Information review and instructional design at The Astonishing Tribe.
# Table of Contents

Abstract ................................................................................................................................. 4
Preface .................................................................................................................................. 5
Introduction ............................................................................................................................ 6
  Background ........................................................................................................................... 6
    The Astonishing Tribe and their products ......................................................................... 6
    The Information Process at TAT ...................................................................................... 8
  Limitations ........................................................................................................................... 9
Theory .................................................................................................................................... 9
  Cognitive Load Theory ........................................................................................................ 9
  Instructional Design ............................................................................................................ 10
  The Paradox of the Active User .......................................................................................... 11
  Minimalistic Instruction Design ......................................................................................... 11
Method .................................................................................................................................. 12
Interviews ............................................................................................................................... 12
Grounded Theory ................................................................................................................... 13
Analysis ................................................................................................................................. 14
  Part 1, Information presentation ......................................................................................... 14
    Initial experiences ............................................................................................................ 14
    Workshop ......................................................................................................................... 15
    Training ............................................................................................................................ 18
    Evaluation ......................................................................................................................... 19
    Analysis summary ............................................................................................................ 23
  Part 2, Improvement on information ................................................................................. 23
    1. Initial experiences ......................................................................................................... 24
    2. Workshop .................................................................................................................... 24
    3. Training ....................................................................................................................... 24
    4. Evaluation ................................................................................................................... 25
Discussion .............................................................................................................................. 27
Future research ..................................................................................................................... 29
Conclusions ........................................................................................................................... 29
References .............................................................................................................................. 31
Appendix ............................................................................................................................... 32
Interviews ............................................................................................................................... 32
Abstract
The thesis sets out to answer two fundamental questions about the information given to customers by The Astonishing Tribe; how is this information perceived, and what improvements can be done. The information was gathered by interviewing customers and employees at The Astonishing Tribe, and analyzing this data using grounded theory methodology. The emerging patterns were then taken and examined from an instructional design viewpoint, presenting possibilities of improvement in areas of examples, tutorial material and documentation, drawing on earlier instructional design research.
Preface

From this thesis, I have personally become more interested in instructional design. Who could have thought that it would be so fascinating, thinking about how to best reach out to users with support from technical documentation and educational material?

I would like to give thanks to my thesis supervisor, Fredrik Stjernberg at Linköping University for his help and responses to my questions. I would also like to give thanks to my contacts at TAT – The Astonishing Tribe AB; my supervisor Johan Alexandersson and my thesis client Markus Landin, for their support, encouragement and for letting me do my thesis at TAT. Last but not least, I give my thanks to all the employees at TAT that I have interviewed and disturbed their busy schedules, and the TAT customers that took the time to meet with me and share their thoughts. This thesis wouldn’t have been possible without you all.

After many trips back and forth and around, I’m finally finished.

/Marcus Meurling, 2010
Introduction

Why perform a study about instructional design, something most people find to be utmost boring? For that reason, studying instructional design is important. What can you do with instruction to make it easier to read and understand, and to make the users actually read it? Doing research in instructional design is important because of the huge amount of information people need to deal with daily. The difference between good and bad instructions can have a big consequence in the perception of the actual product, or the efficiency in the work the user might be engaged in. Instructions presented in the right way can give readers more benefits, such as better comprehension, easier learning curve and a more pleasant experience using them.

In the following chapters, information taken from interviews made with customers of TAT –The Astonishing Tribe AB (henceforth abbreviated TAT), and from employees of TAT is analyzed to answer the two different questions that have been the cornerstones of this thesis:

- How is the presented information and material perceived by customers and TAT employees for each step in the information process?
- How can this information be improved from an instructional design viewpoint?

*The information process is defined as the process that starts from the first contact between customer and TAT and goes on to the evaluation and continuous use of the product after sales.

A Non-Disclosure Agreement (NDA) has been signed, so all information presented in this thesis has been agreed by TAT to be published. The NDA has not had much influence on this thesis in form of the presented material herein.

Background

Designing a powerful, fast and flexible framework for user interfaces is great; it gives the customers the freedom to do really interesting things with it, and this in turn affects the end-users of these products, incorporating something that feels great and looks appealing. The potential drawbacks of a framework like this can be that the flexibility makes it hard to learn. To know what your customers think about your product can be invaluable, yet there is no easy and cheap way to do this.

The Astonishing Tribe and their products

TAT is a Swedish software technology and design company offering products and services that differentiate and enhance user experience of portable devices. TAT’s primary focus has been in the mobile phone market, but has in the last couple of years started expanding their products as to include automobile manufacturers and other companies not related to the mobile phone segment. TAT’s software can be found in over 400 million mobile devices.

TAT offers several products and services, but their main product is TAT Cascades, a UI (user interface) framework for the production of advanced user interfaces. Applications are developed using a declarative XML language and the underlying APIs. Cascades is integrated with the platform or operating system, and applications will be built using Cascades to render their UI’s and control their UI logic.
Introduction

Figure 1 - Cascades integration. This picture shows a very high level view of Cascades in the software platform

Usually, more traditional UI frameworks do not allow creation or modification of the user interface without having a major impact on the UI software. TAT Cascades utilizes a design principle that separates the application logic from its appearance (also known as MVC, model view controller, see figure 2 for an overview). By separating the application logic and the appearance, you open up for extensive skinning of your visuals, while the logic needs no modifications. This lets designers change the design of the application, without having to involve the programmers. For older UI frameworks, if you want to change the visuals of an application, the designer will certainly need to include the help from the actual coders of the program. TAT combines this MVC with a declarative programming model that reduces both the amount and the complexity of the code.

Figure 2 - An overview of the MVC model
The information in the picture above (figure 2) will be rendered by Cascades, presenting something like this to the user:

![Figure 3 - The presented information to the user from the MVC in figure 2.]

TAT also offers a XML development environment for Cascades called Motion Lab. Motion Lab is a development environment and visual UI development tool for TAT Cascades. It offers comprehensive XML editing features like automatic completion and validation of elements, tags and attributes, and has the capacity of visual editing.

**The Information Process at TAT**

What is the information process? The process is the different steps that customers are involved in when dealing with TAT and their products, from a first contact to where the customers work on a project of their own with TAT’s products. It can be described as being comprised by the following steps:

**First meeting**

The initial meeting is usually to present TAT and their products where they showcase some demos for the prospective customers. This meeting can for example be initiated by a customer that is interested in TAT’s products. Customers receive information about the product and what it is capable of doing. This also serves as a meeting where the qualification comes into place; is the product good enough for the customer, and is this customer something TAT is interested in. This meeting is for building up a trust of TAT and for the products that they supply, and includes a sales person from TAT, and sometimes a technical consultant. These meetings can be done as a telephone conference, or if the customers are not too far away, as a regular meeting at either the customer or at TATs office.

