS.asp

has, for instance, studied the features of the various tools, deployed and observed the tools in various real-life situations, or something else.

3.2.22 ThinWire

Claims

1. "[…] **the framework excels** with its highly interactive and rich user interface components."[98]
2. "Use ThinWire to […] and you'll be able to **provide an unparalleled user experience**, while at the same time completing your project **faster than ever**."[98]
3. "There is now […] a **much more solid and stable** framework (**dare I say the most stable of all the Ajax RIA's**) […]."[99]

Discussion

ThinWire has on its website a few examples of usage of the framework, as well as one user comment on the front page. The comment speaks highly of the *potential* of ThinWire, and when reading the blog post[100] it is taken from (for which the link given on ThinWire's website is, incidentally, incorrect), it becomes apparent that the comment was written before actually having tried out the framework, thus giving no support to the claims. Moreover, the blog post was written in December 2006.

ThinWire gave no response to the information request. In light of the tendency to market-speak in the claims, such as "unparalleled user experience" and the bracketed segment of claim #3, a lack of response and a lack of empirical evidence on the website were somewhat expected, although the lack of response may surely have various other reasons as well.

[98] http://www.thinwire.com/
[99] http://www.thinwire.com/blog/2007/09/17/after-countless-fixes-and-tweaks-thinwire-v12-rc2-is-here-final-is-near/
[100] http://evolutionarygoo.com/blog/?p=54

3.2.23 TIBCO General Interface

Claims

1. "Now you can **quickly create** and **easily deploy** full-screen rich client functionality […]."[101]

2. "By using TIBCO General Interface, you can **gain competitive advantage** while **driving down the cost** of solution development and deployment."[101]

3. "We've further optimized rendering for IE6 browsers so that **it's at near parity with today's faster browsers**."[102]

Discussion

Even though TIBCO's claims include the standard productivity and cost related statements, TIBCO went further than most in implying that little empirical evidence exists to support the claims. The first response received stated that the most apt person to talk to had left, while the second response declared a suspicion "that there was nothing scientific about the claims anyway" (2008 Peachey). Despite this, the website was of course researched on level with all the other product websites, finding that TIBCO won the InfoWorld Best AJAX Toolkit of the Year Award 2006[103], as well as four customer success stories[104].

The value of awards has been touched upon earlier, but an interesting note can be made in General Interface's case. Winning the 2006 InfoWorld award included getting a higher score in 2005 than Backbase's AJAX solution[105]. However, when discussing Backbase Enterprise Ajax, another InfoWorld AJAX toolkit review was included. That review gave Backbase's tool (and indeed a third tool) a higher score than General Interface and was published in 2006. Although an award based on somewhat limited reviews cannot constitute valid empirical evidence, the above further diminishes the current value of the 2006 award.

---

[101] http://www.tibco.com/software/rich_internet_application/default.jsp
[102] http://www.tibco.com/devnet/gi/default.jsp
[103] http://www.infoworld.com/article/06/01/02/01FEtoyawards_2.html
[104] http://www.tibco.com/software/rich_internet_application/general_interface/default.jsp
[105] http://www.infoworld.com/article/05/08/08/32TCback_1.html

Only one of the customer success stories was a traditional case study; two others were articles, each one describing a company employing AJAX using General Interface, and the third a press release, all with a varying degree of details of the specific benefits received from the tool itself. The case study[106] describes increased productivity and reduced costs as benefits gained, but besides the general case study issues, the validity of this one is lessened by the fact that there is only one and there is no information given to indicate that the case company looked into other tools when selecting General Interface.

[106] http://www.tibco.com/resources/customers/successstory_iconix.pdf

3.3 Summary of Findings

Table 3.3.1 – Summary of Data Analyzed and Validity

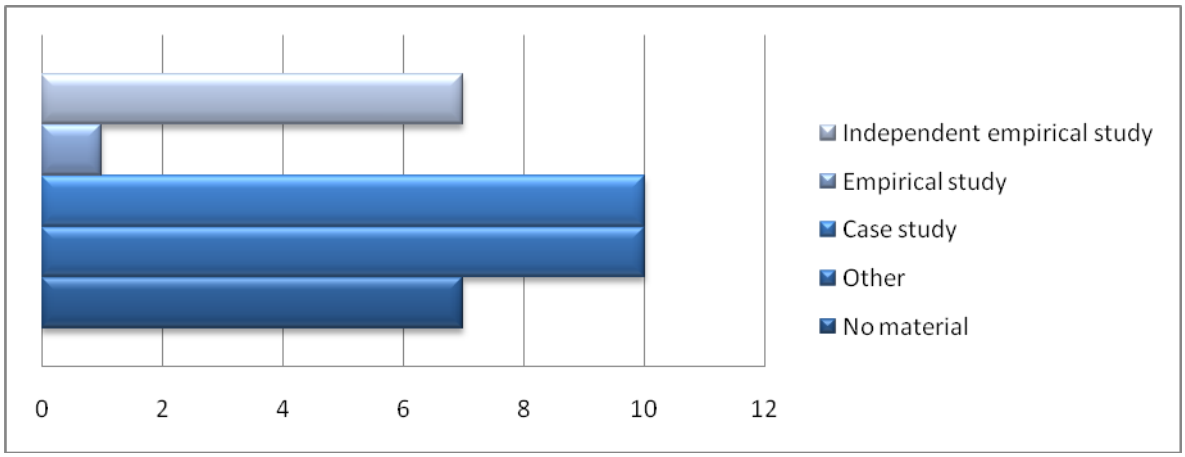
Tool	Case studies	Empirical studies	Ind. emp. studies	Other*	No material	Validity
Adobe Flex Builder 3					1	0
Backbase Enterprise Ajax	1			2		0
BitMover BitKeeper				9		0
Borland Together	26					1
CodeGearJBuilder	1		1	2		2
Coverity Prevent	8		2			2
Dojo Toolkit					1	0
Eclipse IDE for Java Developers					1	0
Gentleware Poseidon for UML				3		0
Google Web Toolkit				4		0
IBM Rational ClearCase					1	0
JetBrains IntelliJ IDEA						0
Microsoft Visual SourceSafe	10					1
Microsoft Visual Studio 2008	14		3	1		2
NetBeans IDE		2		1		0
No Magic MagicDraw UML					1	0
Perforce	16			5		2
Sparx Systems Enterprise Architect	5		2			2
SpringSource Spring Framework					1	0
Tata MasterCraft	5					0
Telelogic Synergy	7		1	1		1
ThinWire					1	0
TIBCO General Interface	1			2		0

* “Other” may include for example white papers, feature comparisons, etc.
Notes: These numbers indicate the empirical (or claimed empirical) material included in the data review above. Companies may have had more material, but which did not (claim to) contain empirical evidence. Also, in some instances there may have existed more of a certain type of material, but the review was limited to the numbers given above. For example, if there were a vast number of case studies, but they spoke of several tools at once or went far back in time, only the most apt ones would be used. Customer testimonials were not seen as being able to support claims, due to the very low level of detailed information, and are therefore not included in the table.