**Technical meeting**

If the customers are interested in TAT’s products after the first meeting, a second meeting is arranged. This meeting usually involves more people, such as the architects from the customer, and technical sales support from TAT to answer the more technical questions about their products. This meeting is to make sure that TAT’s product can be integrated into the customers product, and the customer is trying to make sure that the product can deliver what they want, and what technical capabilities they will get with it. This meeting is very important in making sure that the product is compatible with what the customer wants it to do. It is focused on presenting the benefits of using TAT’s products, and what they actually get if they buy the product.
Workshop
After the second meeting the normal process is for the customer to get an evaluation license of the product so that they can try the development tools and the product by themselves, if they decide to continue their interest in it. Here they receive a workshop to help them learn how to use TAT’s products in their own product. The workshop is usually in such a way that it explains the basics about the integration and supplies answers to the problems related to their products and how they can be solved.

Training
After a workshop, the customer usually arranges for some kind of training, support and maybe even an on-site TAT consultant. The training customers buy usually vary from two days to two weeks, depending on the size of the project, previous experience of the employees and the project budget. Here they receive basic training in using TAT’s products and how they should structure code and projects. This is more educational where the workshop is more of a discussion and education together.

Evaluation
After the training, the customers are on their own, unless they have hired a TAT consultant. This is when they start using TAT’s products to create their own projects. The most important information that they use in this step is the documentation, example suit, and user guide from the developer site.

Limitations
This thesis has focused on investigating what customers think about TAT’s products, and puts forth a few areas that can be improved, coupled with theories how these changes can be made. This thesis does not however give fully detailed improvements, nor does it present any new instructional material. Time restrictions did not allow for deeper involvement in the material, such as the documentation and the educational material available, instead relying on the support personnel interviews as a base of understanding.

This thesis has only analyzed the information available in the first meetings, workshop, training, and the evaluation phase. It does not consider potential information given outside of this process. Time is also a factor that has made the process more difficult to review.

The limit to instructional design was made because of the otherwise endless possibilities in information presentation.

Theory
The theories I have based my work on are from research done in how people process information, how instructions can be made easier and more scientifically anchored, and how you get users to actually read this information.

Cognitive Load Theory
There are two critical learning mechanisms: schema acquisition and the transfer of learned procedures from controlled to automatic processing. (Sweller J., 1994) Cognitive load theory proposes that since working memory is limited, learners may receive a massive load of information and, if the complexity of their instructional materials is not properly managed, this will result in a
cognitive overload. This cognitive overload impairs schema acquisition, later resulting in a lower performance (Sweller J., 1988). Because of this, cognitive load theorists recommend that learners first study worked examples to promote schema acquisition. This is recommended as opposed to problem-based learning. As they become more proficient, some researchers, for example Kalyuga et al. (2003) (in (Lewis, 2008)) suggest that dropping the examples in favor of partially completed problems and practice solving whole problems to facilitate skill automation. Integrated formats (such as presenting a picture together with text inside it) are superior in instructional material than the split type (presenting a picture, with text at the side), because of the split-attention effect. This split-attention effect means that the reader has to divide his attention between the text and the graphic, thereby introducing a higher cognitive load, which is something that should be prevented. (Morris on, 2007) Instructional material should only include essential information, removing the non-essential information because it will lead to a higher cognitive load. If two sources of information say the same thing, one of them should be removed, otherwise it imposes additional cognitive load. (Cooper, 1990)

One of the important parts of cognitive load theory is the goal-free effect. When faced with a novel problem, learners often use a means-ends analysis to solve a problem, where they determine what is known and try to find the shortest solution. This is often highly efficient, but Sweller suggest that it is not the best approach to learn how to solve problems. This is because the learner need to develop a schema, and the means-ends analysis does not help us understand the relationship between the elements, nor does it help us automate the calculation process. (Morrison, 2007)

**Instructional Design**

Instructional design is the practice of making learning more efficient and effective and to make learning less difficult. Often, well designed instruction saves both time and money. Instructional design is often used to describe a systematic design process for making instruction material, and is based on what we know about learning theories, information technology, systematic analysis, research, and management methods. Training in an organizational setting is defined by the information needed to perform a specific task or related tasks. We can employ instructional design in any context where people are performing a task. (Morrison, 2007)

Operating from theory rather than intuition accomplishes several important purposes. The theory provides guidance regarding how different strategies or manipulations are likely to work. An instructional designer might draw upon behavioral theory to predict that learners who are getting reinforcing feedback to correct responses will increase their motivation to perform well. The designer might also draw on cognitive theories in projecting that personalized word problems will be more readily understood than abstract problems for different learner groups. The theories also give order and consistency to what we do. Subjectively combining instructional strategies into a lesson may work in the short run, but might prove difficult to reproduce when the context changes due to differences in learner ability, subject difficulty, time allotted and so on. The theory provides a blueprint to keep us on course. In this era of “evidence-based practices”, designers may have to justify the theoretical base supporting new instruction. In many of these cases there will be requirements for the instructional practices and methods to be “scientifically based” (Morrison, 2007)
The overview of the design model that Morrison, Ross and Kemp present can be explained with questions for the four fundamental parts of the model:

1. For whom is the program developed? (characteristics of learners or trainees)
2. What do you want the learners or trainees to learn or demonstrate? (objectives)
3. How is the subject content or skill best learned? (instructional strategies)
4. How do you determine the extent to which learning is achieved? (evaluation procedures)

Heinrich et al. (1999) (in Morrison, 2007) suggest that designers initially consider three categories of learner characteristics: general characteristics, specific entry characteristics, and learning styles. General characteristics are the broad identifying variables such as gender, age, work experience, education etc. The specific entry characteristics are the prerequisite skills and attitudes that the learners need to possess to benefit from the training. Learning styles refer to how individuals approach learning tasks, where some find certain methods more appealing than others.

**The Paradox of the Active User**

Designers of reference and help systems count on the users to recognize opportunities to use new methods, but instead users often figure out how to use what they already know to achieve new goals. If they can use the methods they are already comfortable with to achieve the same goal, they are reluctant to explore new functions or search out information. (Carroll & Rosson, 1987)

The first paradox is the "production bias". The paramount goal for a user is throughput. This reduces the motivation to spend any time just learning about the system, and this in turn makes the user less likely to learn a new procedure, even if that procedure could handle a task more efficiently.

The second paradox is the "assimilation bias". This is a cognitive paradox; people apply what they already know to interpret new situations. This paradox can be helpful, when there are useful similarities between the new and old information, but it can blind learners if there are irrelevant or misleading similarities.

According to Carroll (Carroll & Rosson, 1987), these paradoxes are mutually reinforcing, exaggerating the effect either problem might separately have on early and long term learning. Users tend to make use of the functions they know about to get a result, regardless of the efficacy of the method entrained.