Table 3.3.1 above summarizes the material found or received and which was intended to support the claims with some level of empiricism. As can be seen from figure 3.3.1, case studies were the most popular method for empirical convincing, as for 11 out of the 23 tools, case studies were available (not including customer testimonials, ref. the

note below the table). In fact, the 10 (Microsoft appears twice) organizations had among them at least 94 case studies, with one tool, Borland Together, accounting for more than a quarter of that. Figure 3.3.2 further down sums up the validity column.

Figure 3.3.1 – Bar Chart Showing Number of Companies Providing Material Type

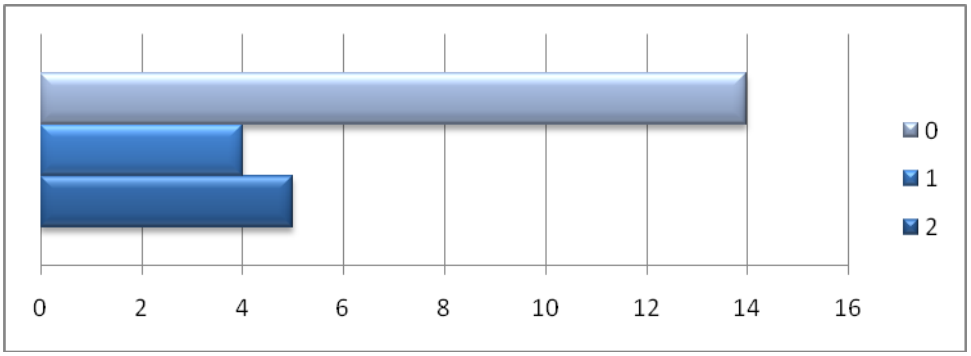


The bar chart above shows that “Other” was the second most popular category, including anything from white papers to feature comparisons. By and large, this data was not considered to empirically support the claims made. More surprisingly, perhaps, independent empirical studies come in a shared third place with “No material”. At first glance this might be seen as a relatively positive result, but the review of the material came to the conclusion that not all these studies gave a large degree of support to the claims in question. In particular, the InfoWorld reviews referred to by TIBCO and Backbase fall into the independent empirical study category. However, as was discovered, in Backbase’s case it was found that while the tool, Enterprise Ajax, performed overall best in the review, the difference was minute and the rival tools were in fact better in some categories. In TIBCO’s case, the tool came out on top in its review, but has been outperformed by other tools since. This goes to show that even when there are studies that are both independent and based on measurable observations, the result may require further analysis before being accepted as evidence.

Somewhat surprisingly, for only seven tools (approximately 30%) was there no material that could potentially give empirically valuable support to the claims made. The reasons considered during the review for why some tool providers appear to make no empirical material available have included a lack of resources (Dojo), brand recognizability (IBM, Adobe), and marketing principles (Eclipse).

The validity column is a crude manner in which to compare the value or validity of the provided and discovered material, and as such cannot be considered a definite interpretation of the results. It is important to note that the scale is not an objective measure, but my overall conclusion based on the material at hand in each of the 23 cases. It should be viewed as giving an indication of the number of tools for which the data provided some valid empirical evidence (2), little or less valid empirical evidence (1), or no valid empirical evidence (0). No companies could fully support all their claims with completely valid empirical evidence. More specifically, a score of 2 was given to those tools for which a portion of the claims appeared to be supported in part by independent sources (e.g. Coverity Prevent's two studies) or by very well defined and detailed non-independent empirical material, such as in the case of Perforce's five tool comparisons. However, even the cases scoring a 2 did not seem able to back up all their claims, and not all the independent sources were acceptable as supporting evidence (as was explained above for Backbase and TIBCO, who were given a 1 and a 0 respectively). A 1 indicates that either a very small portion of the material was considered valid or that the material in general was considered to have many flaws, but gave positive indications with regard to a minority of the claims. Figure 3.3.2 below shows the spread of the scores among the 23 products.

Figure 3.3.2 – Bar Chart of Indicative Validity Score



As the chart shows, nearly two thirds (61%) of the tools were considered to have no valid empirical evidence to back up the related claims, while little such evidence was available for four (17%) of the tools. Only five of the tools, or 22%, had what was evaluated to be some empirical evidence to back up some claims. No companies had enough valid empirical data to confirm all claims made regarding their tool. With only just over a fifth of the tools' claims being somewhat corroborated by valid evidence, the outlook for EBSE and similar procedures seems gloomy, the consequences of which will be discussed in chapter 4.

Of the 11 companies that gave no response to the information requests, only one was given a rating of 2 (CodeGear JBuilder), while three received a 1 rating (Backbase Enterprise Ajax, Borland Together, and Telelogic Synergy), and the remaining seven a 0. The responding companies were spread out in a similar manner, with an equal amount receiving a rating of 0, one receiving a 1, and four receiving a 2. Hence, the main difference between the responding and the non-responding organizations was that the responding ones dominate the 2 rating, the non-responding dominate the 1 rating. Two conclusions may be deduced from that. On the one hand, one might expect the amount of valid data available to have proved larger if more companies had responded. Alternatively, the implication might be that yes, some proper empirical material is available, but not easily accessible via the product websites. Rather, such corroborating information is made available upon request. In fact, this seems to be supported to a degree in this case, as the most valid (independent)

empirical evidence for three of the four companies scoring a 2 was sent to me and not found on the websites.

3.4 Limitations

There are of course certain limitations to this research, one in particular similar to one of the main issues that have been mentioned regarding case studies, namely the lack of complete randomness in the selection process. Thus fully generalizing the results is not possible, but I believe the findings give a strong indication as the current situation of availability and validity of empirical evidence for claims. The results are especially indicative of the state of affairs with regards to the certain popular tool and technology types predominantly studied, namely IDEs, Ajax toolkits, VCS/SCMs and similar, and modelling tools.

Another limitation refers to the fact that only specified external sources were looked into. There may be a certain amount of empirical research available from independent sources, at least if Fenton et al.'s (1994) recommendations have been followed. However, this thesis specifically wished to see what evidence the tool providers themselves had available. The desired end result is that companies considering a tool implementation would be able to acquire independent data from the tool providers themselves, rather than spend resources on going through other channels. It is also a fairly acceptable assumption that a tool provider would have a stronger case if they were readily able to support their claims with independent empirical experiments.