Most training and reference material are designed under the assumption that people who need to learn are willing to read about it. This is empirically unsound; learners at every level of experience try to avoid reading. (Carroll & Rosson, 1987)

**Minimalistic Instruction Design**

*The key idea in the minimalist approach is to present the smallest possible obstacle to learners’ efforts, to accommodate, even to exploit, the learning strategies that cause problems for learners using systematic instructional materials. The goals is to let the learner get more out of the training experience by providing less overt training structure. – (Carroll J., 1990, pp. 77-78)*

Often, users respond to the desires to try things out, to get things done. But jumping the gun like this, and relying on exploratory learning strategies instead of the step-by-step rote structure of a manual or on-line tutorial, can be costly. The minimalistic design approach emphasizes the
encouragement and supporting work on realistic tasks from the start and throughout learning: learning by doing rather than learning by reading. Carroll writes that instructions should be absolutely modular; they should make sense in any order.

There are in essence four major principles of minimalist instruction. These include choosing an action oriented approach, where you provide an immediate opportunity to act, and where you encourage and support exploration and innovation. The instruction should always be anchored in the task domain, where you present genuine tasks that feel meaningful to the learners. By including error information in the learning material it supports more flexible and ambitious action-oriented learning. Research has shown that there is significant improvement using a manual with error information designed according to minimalist heuristics than without error information. Error information should be provided that supports detection, diagnosis and correction as close to the erroneous executed actions or methods as possible. A key idea is to not spell out everything; things that are inferred easily can be omitted in the instructional material. (Carroll J. M., 1998)

Minimalistic instructional design is not without risk. Carroll acknowledges a potential problem: “Learners might not have access to enough information to reason successfully and might be anxious about bearing such responsibilities.” The degree of risk depends on how radically information is cut and what is cut and how. This is usually handled by reiterating usability testing on the instructions in theory, but in practice, documentation departments tend to be understaffed and software development schedules are often demanding so that the employees in charge of documentation lack the opportunity to test and retest their documentation. (Carroll J. M., 1998)

Method
This thesis is made from a qualitative viewpoint, where I have used interviews to collect my data, and grounded theory for the analysis.

Interviews
All interviews were conducted as semi-structured qualitative interviews. The semi-structured interview form allows for great flexibility, letting the interviewer pursue interesting sub-topics laid forth by the interviewee.

The persons selected for interviews were based on the availability and the interest of Tat’s customers to participate in a study like this. The total number of customers interviewed was six, and the number of employees at TAT that were interviewed was nine. These were selected from different professions, including managers, designers and programmers from the customers, and manager, sales and support when it comes to the employees at TAT. The customer interviews represent data from two different companies that use TAT’s products. More information about the interviewees can be found in the appendix.

The interviews were done in two sets. The first set, consisting of the employees of TAT and a customer from Company 1, were made at TAT’s office in Malmö during one week. All interviews were booked in advance, and conducted in a conference room. The second set of interviews was conducted in a conference room on-site at Company 2, with all the interviewees being chosen and booked by the project manager at Company 2. All interviewees were informed about their
participation in the study, and of their rights to whenever cancel the interview without explanation. All interviewees were recorded with their consent, and none refused to be recorded. All interviews were done during working hours.

All interviews were recorded with a mobile phone with recording capabilities. Annika Lantz (Lantz, 2007) writes that by recording the interviews and taking notes, you have several advantages over only taking notes. You avoid the almost unavoidable fact that while taking notes, your data is reduced in an unsystematic way. If the interviewer is taking notes during the interview, he or she doesn’t write everything down. This is partly because you don’t have time to do it, and also that you more or less is writing down selective parts of the material. We filter what is said, and this “filter” is made up by our understanding, but also about cultural and personal traits that is deciding our interpretation of the words meaning. This difficulty is more noticeable for interview forms that are more open, where open interviews are the hardest, and structured are the easiest. (Lantz, 2007) The transcriptions of the recorded data were then used in the analysis of the material.

Some of the interviews were in Swedish and some were in English. All Swedish interviews were transcribed in Swedish, and the English interviews transcribed in English. To use as quotes throughout the thesis, the Swedish interviews were translated into English, keeping the translation as close to the original as possible.

Grounded Theory
Kathy Charmaz writes that grounded theory (abbreviated GT) is a set of systematic inductive guidelines for gathering, synthesizing, analyzing, and conceptualizing qualitative data to construct theory (Charmaz K. , 2003). Grounded theory is a very flexible set of methods, and can be used to answer most research questions. Charmaz mean that the foundations of GT as in coding, memo writing and constant comparisons are easy to use in the analysis, even if the researcher chooses not to use GT all the way. The way Charmaz presents grounded theory methodology as tools rather than recipes to follow is a good way to use the methods as building blocks but not necessarily as a rulebook. Grounded theory comprises of three important steps; Gathering rich detailed data, coding and labeling data, and writing memos. (Charmaz K. , 2006)

To gather rich detailed data, we must first know what we want to study, or we might miss important parts or gain too much data for a reasonable analysis. We might also choose a method that won’t capture all aspects of the thing we are studying. For my thesis I want to know what customers think about the information they have been given, and if there is information missing. This led me to the conclusion that semi-structured interviews would best suit my needs.

The coding generates the foundation of the analysis in grounded theory. The coding shapes the analytic frame from which the analysis is built. These codes are different from quantitative codes, where the codes are applied from preconceived categories, whereas grounded theory codes are created by defining what is seen in the data. The logic of coding takes into account what the data is a study of, what the data suggests and from what point of view. The coding should stick closely to the data, and the codes should try to see actions in each segment of data rather than applying preexisting categories. This coding can showcase areas of the material where the data is lacking, and this is part of the analytic process. (Charmaz K. , 2006)
Coding was done on the individual transcriptions, and compared within the groups (customers in one group, TAT employees in one).

Memo writing is when the researcher stops and analyzes the idea about the codes in any and every way that occurs at that time. This will prompt for an analysis of the codes at an early stage of the research process, and helps increase the abstraction of the researcher’s ideas. Codes can also be formed from successive memo writing. (Charmaz K., 2006)

**Analysis**

The analysis will be presented in two parts, one for each of the research questions. The first part will be broken down into several sub-categories, depending on the information process and the type of information involved in each step. The second part of the chapter focuses on presenting improvements that have been compiled from the customer data. Customers are presented as Person 1 to 6, while interviewees at TAT are presented as TAT 1 to 7.

**Part 1, Information presentation**

**Initial experiences**

The initial experiences from the first meetings with TAT are perceived as positive by the customers interviewed. They felt that TAT had good experience in dealing with integration of Cascades into customers frameworks, and provided good support by helping the customers setting up their project team and giving them directions how to begin.

*Person 1(manager): They have good experience how the integration of the products has been made at other companies. Team setup, how to start. [...] felt professional.*

At the technical meeting the audience was bigger, including some lead designers. One of them presents the first meeting from his perspective like this:

*Person 5(usability): [They] came here and they held a two hour presentation that didn’t give that much, I would say, for me, because they didn’t answer any... they showed examples of what you could do and cool stuff you can do but that wasn’t relevant for us, actually.*

The presentation here at the initial meeting felt uninteresting, the examples showcased what you can do with the product, but nothing about how the product would work in their hardware, which this person felt would have been interesting. The questions regarding the specifics of the implementation toward their own platform went unanswered because the meeting was more of a showcase of the products capabilities than directed to their specific goals and needs.