As mentioned in the previous section, ten of the non-respondents scored a 1 or a 0. This might indicate a limitation as to the correctness of the overall results. More specifically, the lack of response may be due to more than a decision to not participate. For instance, in Coverity's case, the response was delayed because "your email was routed to me after others didn't know what to do with it" (2008 Chou). There is no way to know whether some companies have so far neglected to respond due to the information requests not reaching the correct person in time, although

every effort was made to find the most direct and appropriate contact information. In some cases, however, no more precise e-mail address than e.g. info@....com was discovered. Another factor that may have affected both the response rate and the response quality is the fact that the information requests were sent by a student. The responses might not have been similar for a potential customer. IBM, for example, did indeed reply, but stated in one response that they could not help me as a student.

4. Discussion of Results

4.1.1 Claims

Approximately half the tools addressed in this study have made claims that seem to lack a foundation in fact. More specifically, claims such as “no quicker ways to build an Ajax application” (Enterprise Ajax) and “Experience unparalleled productivity” (JBuilder) make inflated statements that cannot realistically be tested. It would be far too wide an operation to set up experiments testing all available Java IDEs in various situations in order to prove JBuilder’s “unparalleled productivity”. So why do the providers of these tools make such claims? Presumably it is because they work, despite coming from a source that in the marketing context can hardly be judged as credible. The sleeper effect (2004 Kumkale and Albarraci) indicates that a recipient of information with a discounting cue (such as the source having low credibility) may over time be more persuaded by the information than the initial impact, as the connection between the information and its source becomes diffuse as time goes by.

A study by Jørgensen and Grimstad (2008) looked into the impact of misleading, one-sided information on software professionals’ judgements and the ability to readjust their opinions when the information was revealed to be misleading. The findings showed that the final judgements may be “permanently distorted” by the original, wrongful information, even when later being told the arguments had no or little validity. The results also showed that predicting when one’s judgements would be readjusted correctly can be difficult. With that in mind, making very strong claims about one’s tools may have a lasting positive impact on potential customers, even if the claims are later shown to be invalid. This, in turn, means that it may not be enough to simply request empirical evidence when in the process of finding a software engineering tool to use in one’s organization, as by doing so, one will likely already have encountered some of the claims and been impacted by them. Therefore a consequence might be that when in the decision-making process one should look at

external sources of empirical evidence first, perhaps through other companies known to have used the tool themselves or through the use of independent research organizations. The problem then lies in finding valid evidence from those sources. As my review of the available material has shown, case studies are not so difficult to come by, but the truly useful ones are.

A more general reason why there is a strong lack of valid empirical evidence in circulation, both for the types of claims mentioned above and the less outlandish ones, may be that while the claims can be difficult or costly to prove, they may often be far easier to disprove (or at least diminish the value of). For instance, when Borland says that Together will enable improved “quality, cost and team communication”, a single study may show that this is not always so. However, when I have commented that decision-makers need studies that include enough information to determine if the situation and requirements are comparable, that does not only relate to the positive studies. If a study makes a negative conclusion regarding a product, it should again be taken into account whether the study can apply to one’s own case or not, as well as what other studies show.

4.1.2 Case Studies

Overall, case studies are no doubt the most popular empirical tool employed by the companies researched to convince potential new users that the tools live up to the linked claims. In fact, Darke et al. (1998) say such studies are “the most widely used qualitative research method in information systems research”. Case studies are indeed effective to an extent in showing how a tool works in real-life settings, but with case studies being such an integral part of marketing of software engineering tools it is important to consider whether or not they can be trusted to tell the whole truth.

Although the case studies in question may very well give a correct depiction of experiences, one of the major points to contemplate is how the reference clients were selected in the first place. It is highly unlikely that an SE tool vendor would present any negative case studies, but it is also unlikely that only positive experiences exist.

As far as I have been able to determine, none of the case studies encountered during my research delved into the case company selection process. During a presentation at JavaZone 2007, Jørgensen (2007) gave an example of how Microsoft requested new reference clients by asking them to explain how Microsoft solutions and products helped strengthen business opportunities. Furthermore, the reference clients would not only get the chance of being showcased "as a company taking new and cost-effective technology in use"[107]; quick respondents would be included in a draw for 10 gift certificates. In other words, Microsoft was only interested in hearing from clients who had had a positive experience with their solutions. Additionally, respondents would get valuable, free advertising as a showcased company, making it potentially tempting for reference clients both to put themselves and their experiences in an especially positive light. Furthermore, a slight psychological barrier might prevent clients from writing negative about a company promising the chance of gifts. As Jørgensen also notes, Microsoft are probably not alone in using this approach.

As far as measuring of effects mentioned in case studies, providing quantitative data is the exception, not the rule. Empirical evidence relies strongly on metrics proving effects – a specific percentage increase in productivity compared to the previous tool used, a certain percentage lower costs compared to using competing products, and so on. Of course certain elements are difficult, if even possible, to measure in numerical values, such as ease-of-use. In those cases, though, it should still be possible to describe how many users found the new tool easier to use, how such a conclusion was reached, and so on. Naturally, all the case studies described benefits – that is the whole point. How, though, can a reader determine exactly what caused the benefits, how strong the improvements were, and compared to what? In order to get valid empirical evidence from the case studies, details regarding which situations or products are compared and metrics from the comparisons are needed. Since case studies claim benefits, one will be inclined to believe that comparisons have been

[107] Translated from Norwegian: "som et selskap som tar ny og kostnadseffektiv teknologi i bruk".

made, whether truly scientific, with metrics (when applicable) recorded, or based on gut feelings. This latter point signifies another important point. Potential clients should not only be interested in finding out why a reference client experienced improvements, but also how they came to conclude that improvements were made. Did they undergo a thorough study after implementation of the new product, to measure the effect on productivity, code quality, TCO, time-to-market, and other claimed gains? For instance, in BitKeeper's case, a client claimed saving thousands of hours by migrating to that tool. Being a testimonial and not a case study, even less detail was available. So how can we know or believe that the client had performed strict tests in order to conclude that thousands of hours were saved? Detailed descriptions help, but detail beyond for example "performance was higher" or "the tool was by far the easiest to use" was rare in almost all the case studies.

Another reason why case studies ought to include more detail regarding the cause behind the benefits is that a reader might easily suspect that not only the new tool or solution, but other, external factors have influenced the stated effects. In the book "Case Study Research" (2003), Yin explains that as a typical "rival explanation" and that someone conducting a case study should be aware of this issue beforehand so that evidence as to the effect of the other factors could be collected. Such information has been exceptionally rare in the case studies I have processed. Whether there is a lack of understanding for the value of the extra effort or such data in fact does exist, but was simply (in more or less all cases) neglected, is of course not known. One would imagine, though, that the information would have been included if available, unless deemed unimportant.

As mentioned in the Microsoft example above, something to keep in mind when reading a case study is that such a study is "free" advertisement for the client reference as well. Case studies tend to speak *very* positively about the solutions created by the studied company and the company in general, implying both that a successful company used the tool and that the tool helped the company achieve success. It is highly unlikely that any reference client would accept publication of a

case study showing a lack of or reduced market performance. Besides that there is the aforementioned possibility of other factors affecting success (or for that matter, failure). Therefore any case study not done by an independent, third party, and elaborating on the success of the case company, should be taken with an additional pinch of salt.

4.1.3 Marketing Principles

According to Armstrong (2008), endorsements are a powerful persuasive tool. However, when it comes to high-involvement products, which SE tools surely are, Armstrong writes that endorsements by experts may be stronger than endorsements by celebrities or typical consumers. Certain individuals (such as Linus Torvalds in BitKeeper's case) and certain companies (like Google in Perforce's case) may perhaps be seen as "celebrities" in the SE world, due to their generally high status in the SE community. However, while they are experts in their respective fields, they may not by all be considered experts within the subject area of the tool they are endorsing. The same certainly goes for the individuals and organizations who do not have quite as high status (and thus not considered "celebrities" in this context), but who as "typical customers" endorse the products through case studies. Following on what Armstrong writes, endorsements should be made by independent, third-party experts in order to have a stronger marketing effect. Furthermore, the endorsements should be easily accessible and mentioned earlier rather than later in the marketing, to increase the persuasiveness through a better first impression.

Although Armstrong writes about advertisements in general, the advise is likely transferable to the SE area as well. As such the above indicates that not only should the SE tool providers place case studies and similar (e.g. testimonials) within easy access from the placement of the claims (something which very often was not the case for the tools studied), but they should be the product of independent experts for best marketing effects. Simultaneously, there is little doubt that potential customers of SE tools would benefit greatly by having valid, independent empirical evidence as basis

for their decisions. The above thus appears to suggest that independent empirical evidence is a positive for both sides of an SE tool implementation process. Therefore it becomes somewhat of a mystery that so little such evidence is provided.

By going through the 23 tools in this study, something that becomes quite apparent is the similarity in claims. Productivity, performance, code quality, ease-of-use, cost; these are all benefits that are claimed by the majority of the tool providers. This goes against the marketing concept of a Unique Selling Proposition (USP); the act of marketing a unique point to a tool, gaining an apparent advantage to similar products. That is not necessarily to say that the tool in question is the only one that produces or gives the marketed effect, but that it is the only one to advertise it. The consequence of the SE tools *not* employing this technique, but rather focusing on the same general issues is that, ultimately, some of the claims must be faulty. Imagine several IDEs claiming improved productivity, as was the case among most of the IDEs in this study. If all the IDEs are compared against each other, pure logic defies the claims that all of them will improve productivity. Common reasoning dictates that at least some of the claims are either wrong altogether or do not tell the whole story, which in turn may make potential customers more sceptic towards the tools. It may be that these claims are considered efficient enough by the tool providers, but the USP concept indicates that there are even more efficient possibilities. A further consequence of the above is that it becomes even more important for anyone considering acquiring a new SE tool to closely examine the claims and request empirical evidence for them. If “everyone” claims the same, find out whose claims have the most empirical support.

Brand or company name recognizability has been launched as a possible reason for the lack of evidence for some studied tools. Especially IBM’s situation could be construed as boiling down to the power of brand and company names. IBM are huge in the SE industry, while their brand Rational is also widely known and accepted. The way IBM presented their claims, with three of them starting with “Rational ClearCase”, further supports this theory. According to Armstrong, brand exposure

affects the attitudes of customers, unbeknownst to the customers themselves. An implication is that SE decision-makers ought to make an effort to be aware of brand power when comparing products and look at the results of tests as objectively as possible. Perhaps a solution would be to delegate the gathering of information (preferably empirically valid at that) to another individual, and have the results made anonymous before evaluating them.

4.1.4 Valid Evidence

The findings summary in chapter 3 indicated that 22% of the tools had what was defined as “some empirical evidence” for the related claims, meaning that a portion of the claims appeared to be verified by either independent studies or highly detailed non-independent studies. While generalizing my results is, as mentioned earlier, not entirely statistically possible, this number is still interesting in the light of a mid-90s study which surveyed then-recent research articles in refereed computer science publications (1995 Tichy et al.). It was found in that study that over half the articles relating to software engineering and about new designs and models lacked experimental evaluation. When considering only articles spending at least one fifth of the article space on evaluation (admittedly a somewhat weak measure), only 20% of the SE related articles passed. While the study looked at empirical experimental evaluation for claims made in research articles and I have looked at similar information available via tool providers, it is interesting to note that the amount of valid material – or rather, the amount of tools/models related claims with valid support – does not appear to have increased over the years, but quite the contrary. As such there is an indication that still more needs to be done regarding empirical evaluation of claims than what is currently being done.

5. Conclusions

5.1 Summary and Conclusion

There is no doubt many software engineering practitioners have large volumes of knowledge as to the effects of various methods, technologies and tool features. Such individuals may therefore regard empirical evidence as less of a necessity when evaluating new SE tools, relying instead on personal expertise. However, the point of empirical evidence is to see if the tool manages to produce the desired effects for your specific situation. Not only that, but do the tools manage to produce the desired and claimed effects at all, or is the implementation of the, possibly tried and true, techniques faulty?

The results showed that very little valid empirical evidence exists to support the claims presented. Additionally, case studies were the most popular empirical persuasion tool. Presumably most of the SE tool providers know what works best and use case studies because potential customers appreciate them. However, as has been shown, the case studies have several rather severe faults as far as being valid as empirical evidence. It was noted how all the case studies, as far as I was able to determine, lacked information about how the case companies were selected. The Microsoft example showed how companies may acquire client references in a manner hardly acceptable as random, but instead indicated that the companies go out of their way to get positive studies. Furthermore, the case studies lack quantitative information, i.e., measurements of results from usage. In the few cases where such metrics were at least partly present, detailed information regarding what the data was derived from was missing, including potential external factors. Finally, case studies were shown to function as advertisements for the case companies as well, limiting the objectivity. Overall, the validity of case studies, the most widely used empirical material, was considered very low in general.