This shows that that the information here might be well suited for some needs, but it does not consider all needs of the different roles of the attendants, at least for this particular meeting.

The purpose of these meetings from TAT is to make the customer interested in the products, and explain the capabilities and benefits of them. These meetings main focus is to sell the product to the customer.
TAT 7(sales): Step one is to build confidence for TAT as a company. Step two is to prove the advantages with the products. What they really get. Step three, how it works technically.

Workshop
These excerpts are taken from several different interviews with employees at two different companies. The specific meeting they attended and their experiences differentiate quite heavily. I will here present a few opinions about the workshop meetings.

The workshop is usually a time for the customers to discuss, together with TAT representatives, both TATs product and their own in detail, to clear doubts about the integration between them.

Person 6(integration): [...] we had like a two week workshop where we discussed various things like clarifying doubts about Cascades, and they clarified doubts about our platform. [...] So I think that workshop was like, the time when we gathered the requirements, basic designs and everything so that we could go back to our respective companies and start working. [...] I think it was very useful, because we were meeting everyone at the same place, and also having the HQ guys over telling us what requirements they wanted from this.

On the workshop they had a chance to discuss more serious issues about the implementation of the product into their own platform. Getting to see everyone and putting all the requirements and basic designs together were very positive for most participants. By getting together to discuss the various issues and solutions, both parties have a good opportunity to learn about the others respective challenges and how to best use and integrate their products. Because this was a big project, the workshop was scheduled for two whole weeks, and this time was needed to be able to discuss the project in detail. It is crucial that the information here is correct and presented in a way that makes the customer feel that the product is well suited for their needs. For Person 1, it was really important to meet the people involved, and having this workshop gave them an opportunity to talk face to face. By understanding the customers’ problems and learning about their platform, the technical experts from TAT can give important feedback and what they think is the best way to integrate the product into the platform. This seems to have been greatly appreciated by the customers, since their knowledge about the product is minimal, but their understanding of their own platform is rich. By explaining their platform to the technicians from TAT they give them a good idea of what their needs are from the product.

Person 6(integration): [...] they [got] an opportunity to learn about our platform and give us input “oh your platform works this way so Cascades can be integrated best in this manner on your platform” so it was a two-way interaction. I think that was essential for the purpose of integrating it into a platform.

The individual experience from the workshops varies from person to person, and has a lot to do with the different roles they have in the product development.
Person 2(programmer): The first thing I was participating in was a workshop; we discussed what challenges we had in [our products]. What we want to do that is difficult in the GUI [*], how we could solve it, that it was possible to solve with TAT’s products. It was mostly selling, it was just funny. [...] we didn’t really get to the implementation details, it was mostly fluff. How you approximately solve it. It wasn’t really my side, it was more about usability.

* Graphical User Interface

This person felt that the information in the workshop wasn’t aimed for them, that it was outside their part of the development. They were more interested in the implementation details, which were not presented in detail at this meeting, which they would have wanted. The experience about the workshop as a whole was positive, but there was a lack of information that was aimed directly at the experienced programmers. They also thought that the workshop was aimed to be very convincing in selling the product, which was perceived as being funny. This is mostly because the information presented and discussed at the workshop were a bit too abstract for the programmers. They did however feel that the workshop was a good way to get a basic understanding about how TAT’s products would work and what they could do. They also had their doubts cleared about the products capabilities and flexibility.

Person 5(usability): [...] before [the workshop] I had prepared some questions about the things I saw as the difficulties with [the product]. And I received very good answers [...] I think it was a very giving workshop those two days. It was a very successful thing.

This person is from the GUI part of the development, and what he thinks about the workshop is somewhat in contrast to the others earlier. He presents a different story, which on the whole is much more positive. The topics discussed at the workshop were clearly more aimed towards his area of expertise, and he found the workshop very giving. Having prepared questions for the workshop in advance shows his interest in getting answers to what he believes are the difficult parts of the integration between the product and their framework.

It is somewhat impossible to present information at a workshop that will be interesting for everyone that is participating. The customer chooses which employees should attend these workshops, and it can be a wide range of people from different areas of expertise, each mostly interested in their own field.

TAT 2(support): It’s usually a very wide spread on the people in a workshop; some are more designers, while some are more technicians, so it’s hard to have a course that works for everything.

Depending on the information that is available about the previous experience of the customers’ engineers, TAT tries to modify the material to better suit the customer needs. If a customer’s employees already possess certain knowledge, then that part can be excluded from the training material and replaced by more productive information.
TAT 4(support): You have to adapt it. There are some areas we think about [when we do it].

The timeframe of the different parts of information was also discussed with the customers. Most people felt that the information available to them in each stage was good. Some felt that there should be a better type of introduction, since they had nothing to start with before they had been formally introduced to the product by the TAT employees. The TAT employees expressed their views of giving customers a lot of self-training information before the workshop and formal training as both positive and negative. The positive sides could be that they already had an understanding for the product, and that they could use the time for a bit more advanced types of training, while the negative aspects could be that they start off in a wrong direction, learning to use the product in an non optimal way, and having to rethink their knowledge when they get formally introduced.

Not everyone thought the timeframe of the different presentations was good; one person said that the workshop should replace the early meetings, so that the technical and integration parts can get discussed earlier.

Person 5(usability): You can say that I would have wanted the workshop much earlier in the process. [...] when deciding to buy, you reason “we want to see the whole product, we want to decide first” [...] I would have thought that doing this workshop in two days is a must to know if this will be good enough for us or not, so I would rather have seen the workshop done then; Already when the people from TAT were here the first time. [They] should have done it then actually. And come up with concrete problems.

This is something only Person 5 expressed, so it’s not very representative, but it feels important to mention since he is one of the few people interviewed that were present at both the early meeting and the workshop. He considers the workshop to be the first real experience you can get with the product, and is therefore very important in the consideration of buying it. He would personally have preferred the workshop replacing the first meeting he was on with TAT, getting the deeper understanding and explanation of the capabilities and difficulties related to their own framework and hardware. This isn’t very representative, because there are very few people involved in the early process, and this is just one person. All interview subjects have been asked when they were first introduced to TAT’s products, and what parts of the information process they participated in.

This also shows that the information presented at the different meetings is directed to different groups of people; the first meetings showcasing the capabilities of the product, while the latter ones, e.g. the workshop, focuses on the implementation with the customers framework and hardware, discussing various issues and how they can be solved. The empirical data here is thin; no concrete conclusions can be drawn about how the information presented to the customers is perceived by other participants of these meetings since they were not interviewed.

A note on workshops
As far as communication goes, an interesting fact has been noted during the interviews and thesis work. The word workshop is used in a very broad sense; it can include anything from a technical meeting, seminar or training session. This in itself might not be a problem, but it can be, if a customer and a TAT representative have different perceptions about what they are talking about when they
use the word workshop. This has been noted with both customers interviewed and with employees at TAT, that use the word workshop to describe presentations, workshops and training sessions.

Training

The amount of time spent on training and workshops vary from projects, but the usual amount is about one to three days for workshops, whereas the training usually ranges from two days to two weeks. People perceive the training they receive very differently, and this can have many reasons. One reason is the personal preference of learning, where some people enjoy lectures and learning through observation and oral presentation, while some prefer the hands-on experience more. The reasons for differences in individual learning styles are many, for example culture (Joy & Kolb, 2009), cognitive differences, (Riding & Sadler-Smith, 1997) etc.

Person 2(programmer): I usually think that it doesn’t give that much to have a lot of training; you have to work with it yourself. There are a lot of people that think it is good with longer training, but I think it was good with the two days we had.

Person 3(programmer): If I were on some training for one week, or two weeks, after that when reading the documentation I could understand more actually.

These two quotes could be explained by the fact that while person 2 means that the training won’t give him much knowledge about how to actually work with these programs, person 3 explains that the training would help him understand the documentation. These two views are not exclusive to each other, but they bring forth different strategies to using the product, bottom-up approach and a top-down approach; The bottom-up approach, learning the documentation and syntax to use the program, and the top-down approach, trying new things and doing examples, and understanding the documentation this way.

Person 1(manager): We did a two day training in workshop, I think that’s the most effective. One focused effort, get people into the thinking, and then just go with it. It’s probably more effective than two weeks in a school bench looking at powerpoints. Get a concentrated, and get to work with hands-on, and have their support, and especially have someone from TAT on site. We have experience with that from before; you have to start working with new technology, preferably with skilled consultants, to spread knowledge. [It’s a] big risk not having an expert from TAT present in a first project.

Most customers incorporate one or more specialists from TAT full time to be part of the first project team or during the first months of development. This is quite normal with a product this complex, and gives them both the expertise needed to start up the project, and on-site help for the new users of the TAT products. This also cuts down time needed for training, since the specialists will be able to help and explain more concrete problems when they are on-site.

Since training is expensive, many customers try to pay for as little training as possible, figuring it won’t give them a good return of investment if they get more training. This has not been evaluated, but it is something that a few people at TAT bring forth as concerns. Giving the customer’s
employees a good base of knowledge takes time, and some things are hard to teach from self-study material and written documentation.

_TAT 2(support): Maybe not enough training and not enough help from us how to build a structure for the projects. Like architecture and best practices and such. So you push them in the right direction more._

One example of something that new learners have to understand to work with TAT’s products is the separation of the data and visuals. When you code in other languages the control logic and the visuals are meshed together, whereas in TAT’s products, they are not. This can make learning problematic, because of the assimilation bias. This is something that TAT is pushing as very important in their workshops, if the customers don’t understand this concept, it can be hard to grasp the basic principles of the product.

_TAT 2(support): The product is built upon a specific architecture. And people don’t really understand it, they can sit around two to three days, then they understand. […] If you don’t understand [the MVC] you don’t understand the basic principle of the product._

**Evaluation**

Several customers found the examples on the developer site lacking a standard answer how to implement a specific feature. What they mean about this standard answer is that they want to know what the best or usual way is to implement this, because with TAT’s product most things can be made in more than one way, because of the flexibility.

_Person 2(programmer): […] you didn’t really get a feeling about what the standard implementation is. That’s something I miss, a standard “this is how you do this”._

The customers refer to the examples and documentation that usually tells of different ways to implement a feature, but it’s not explained what the benefits of the different implementations are or when you should use a specific one. Not knowing when to use a certain implementation makes the programmers uncertain, and if you chose the wrong implementation you might have to redo your work later on with another implementation. The customers talk about how the lack of a standard implementation might be because the product is very flexible, which in a way is a very positive thing, but as the same time it makes the product more complex. If you don’t know which control to use, you may have to try them to see which one that works best.

_Person 5(usability): It’s hard to know “how do I solve this, should I use this type of control or should I use this [other] type of control”. […] I basically had to go through testing all the controls._

_Person 3(programmer): If the documentation could say which control to be used in which scenario, it would be more helpful._

This problem is something that is acknowledged by employees at TAT, and something they try to help customers with. Because of the flexibility of the product, you can go about a problem from different angles, but there is usually one or a few ways that will yield the best performance for a specific task.
The lack of a proper tutorial was something that was brought up by several people. They felt that the examples did a good job in teaching them how to implement a specific feature, but the fact that they weren’t linked together made the progression hard when trying to learn the product.

Person 2(programmer): There is no real tutorial, which they could put up. They have examples, but they are not connected. It isn’t easy to penetrate.

This is something that is mirrored by many employees at TAT, that the lack of tutorial and good practice examples can make the initial learning difficult.

TAT 2(support): That’s something I’ve been asking about, if I had been new to Cascades, I would have wanted a step by step tutorial, or exercises combined with film.

TAT 4(support): [...] some kind of tutorial series that goes from setting up your project in Visual Studio until you have a user interface for [a mobile platform].

The documentation as a whole is viewed as pretty good, but could use some more elaborate explanations for the controls.

Person 3(programmer): Simply more than listing the attributes and operations and even handlers. It should define the actual role of that control and the possibilities, problems and limitations, and put in numbers what you can do with that particular control.

This is also related to the earlier part about the best use practices, which control that is best suited for certain things. Customers also ask for more explanations how modules interact with each other, instead of just descriptions of the modules.

In this study, most of the customers interviewed say that they use the documentation before they send support requests to TAT, but this is something that many employees at TAT feel isn’t the case. This could very well be because of the small sample size, where the customer narratives that this thesis has used are all in minority. The reason why the employees feel that customers aren’t checking the documentation before sending support questions can be because of the fact that customers actually can’t find the information they are looking for, or that they aren’t really trying.

TAT 6(support): Most customers aren’t checking [the documentation], you have to point to it in that case.

TAT 5(support): They are pretty bad at using it. They like pushing the problems onto us. You don’t know if they have checked the documentation [before contacting us].

This information, although not supported by the customers interviewed, supports the paradox of the active user theory.
Examples
Most customers agree that the examples are what they use the most, and what they can learn most from. Having a good selection of examples in varying difficulties and complexity give customers a good insight in how different controls can be integrated. They feel that the current examples are good, but they would want more of them.

\textit{Person 4(programmer): The examples would be most important to expand. If you put more complex examples inside and cover more areas, it would be helpful even if the documentation was not so detailed.}

\textit{Person 5(usability): Better and more detailed examples, coupled with explanations. One thing that was good was code that was written with explaining text. “Now we do like this because…”}

When you have examples that are coupled with explaining text, and with the code commented, it’s easier for the customers to see what certain parts of the code does, and how it can work together with other parts. The most requested feature from all the customers interviewed are to be given more examples, especially more complex ones where you can learn how certain parts of the code interact with other parts. Most employees at TAT also express the need to develop more examples, especially the more complex and detailed ones.