The frequent use of case studies as empirical support for claims has in this study often been shown to be insufficient in the SE environment. As a result, software engineering practitioners should be made aware of the potential lack of their validity, and thus be vigilant when scrutinizing them. They should also keep in mind the sleeper effect, namely that they dissociate the non-credible source of claims over time, the difficulty in readjusting the impact of a message from an original low credibility source, as described by Jørgensen and Grimstad (2008), and also the subconscious effect of powerful brand names, as detailed by Armstrong (2008). The above indicates that when in the process of acquisition of a new tool, one should start by looking towards independent sources. However, previous research has shown that there appears to be a lack of interest in empirical validation among SE researchers in general as well. Besides, as I have stated earlier, the providers of a tool ought to serve as the ultimate resource of independent empirical research on the tool. As such, I have suggested using an intermediary – be it another individual in one's organization – to gather and make anonymous empirical research from the SE tool providers.

As the USP concept indicates, as well as common sense, not all claims by all vendors of a certain type of tool can be entirely truthful, specifically claims related to the same benefit. SE practitioners should use valid empirical evaluation to distinguish between tools that can deliver and tools that cannot, for instance through processes such as that outlined by EBSE. For that to be possible, there needs to be independent empirical data available on a fairly large scale, something my research indicates that at present is not the case, while at the same time showing through a few positive cases that producing such material *is* possible. A logical reason why the material is not available on the desired scale is that the customers do not request it. Basic economics tells us that if there was a large demand for empirical research on the validity of claims made regarding SE tools, then there would also over time be built up a supply. My advice to SE practitioners is therefore that they utilize their power as consumers of SE tools. Become aware of the pitfalls of current marketing practices by SE tool providers and start demanding truly valid empirical evaluations that can support the tools. Only then does it seem likely that such data will materialize on a sufficient scale. As

endorsements by independent third party experts is claimed to have a strong effect on consumers, a larger supply of valid empirical research should be considered a win-win situation. That is, unless empirical experimentation refutes the SE tool providers' claims – in which case only the customer wins.

5.2 Further research

The findings of this thesis provide a basis for several potential further research endeavours. One very interesting problem to look into is the amount of empirical evidence that creates the basis of reference clients' claims. Do they conduct valid empirical experiments, or do they base their conclusions on perceived experiences? Additionally, how is an implementation evaluated as successful? If it is measured in monetary terms, such as gains in revenue, have all measurable external factors been accounted for? Furthermore, not only how, but *is* success evaluated? Conducting an evaluation before purchasing a tool is expensive enough in itself. Having finally decided on and implemented a tool, are there resources, or even a will, to fully evaluate the success or failure of the implementation?

Another interesting case is the selection of reference clients. As I discovered during my research, I was unable to identify such information in any of my material. Knowing the procedure behind the selection could add a great deal of credibility to the case studies. Conversely, it could also undermine their value completely.

A third possibility is looking into how decision-makers in an SE tool implementation process perceive the necessity and value of empirical evidence. Do they rely on their own skill sets, do they seek truly independent material, do they consider the various factors mentioned in this thesis, or do they have completely different methods?

6. Appendix A

6.1 Example – Information Request E-mail

To whom it may concern

As a master's degree student at the University of Oslo (UiO), Norway, and with the guidance of my supervisor, Professor Magne Jørgensen of Simula Research Laboratory, I am working on a thesis about software development tools, related to evidence-based software engineering (EBSE). More precisely, I am looking into statements made by producers of such tools and the existence and availability of empirical research and evidence behind those statements. The generalized conclusions of the thesis are likely to be of interest to companies working on selecting tools to use in their software development process.

Your tool, *[NAME OF TOOL]*, was chosen to be included in my thesis. Researching the associated website, I have come across certain claims made regarding the tool, reiterated below:

1. *[LIST OF CLAIMS]*

What I wish to request from you is any empirical information (preferable scientific studies with a control group, but also case studies describing industry experience or other types of studies) you may have to support the claims above. We are aiming towards publication in IEEE Software. We will, if it is of interest, send you the full report.

Best regards,

Espen Brunsvig, UiO
Prof. Magne Jørgensen, Simula

7. Appendix B – Correspondence

This appendix includes all correspondence between the SE tool providers and myself, except the original information requests, the information request repetition for those who did not respond initially, any auto-responses or similar, and “thank you” e-mails from me.

7.1 Adobe Flex Builder 3

No response.

7.2 Backbase Enterprise Ajax

No response.

7.3 BitMover BitKeeper

No response.

7.4 Borland Together

No response.

7.5 CodeGear JBuilder

No response.

7.6 Coverity Prevent

Chou, Andy <achou@coverity.com> wrote Sat, Apr 19, 2008 at 1:58 AM:

Espen,

I apologize for the long delay in replying, your email was routed to me
after others didn't know what to do with it. I can help you with some

basic answers to some of your questions, and hopefully that will be enough for the purposes of your thesis.

> evidence-based software engineering (EBSE). More precisely, I
> am looking into statements made by producers of such tools
> and the existence and availability of empirical research and
> evidence behind those statements. The generalized conclusions
> of the thesis are likely to be of interest to companies
> working on selecting tools to use in their software
> development process.

We do have an academic licensing program, but it is restricted to "classroom use" -- specifically for use on class projects and in some cases for classes intended to teach students about state of the art software tools. Here's a publicly available report written by some students at CMU who evaluated our product (a relatively old version now):

http://www.cs.cmu.edu/~aldrich/courses/654/tools/cure-coverity-06.pdf

We did not provide any input to these students except for the product and its documentation.

A commercial customer of ours, Ericsson, in conjunction with a professor at Linkoping University, wrote a comparative study of our product against a couple of others:

http://www.ep.liu.se/ea/trcis/2008/003/trcis08003.pdf

Ericsson is currently a significant Coverity customer.

> Your tool, Coverity Prevent, was chosen to be included in my
> thesis. Researching the associated website, I have come
> across certain claims made regarding the tool, reiterated
> below (bold formatting not original):
>
> 1. "Prevent is the industry standard because only
> Coverity understands the strict requirements of static source
> code analysis."[1]

This is partly market-speak, however it is true that Coverity pioneered many practical techniques that were needed to successfully scale and deploy static analysis while providing low false positives. In terms of scale, we regard 500k LOC as small, whereas most academic tools would consider that pretty large. In terms of deployment, we were the first to devise methods for seamless build integration that requires no changes to the build system.

> 2. "Prevent identifies more information about source

> code than any other product on the market today [...]."1

We believe that some other tools don't truly parse the source code, but rather are semi-parsing it. We use the proven EDG front-end for C/C++ and actually perform full parsing. We also believe that our build integration is the most accurate, ensuring that source files referenced by the build system are not missed.