\textit{TAT 5(support): But I believe examples are the most important. Especially many commented and detailed examples.}

There are however some employees at TAT that have expressed a mild concern about using examples too much, because customers can develop a habit of copying code from examples, not trying to understand what the code actually does or how it integrates with other code. This of course is up to the customers’ engineers, if they value results or actual understanding. By copy pasting too much code without proper understanding about the actual controls, you risk running into problems later on, problems which you may not be able to solve without doing extensive changes. By understanding what you are doing and working on, you have a better chance of doing great implementations and optimized code. There have been times when code has been sent to TAT from a customer that consists mostly of code from examples that has been put together.

\textit{TAT 5(support): You can place blame that it’s not so good to copy paste too much, because if you do that you don’t understand what you have copied.}

This is a minor concern, and even those that expressed it also recognize the value of learning by examples, it all depends on if the person using the material wants to learn or not, and more often than not, they do. For example, earlier studies in this field shows that studying examples is the most cognitive efficient method of instruction (Lewis, 2008)

Examples are often regarded as educational tools, and many users will actively use the examples to learn from. Customers wanted easier overview of what every example contained, and what you could learn from them. This is something that could make the examples easier to use and at the same time provide a better learning experience.
Person 3(programmer): The examples suite needs to have more description about what the example has. All it has is one to two line description and the source code. It would be very good if you could pinpoint what every example has and what you can learn from them.

Error diagnostics
Because of the way the XML is built up and the fact that users don’t have access to the source code of Cascades, users cannot employ debugging in the most common way. This is something customers miss having the option to do, since it’s easier to troubleshoot your code if you have the option of debugging.

Person 4(programmer): Something I really miss is something about the debug. If you have made a mistake you don’t know where this mistake is.

Person 3(programmer): It would be good if we could get more debugging features in Motion Lab. Right now in most of the cases it’s not specific.

Search
Customers find the search function on the developer site to be hard to use effectively. The reasons they put forth is the different documents you can search, and the fact that it searches on keywords, which either gives too many irrelevant hits or is hard to use if you don’t know the exact keyword to search with.

Person 3(programmer): In most of the cases you have to search with the exact control name [...] Now the search engine simply searches for the keyword in all document items. [...] it should be more relevant to which that word we are searching.

The search function is in itself not something that can be improved upon from the viewpoint of instructional design, other than that it should be easier to use. Since it’s already hard to get users to actually work with material and documentation, having a good search function is the only way to ensure that they at least try to find the help themselves. This is something that several people working on TAT also want to see improvement upon. Customers that can’t find the information ends up asking them instead, so that they have to search the documentation and maybe ends up sending back a link to the documentation (or a personalized explanation).

Several customers said that one frustrating part of programming with Cascades was that they were unable to use their favorite tool when programming; search engines. The fact that the coding language is closed makes it impossible to get help by utilizing a search engine, and this is something that these programmers are used to while programming in other languages. It is much more difficult looking up information in documentation than searching for a solution, since you don’t need the specific programming keywords while searching using a search engine, just a problem statement or a few keywords about what you want to accomplish.

Person 2(programmer): [...] that’s how I program in other languages, I google program. “Oh this doesn’t work” and then you search and find someone that has the exact same problem.
Person 3 (programmer): The problem is, if you are working on some other technology, if you are searching google you can get any information you want. The problem arises mainly from TAT.

Analysis summary

1. Initial experiences; the manager interviewed was impressed by the meeting, the designer was not. This shows that the information presented were right for some people but not everyone.
2. Workshop; the workshop seems to have good information, although not suited for everyone. Most people interviewed regarded the workshop as a very important part.
3. Training; most people seem to enjoy shorter training periods. Where this is a matter of money, most customers choose to employ a consultant rather than have extensive training sessions. This gives them fast access to help when they need it.
4. Evaluation; there is a need for examples; most programmers generally request more examples, and specifically more complex ones where different controls interact with each other. They also want best practices, since there are many ways to incorporate different modules and controls. Customers would also want an expansion of the documentation, a better search function and some more help with error diagnostics.

Part 2, Improvement on information

How can the information be improved? First, the parts that can be improved through instructional design have to be identified.

Following the method of needs assessment from Morrison’s book Designing Effective Instruction, there are four phases to work through. These phases have all been done in this thesis, and they are:

Phase 1: Planning; choosing a target audience, data collection and participants.
Phase 2: Collecting Data; choosing sample size and schedule the data collection.
Phase 3: Data Analysis; Analyze and prioritize the data.
Phase 4: Final Report; The inclusion of the purpose, the process, the results and the recommendations for improvements.

The summary of identified needs from the empirical material is:

1. Initial experiences; the empirical data is too thin to draw any conclusions, since the empirical material consists of only one person for each meeting. It seems that customers get adequate information from the data I have collected.
2. Workshop; the workshop seems to have good information, although not suited for everyone.
3. Training; the basic training customers take seems to be a good base of knowledge.
   Tutorial; a tutorial for new learners, to give them a better introduction to the program
4. Evaluation;
   I. Examples;
      i. More complex examples; the need to understand the connections between several controls and how they interact with each other.
      ii. The examples should include best practices, what the benefits are for a certain solution, and if possible, present problems with other solutions.
II. Documentation; expansion of the documentation. The documentation is sometimes perceived as being too shallow in its explanations.

III. Search; a need for a better search function. Customers and employees at TAT agree that the information is hard to reach by searching, either because you need to know the exact control name or because you get too many irrelevant hits.

IV. Error diagnostics; providing error diagnostics, prevention and/or recognition; much time is used when troubleshooting problems, time can be saved by providing better support of these.

Each of these will be discussed in this part of the thesis, but not all of them are solvable with instructional design, which has been the main focus.

There are a few preinstructional strategies that can be used when presenting instructional material to users. The ones most applicable in this setting would be the traditional paragraph introduction (the overview). The overview focuses on presenting the summary of the content, or poses a problem the instruction will help the user solve. Explicit signals in forms of lists, comparisons, temporal sequences, causes, and definitions can help manipulate the text to provide signals to important points in the text. Typographical signals are used to signal the structure of the text, with headings as key words or phrases to identify the content of the sections of text information. Illustrations are particularly helpful when used to show spatial relationships described in the text. (Morrison, 2007) Something that is important to think about is to maintain a consistency in the instruction material.

1. Initial experiences
Because the empirical data is thin and the perceptions are vague, there aren’t many conclusions that can be drawn about the information at the initial meetings. Some persons interviewed find it satisfactory, while some would have wanted more concrete facts and earlier information about the integration of the product with their own.

2. Workshop
The workshop is something that might seem like an impossible task to please everyone that is participating, and it probably is. What is apparent is that the workshop seems to be a very positive experience for a majority of the customers that are participating, and this should be seen as a success.

3. Training
The training is perceived as good, but different learning styles makes customers differ in how much training they actually want. Since most employ a TAT consultant, the amount of training needed is reduced. There might be a need to supply self-study material for training in form of tutorials. The fact that customers try to avoid paying for additional training makes the need for tutorials greater.

Tutorial
Tutorials often begin with an explanation of how the application and instruction work, or an orientation to the semantics of the domain. These explanations are valuable, but positioned at the very beginning of the manual they constitute a distraction. The learner is confronted with prerequisites to action instead of the opportunity to act. An alternative is to present a simple but realistic activity that conveys some of the explanatory content, and then follow this with the balance of explanation. The learner still gets the explanation, but after having been given the chance to
experience an activity. (Carroll J. M., 1998) A tutorial should let the learners check their work against a solution. Comparison should be encouraged, but as long as the result is valid there can be many correct solutions. Users should be given advice on the best practices in solving the exercises. Giving users ideas about best practices is important, if there is doubt about how something should be implemented. This will both help users distinguish which control suits a certain need better, and give them knowledge of how they should be implemented in the best way.

The tutorial can be created from some of the examples from the examples suite, so as to use the material already done and not develop entirely new material for the tutorial. If the examples are plenty and can be presented to the learner in a good difficulty progression together with explaining text, this would be a good way to start a tutorial. Creating a tutorial that complies with the minimalistic instruction design and cognitive load theory would be a very powerful introduction to learning Cascades. The important parts for this are to let users work with material that they feel they will benefit from learning, with material anchored in their task domain, and that the instructions are brief but well aimed. There is no reason to have a lot of explaining text; users won’t be willing to read through it. Using for example the integrated format of pictures and text, you can get the users interested easier and at the same time lower the cognitive load they are exposed to.

4. Evaluation
In the evaluation process, the most important parts of information for the customers are the examples and documentation, and getting proper feedback when errors occur.

I. Examples
People trying to learn skills are eager to act, this eagerness is the source of the production paradox mentioned earlier. When research has shown that learning by studying examples is the most cognitive efficient way of learning, this seems like a good area to improve upon. Since minimalistic instruction is always action oriented, a good way to implement this is from examples, where you encourage the learner to work actively with the program. A suggestion would be to create the examples from a view of modularity, so that they can both be used as stand-alone examples, but also be included in a learning tutorial, where you focus on a linear progression in difficulty. These will then range from simple beginner examples, to the most complex ones where several functions interact with each other.

The examples should always be anchored in the task domain, where the user can feel that what he is doing or learning in the example is applicable directly to his own goals. Users will recognize an activity as genuine only if they have adequate prior experience in the task domain. To capitalize most efficiently on the users' interests in and knowledge about the task domain, instructions must build on prior skills, knowledge, and experience by recruiting the scenarios and procedures of the task domain that are familiar to the users. (Carroll J. M., 1998)

When creating these examples, the headings of the instructional tasks should convey the major procedural elements, so that the users can get an overview of the different steps of the example. The text should be fairly brief, and used to invite people to explore on their own. A critical aspect of prompts is how they are formulated. Expressions such as "See what happens" and "Try and see for yourself" lead to more user explorations than neutral statements like "You can also" etc.
When discussing examples, it’s hard not to view them as educational tools. Giving voice to the examples with instructional text will facilitate the learning process from a cognitive viewpoint. These worked examples show each step in the problem solving process. A learner studies the problem by working through each step of the example. Then, similar examples can be presented for practice. Studies in this area have shown that the worked examples are more effective if the prompts are faded. (Morrison, 2007) Using worked examples to facilitate knowledge puts a lower cognitive load on the learner than the usual means-ends analysis that is used for problem solving. (Cooper, 1990)

Examples are useful for expert users and novices alike, but with a few major differences. The experts are usually not interested in any step-by-step introductions, but would rather have just enough information to get started in the right direction. A good use of layering could bring together a version that works for both expert users and novices. Layering is when you make use of the interactive capabilities of online documentation, where you can hide non-essential information or more explaining steps, so you can get a briefer and less intimidating documentation, while at the same time letting the user access this information if they so wish. Layering can be accomplished with the use of tabs, buttons, pop-ups, hover text, etc. The instruction for expert users should assume prior knowledge, take advantage of the ability to skim for salient information and emphasize difficult tasks while assuming that simpler ones will take care of themselves. (Carroll J. M., 1998)

When creating educational tools, such as examples, tutorials, guides or other means, these must be evaluated and the design process iterated to make improvements in the material. There are only generalized concepts on how to create instructional design, there is still a need to evaluate the material created, and make changes that will suit the targeted audience.

**II. Documentation**

Users want good documentation to be available when problems occur, but at the same time they perceive it as unhelpful because it is time-consuming and difficult to find the right information, if you don’t know exactly what you are looking for. (Carroll J. M., 1998) Making the documentation easy to use and with coverage of all functions is the only way to ensure that users actively use the documentation.

Language difficulties could also create a problem with documentation. With international customers, the documentation has to be advanced enough to explain everything in an adequate way, but at the same time be easy to understand. Documentation needs to be thorough in its explanations, but that doesn’t necessarily mean that needs to be a lot of text. The important parts can be summarized, and prior knowledge can be assumed in certain areas.

A good use of layering of information can be used in technical documentation. You could supply information about the suitability for certain commands or controls, as a sort of best practice. This information could either be brief, or layered. Information about how controls interact with each other could also be layered in the documentation, linking to an example that showcases this interaction.

Not being able to use search engines on the internet to find information makes the need for easily accessible documentation very important. Since many programmers today uses search engines when they encounter a problem, as the interviews have shown. Having to give up this opportunity of
information gathering and relying on a single documentation for a product can be perceived as difficult by many programmers.

Users should be able to post comments on the documentation (on examples and tutorials also). By allowing this you can gather information about what parts users find difficult, and improve upon these. The user will be more encouraged to give feedback if it is easy to do and won’t interrupt their work. This will also make it easier for the people handling the documentation to get feedback from customers and evaluate this, making changes if necessary.

**III. Search**
The search function is important to let the user access information in a convenient way. This is not something that instructional design can solve, but should be done with “smart” algorithms and implementations. For example, you should be able to search for the functionality you are after, while not having to supply the correct function name, and you should get hits based on relevance and not only on the supplied search term.

**IV. Error diagnostics**
Error diagnostics is an important tool to figure out what’s wrong and how to correct it. This is something that TAT is already working on at the moment, and will be released in the next major update of the software.

Earlier research in the field of instructional design has shown that learners spend between 25-50% of their time correcting errors. Reducing mistakes and streamlining detection, diagnosis, and recovery may substantially reduce learning problems. In general it is desirable to include error information at a rate of about once every three actions. The frequent inclusion of this information provides a safety net, supports more flexible and ambitious action-oriented learning, and helps deepen the users’ understanding of how the program works. Research has shown that there are significant improvements using a manual with error information designed according to minimalist heuristics, than using one without error information. (Carroll J. M., 1998)

Mistakes can sometimes be prevented by including hints in the manual. Certain errors can be prevented in the development tool itself. Error information should be presented as close to the wrongly executed actions or methods as possible.