> 3. "[...] easy-to-use tools that allow defects to be
> addressed within minutes."1

Hopefully the papers above will provide enough evidence that our tool can analyze some code bases very quickly, and that bugs can be triaged very quickly.

> 4. "Prevent for Java's technical breakthroughs result
> in unmatched improvements to the quality of critical Java code"[2]

Our internal measurements show that we perform better than open source tools such as Findbugs, but unfortunately this data is not publicly available.

> 5. "Out of the box, Coverity Prevent has an average
> false positive rate of less than 20%."[3]

We have internal data that shows this over millions of lines of open source code that is analyzed as part of Scan (scan.coverity.com), however I don't believe that is publicly available. Through trials and customer feedback, we have found that this claim is correct on average for commercial code as well.

> 6. "Coverity Prevent seamlessly integrates with your
> existing environment and can be deployed and configured
> within hours. Other tools can take weeks and even months to
> set up and configure, costing you precious time and resources."3

See the papers referenced above.

I hope this is enough for you to go on. I'd look forward to reading the final result.

Regards,
-Andy

7.7 Dojo Toolkit

Adam Peller <peller@dojotoolkit.org> wrote Mon, Apr 7, 2008 at 4:39 PM:

Hello Espen.

We don't have any budget or resources for our own case studies, though I'm sure many have been done in the industry. Like many open-source development organizations, we don't have very much time to spend on PR. Most of these claims are left as exercises for the reader (demos, tests and benchmarks are included with the toolkit, for example) You should find a good deal of commentary regarding Dojo and other toolkits on websites like ajaxian.com or from industry analysts, though frankly there just aren't enough objective, accurate and unbiased studies. The articles I've seen in print publications seem to be the biggest offenders, often using obsolete versions of the code. Comparisons are often difficult, since the feature sets can differ so much across toolkits. I don't have any references for you off hand, I'm afraid.

My employer, IBM, is using Dojo in products, as is AOL, another major Dojo contributor. We have seen Dojo in use in many high traffic sites, but as our licensing terms are so generous, we usually have no idea who is using our code, and we have no agreement in place to publicize their products.

Regards,

Adam

7.8 Eclipse IDE for Java Developers

No response.

7.9 Gentleware Posedion for UML

Kateryna <kateryna@gentleware.com> wrote Tue, Apr 22, 2008 at 2:39 PM:

Dear Espen,

thank you for your email.

Let me provide you with 2 documents, which can help you in your research.
You also can surf through the website in order to find more information
regarding Poseidon:

http://www.gentleware.com/products.html
http://www.gentleware.com/support.html
http://www.gentleware.com/poseidonfaq.html

If you would like to have deeper view, you can evaluate Poseidon here:
http://www.gentleware.com/eval.html.

We do not provide any scientific researches.

If you have any questions regarding our tools, please feel free to contact us again.

Kind regards,
Kateryna Derevenskykh
International Sales
Gentleware AG

7.10 Google Web Toolkit (GWT)

Sumit Chandel <sumitchandel@google.com> wrote Tue, Apr 8, 2008 at 5:35 PM:

Hi Espen,

Sorry for the delay in replying back to your inquiry.

I think the best way for you to get case studies and potentially empirical data related to GWT that supports those statements would be to approach some of the GWT developers who went through the process of comparing alternatives and who chose GWT to build their applications. These developers are the people who really know how beneficial GWT was to them in creating their application and they express their appreciation of the toolkit on the developer forum.

Below you will find some references that should help you establish case studies that substantiate some of our claims as well as furthering your publication with empirical data. There are also a number of presentations given by the GWT team that you may find useful for your thesis.

Contact Office
Demo: http://www.contactoffice.com/
Contact: Luc Claes (luc.claes@gmail.com)

Lombardi Blueprint
Demo: http://blueprint.lombardi.com/ (you must request a userID to login to the demo)
Contact: Palak Shah (pshah@lombardi.com)
Blog post: http://googlewebtoolkit.blogspot.com/2007/10/lombardi-blueprint-built-with-gwt.html

QueWeb Customer Care
Demo: http://www.queplix.com/
GWT testimonial: http://www.queplix.com/solutions/google-gwt-technology/

Timepedia
Website: http://timepedia.org/
Contact: Ray Cromwell (cromwellian@gmail.com)

Here are some links to presentation material you may find useful:

http://code.google.com/webtoolkit/presentations.html
http://code.google.com/webtoolkit/articles.html
http://code.google.com/webtoolkit/books.html

Hope that helps!

Regards,
-Sumit Chandel

7.11 IBM Rational ClearCase

<askibm@vnet.ibm.com> wrote Thu, Mar 27, 2008 at 3:21 AM:

> Dear Espen,
>
> Thank you for contacting IBM.
>
> We appreciate your interest in writing to us in this regard.
>
> Due to the large volume of e-mail that IBM receives, our representatives are unable to assist students with research requests directly. However, we have provided online alternatives where you can search for the topics of your choice. These resources are located at:
>
> http://www.ibm.com/ibm
> http://www.ibm.com/university
> http://www.research.ibm.com
>
> You may also find information by searching the main IBM Web site to aid you in your research:
>
> http://www.ibm.com
>
> Thank you for visiting our site and contacting IBM.

Yogendra Sharma
Electronic Response Center

Norway Info GSC <ibminfo@no.ibm.com> Fri, Mar 28, 2008 at 3:04 PM:

Hei Espen.

Forelår at du sender mail. til Jo Eriksen i IBM Rational om dette. Hans adresse er: ERIKSEN@no.ibm.com

Mvh. Espen Haukaas
IBM Scotland

Translation: ”Suggest you send a mail to Jo Eriksen at IBM Rational about this. His address is: ERIKSEN@no.ibm.com”. Suggestion followed, no further response received.

7.12 JetBrains IntelliJ IDEA

Serge Baranov <serge@jetbrains.com> wrote Thu, Mar 20, 2008 at 10:10 AM:

Hello Espen,

We don't have such information.

Serge Baranov
JetBrains, Inc

Espen Brunsvig wrote Fri, Mar 28, 2008 at 1:23 PM:

Hello again Serge,

I would greatly appreciate if you could clarify what you mean by "We don't have such information". Am I to interpret that as meaning there is no such information available at all or should I interpret it as meaning that you may/do have information, but cannot or would prefer to not send it?

Regards,
Espen Brunsvig

Serge Baranov <serge@jetbrains.com> wrote Fri, Mar 28, 2008 at 1:30 PM:

Hello Espen,

We didn't perform any special studies or research concerning the mentioned statements. Some of them are obvious and can be easily observed, others can be googled for to find more facts. I don't even know why such studies would be needed.

Serge Baranov
JetBrains, Inc

7.13 Microsoft Visual SourceSafe/Visual Studio 2008

Rune Zakariassen <runez@microsoft.com> wrote Thu, Mar 27, 2008 at 2:15 PM:

Hei Espen,

Jeg jobber nå med å samle sammen empirisk og customer case study material for oversendelse til deg. Planen er å sende deg dette over helgen.

Mvh.
Rune Zakariassen
Fagansvarlig for forskning og høyere utdanning
Microsoft Norge AS

Espen Brunsvig wrote Thu, Mar 27, 2008 at 2:22 PM:

Hei Rune,

Takk for det!

Jeg er akkurat ferdig med å forberede forespørsel nummer to (av to) til Microsoft, denne gangen angående Microsoft Visual SourceSafe. I og med at det er vanskelig å finne frem til kontaktpunkter utover generell support tillater jeg meg å sende den forespørselen også til deg. Jeg håper du har mulighet til å hjelpe med den også, enten ved å samle sammen materiale eller ved å henvise meg videre.

[Request for second tool]

Rune Zakariassen <runez@microsoft.com> wrote Thu, Mar 27, 2008 at 2:25 PM:

Hei Espen,

Har du et telefonnummer jeg kan ringe deg på?

Mitt er XXX XX XXX.

Mvh.
Runez<

Espen Brunsvig wrote Thu, Mar 27, 2008 at 2:25 PM:

Hei,

Nummeret er XX XX XX XX.

Mvh,
Espen

Rune Zakariassen <runez@microsoft.com> wrote Mon, Apr 14, 2008 at 1:49 PM:

Hei Espen,
Beklager at det har tatt noe tid. Men her har du matriellet jeg lovte.

Mvh.
Runez<

Espen Brunsvig wrote Thu, Apr 17, 2008 at 7:38 PM:

Hei Rune,

Takk for dokumentene du sendte, nå har jeg fått tittet litt på dem. Jeg ser at de i all hovedsak omhandler Microsoft Visual Studio Team System, noe 2008, mest 2005. Betyr det at du ikke fikk tak i dokumenter angående MS Visual SourceSafe?

For å oppklare eventuelle misforståelser; stemmer det at både IDE-en Visual Studio og produktet VSTS hører inn under "Visual Studio 2008"? Ut i fra oversikten over de respektive utgavenes egenskaper (på Microsofts nettsider) ser det ut til at VSTS-utgavene inneholder alt VS-utgavene inneholder, pluss en del ekstra egenskaper.

Mvh,
Espen

Rune Zakariassen <runez@microsoft.com> Fri, Apr 18, 2008 at 5:11 PM:

Hei Espen,

Som jeg nevnte i vår telefonsamtale så vil det ta noe tid å samle inn informasjon om Visual SourceSafe. Jeg er opptatt et stykke frem i tid.

Du har helt rett i forhold til VSTS. Det er VS + mye ekstra rundt utvikling av programvare i team.

Mvh.
Runez<

7.14 NetBeans IDE

Tori Wieldt <Victoria.Wieldt@sun.com> wrote Wed, Apr 9, 2008 at 12:28 AM:

Hello,
We don't have any data about the specific claims you mention, but you can find reports from our usability tests:

http://ui.netbeans.org/

and statistics about use of NetBeans IDE, describing the quality and stability of NetBeans:

http://statistics.netbeans.org/analytics/
--
Tori Wieldt, Managing Editor, www.netbeans.org

7.15 No Magic MagicDraw

No response.

7.16 Perforce

Dave Robertson <eval-demo@perforce.com> wrote Wed, Mar 26, 2008 at 5:40 PM:

Hello Espen,

Thank you very much for your email.

> I am working on a thesis about software development
> tools, related to evidence-based software engineering (EBSE). More
> precisely, I am looking into statements made by producers of such tools and
> the existence and availability of empirical research and evidence behind
> those statements. The generalized conclusions of the thesis are likely to be
> of interest to companies working on selecting tools to use in their software
> development process.
>
> Your tool, Perforce, was chosen to be included in my thesis. Researching the
> associated website, I have come across certain claims made regarding the
> tool, reiterated below (bold formatting not original):

We'd welcome the chance to assist you in your thesis.

Could you give me an indication how much time we have to prepare a response for you? This will allow me to plan the best way to respond.

Regards
--
Dave Robertson

Perforce Software Europe
eval-demo@perforce.com

Espen Brunsvig wrote Wed, Mar 26, 2008 at 6:55 PM:

Hello Dave,

Thank you for your positive reply. To allow me ample time to analyze responses before my final thesis must be handed in it would be preferential for me to receive a response from you within the first couple of days of the week beginning April 7th. I hope this gives you enough time to put together a response.

I notice that you have a case study on your website (about National Instruments) which I plan to look into as well.

Regards,
Espen Brunsvig

Dave Robertson <eval-demo@perforce.com> wrote Thu, Mar 27, 2008 at 6:48 PM:

Hello Espen,

> Thank you for your positive reply. To allow me ample time to analyze
> responses before my final thesis must be handed in it would be preferential
> for me to receive a response from you within the first couple of days of the
> week beginning April 7th. I hope this gives you enough time to put together
> a response.
Whilst we have limited resources to responded in the amount of detail you ask for, we will try to give you some empirical data next week.

I have asked our marketing team in the US for their help.
- Hide quoted text -

Regards
--
Dave Robertson
Perforce Software Europe
eval-demo@perforce.com

Espen Brunsvig wrote Wed, Apr 9, 2008 at 1:36 PM:

Hello again Dave,

Although I realize your resources are limited, I wish to send a friendly reminder regarding our data request. Will you be able to send us any

empirical data this week?

Best regards,
Espen Brunsvig
Prof. Magne Jørgensen

Dave Robertson <eval-demo@perforce.com> wrote Thu, Apr 10, 2008 at 10:08 AM:

Hello Espen,

> Although I realize your resources are limited, I wish to send a friendly
> reminder regarding our data request. Will you be able to send us any
> empirical data this week?
Apologies for the silence from this side. We have our European User Conference
in London next week and I've had to enlist help from some colleagues in the US.

We hope to have something for you in the next few days.
- Hide quoted text -

Regards
--
Dave Robertson
Perforce Software Europe
eval-demo@perforce.com

---

ttyler@perforce.com wrote Wed, Apr 9, 2008 at 9:53 PM:

Hello, Espen!

I am writing in response to your query regarding your Master's thesis.
I am happy to be your contact regarding this. I have your initial list
of claims for, and I will find whatever case studies or other
information we might have that will support the listed claims.

Cheers!

Tom

C. Thomas Tyler
Perforce Software
Consulting Services

Espen Brunsvig wrote Wed, Apr 9, 2008 at 10:02 PM:

Hello, Tom!

Thank you for your positive response! As I have certain deadlines to meet,
I will be looking forward to hearing from you again soon with whatever
material you find.

Best regards,
Espen

Tom Tyler <consulting@perforce.com> wrote Mon, Apr 14, 2008 at 2:32 PM:

Hello, Espen!

I am working on this presently, and will get back to you this evening your time.

Just FYI, I am actually in *[location]* this week.

Cheers!

Tom

Tom Tyler <consulting@perforce.com> wrote Tue, Apr 15, 2008 at 4:06 PM:

Hello, Espen!

I have attached a document providing some evidence for those claims made, mostly in the form of case studies, but also including some commentary from my own experience consulting with Perforce.

Please don't hesitate to ask any additional questions. I apologize any delays in getting this to you, as I have been extremely busy these past few weeks. I find your EBSE topic extremely interesting, and will be very much interested in the full report.

Good Luck with your thesis!

Cheers!

Tom

P.S. Over the next 2 days, I'll be attending and presenting at the Perforce European User Conference, and so I'll be difficult to reach, but then easier to reach after that.

7.17 Sparx Systems Enterprise Architect

sparxmarketing <marketing@sparxsystems.com> wrote Thu, Mar 27, 2008 at 8:01 AM:

Hello Espen,

Thank you for your email.

I am currently collecting our case studies and additional research on the points below. I hope to be in contact soon.

Your introduction suggests you are primarily interested in empirical research conducted ourselves, however if there exists evidence compiled by external sources, would you wish to be referred to that material too?

Best Regards
Estelle Gleeson
Marketing Coordinator
marketing@sparxsystems.com
http://www.sparxsystems.com

Espen Brunsvig wrote Thu, Mar 27, 2008 at 10:09 AM:

Hello Estelle,

Thank you for your positive response. I would indeed be interested in being referred to external sources as well.

Hope to hear from you again soon.

Best regards,
Espen Brunsvig

---

sparxmarketing <marketing@sparxsystems.com> wrote Thu, Apr 10, 2008 at 5:22 AM:

Hello Espen,

Please find links below to all the research available. I have also attached a list of citations of Enterprise Architect in industry.

Case Studies: http://www.sparxsystems.com/press/index.html#CaseStudies
Independent Analyst Reports:
http://www.sparxsystems.com/press/index.html#AnalystReports

We are also at present conducting a survey, however the results are not yet available.

Best Regards
Estelle Gleeson
Marketing Coordinator
marketing@sparxsystems.com
http://www.sparxsystems.com

7.18 SpringSource Spring Framework

No response.

7.19 Tata MasterCraft

No response.

7.20 Telelogic Synergy

No response.

7.21 ThinWire

No response.

7.22 TIBCO General Interface

Michael Peachey <mpeachey@tibco.com> wrote Mon, Apr 7, 2008 at 6:29 PM:

Hi Espen –

Our marketing person has left, and he would have been the right one to talk to,

Sorry.

michael

Espen Brunsvig wrote Mon, Apr 7, 2008 at 8:28 PM:

Hello Michael,

Thank you for your response. I take what you wrote as meaning your marketing person is not reachable in the foreseeable future, is that correct?

Regards,
Espen

Michael Peachey <mpeachey@tibco.com> wrote Mon, Apr 7, 2008 at 10:25 PM:

We don’t have a replacement yet. I also highly suspect that there was nothing scientific about the claims anyway.

References

2008, A. Chou, Coverity, *Personal correspondence*

2008, A. Peller, Dojo, *Personal correspondence*

2008, K. Derevenskykh, Gentleware, *Personal correspondence*

2008, S. Baranov, JetBrains, *Personal correspondence*

2008, C.T. Tyler, Perforce, *Personal correspondence*

2008, M. Peachey, TIBCO, *Personal correspondence*

2008, M. Jørgensen, S. Grimstad, *"Judgment-updating among software professionals"*, In: The 2nd international conference on software knowledge information management and applications (SKIMA), ed. by Hossain and Ouzrout, pp. 62-67, SKIMA conference organising committee (ISBN: 9781851432516)

2008, P. Emanuelsson and U. Nilsson, *"A Comparative Study of Industrial Static Analysis Tools (Extended Version)"*, Linköping University, http://www.ep.liu.se/ea/trcis/2008/003/trcis08003.pdf

2008, J.S. Armstrong, "*Persuasive Advertising – Evidence-based principles"*, Version 216, March 8, 2008 – yet to be published

2007, M. Jørgensen, *"Rasjonelle beslutninger ved valg av metoder og verktøy"*, JavaZone 2007 presentation

2006, F.J. Fowler, Jr., *"Survey Research Methods"*, 3rd Edition, Sage Publications, Inc.

2006, P. Wayner, "Top AJAX tools deliver rich GUI goodness", InfoWorld, http://www.infoworld.com/article/06/11/27/48FEajax_1.html

2006, A. Almossawi, K. Lim, T. Sinha, *"Analysis Tool Evaluation: Coverity Prevent"*, Carnegie Mellon University, http://www.cs.cmu.edu/~aldrich/courses/654/tools/cure-coverity-06.pdf

2005, T. Dybå, B. Kitchenham and M. Jørgensen, *"Evidence-based Software Engineering for Practitioners"*, IEEE Software, Vol. 22(1), p. 58-65

2004, G.T. Kumkale and D. Albarraci, *"The Sleeper Effect in Persuasion: A Meta-Analytic Review"*, Psychological Bulletin, Vol. 130(1), p. 143-172

2003, R.K. Yin, *"Case Study Research – Design and Methods"*, 3rd Edition, Sage Publications, Inc.

2000, S.L. Pfleeger, W. Meenzes, *"Marketing Technology to Software Practitioners"*, IEEE Software, Vol. 17(1), p. 27-33

1998, P. Darke, G. Shanks, M. Broadbent, *"Successfully completing case study research: combining rigour, relevance and pragmatism"*, Information Systems Journal, Vol. 8(4), p. 273-289

1996, D.L. Sackett, W.M.C. Rosenberg, J.A.M. Gray, R.B. Haynes, W.S. Richardson, *"Evidence based medicine: what it is and what it isn't"*, BMJ 312, p. 71-72

1995, W.F. Tichy, P. Lukowicz, L. Prechelt, E.A. Heinz, *"Experimental Evaluation in Computer Science: A Quantitative Study"*, Journal of Systems and Software, Vol. 28(1), p. 9 - 18

1994, N. Fenton, S.L. Pfleeger, R.L. Glass, *"Science and substance: a challenge to software engineers"*, IEEE Software, Vol. 11(4), p. 86-95