**Discussion**
The design of instruction seems to be important to think about when creating educational and instruction material. This thesis has presented a few ways to improve the design of the current material at TAT, and also shows how important customer feedback can be. By asking the customers, you can more clearly see if there are needs that go unfulfilled, and if there are parts where better instruction can help these them.

When designing instructional material, it’s important to ground them in relevant theories that will promote learning and understanding, such as the cognitive load theory. By adhering to this theory while designing the instructions, the benefits will for example include better schema acquisition. Using the goal-free effect and letting the learners use worked examples has been shown to promote learning and reduce the cognitive load on the learners. If you create instructional material without
coupling it to relevant theories, the chance of having created good material will be significantly lower. Interesting work has been done in cognitive load theory by Pollock, Chandler, and Sweller (2002) (in Lewis, 2008) where they showed that it was better to split up instruction into smaller parts, thereby reducing the cognitive load of the overall lesson. When doing this, they say that this promotes learning at the expense of understanding, by which they mean that the learners had a better chance of actually processing the instruction but that they wouldn’t understand the entire lesson. They later showed that the learners wouldn’t understand the lesson anyway without the split, so they concluded that it was better if they learned each part separately, to later combine these sub-schemas into understanding the whole problem.

Minimalistic instruction design shows many similarities with cognitive load theory in the organization of the material, the reduction of redundant information, and in some ways the goal-free effect. Earlier research has shown benefits of using instructions created from a minimalist perspective than instructions that regular instruction material (Carroll J. M., 1998). The reasons to use a minimalistic approach to instruction is the lower need for extensive instruction, that the instructions are easy to begin with since they encourage exploration and use directly, which will make the users more eager to actually read them, and by supplying error correction the learner will have an easier time to get back on track if he made any mistakes.

An interesting fact that was noted during the interviews was that the terminology of the word workshop is used in a very broad sense. This might make discussions about workshops a minor hassle since not everyone have the same idea of what they actually mean with the word workshop. At a few occasions during my interviews I had to ask if it really was the workshop they were talking about, when it sometimes was the training they had been on. The word is used for many things, and sometimes a workshop and a training course is put together, making it even more confusing. This generally should have no implications for the customers, but is something that might be worth noting if you are collecting information from customers about these kinds of meetings.

Something that is interesting regarding international customers is the fact that there are learning differences in play that might be seen but not understood. Some learning differences can be related to culture, where S. Joy and D. A. Kolb shows that there is a significant effect of culture in the decision of a person’s preference for abstract conceptualization versus concrete experience, but that the effect of culture is marginal on the preference between active experimentation and reflective observation. (Joy & Kolb, 2009) There are also learning differences in cognitive learning styles, which can be attributed to different cognitive abilities. Riding and Sadler-Smith talks about how there are two dimensions of cognitive style, the analytic-wholistic and the verbalizer-imager (Riding & Sadler-Smith, 1997). There can also be language difficulties, but this is not brought up a lot in the interviews. Considerations about the implications of language difficulties should be considered though.

Because of the relative small sample size (six customer interviews and nine TAT interviews), there isn’t enough empirical data to make statements about the earlier parts of the information process, and without proper knowledge about marketing and the technical capabilities of the product, there can be no suggestions of improvement either. The small sample size also makes the material subject to critique for not being able to draw generalizations about all customers of TAT; this material can only be used to explain a small part of TAT’s customers perceptions. Gaps in the data were noted
when doing the GT analysis, but were not able to be recovered in some areas because of the lack of more informants in the respective areas.

This study does not include a documentation analysis based on the grounds that it would be too time consuming, and that the information from the customers and the TAT employees would suffice as a base. It would be easier to present concrete improvements of the material if a documentation analysis were done, but this is also true about the ability to construct real instructional material if there had been more time to do so. The scope of this thesis isn’t wide enough to include either of these.

Future research
To make the material more complete, further data would be needed, interviewing customers from other demographic locations, for example Korea and the United States, to get more conclusive results. Getting more interviews with customers that have participated in the early parts of the information process would also help to give a more complete picture of what the customers really think about it. It would also be interesting to interview the people that are developing Cascades, the core product, to get their viewpoint on the information process and documentation.

Conclusions
Considering the analysis done on the material, a pretty coherent picture about the information process takes form. It appears that the customers and the employees at TAT mention the same problems that exist today. The information is perceived as good, although not enough to be considered adequate in some cases. There are a few areas that most feel need improvement, for example the self-study material and the documentation. The first meetings and the workshop seems to have a good effect on the participants, and most are very positive about these. During the evaluation, most customers choose to employ one or more TAT consultants to work in the customer’s project, to spread knowledge and help with the structure. This was perceived by the customers as fairly common in this type of business.

Drawing upon research in instructional design and cognitive load theory, I present some possible improvements that can be made on the current information. This focuses mainly on the educational material, examples, and documentation, leaving the information presented at for example meetings and workshops. The improvements on tutorial material includes a few thoughts from minimalism instruction, for example: less text, material that feels anchored in the task domain and not artificial, best practice information about implementations, and some easy way to check their work against a solution. The improvements presented on the examples are: More explanations about the specific example, what you can learn from it and what it does. The examples should be created out of modularity, so that each example can be used with as little prior knowledge as possible. This will not be possible with the more complex examples, which should include complex interactions between controls and modules. The use of layering is beneficial in most situations, where you can hide text that might not be needed, but that some users might find valuable. The presented improvements on the documentation are: Generally better explanations on some features, and to easily locate the areas that customers feel that the information needs expansion. For this there can be an included comment function on the developer page. This gives the customers easy access to commenting on the documentation without having to go through other mediums and interrupting their current work.
The implications of bad instructional design have led many users of instructional material to perceive it as useless, hard to understand, or that it takes too long to read through. The important thing with instructional design is to get the user interested in learning whatever the instruction is about. These instructions will always compete against the paradoxes of the active user; the assimilation bias and the production bias. Good instructions will be perceived by the user as efficient, relatable and holds a high educational value. Instructions will never be perfect upon creation, it is an iterative process where you have to evaluate the effects of the instructions and make adjustments where needed.

When technology progresses and products become harder to use, there is an increasing need for good instructional design; instructions that you can read and understand, that are designed to provide more efficient learning and be less difficult to go through. The benefits of creating instructions from reliable theories of instructional design and cognitive load are many, compared to creating instructions based on personal ideas on how learning works.
References


Appendix

Interviews

Here is a brief summary of the interviews conducted for this thesis.

<table>
<thead>
<tr>
<th>Person:</th>
<th>Customer 1</th>
<th>Customer 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Project Manager</td>
<td>Programmer</td>
</tr>
<tr>
<td>Length:</td>
<td>About 55 min</td>
<td>About 35 min</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person:</th>
<th>Customer 3</th>
<th>Customer 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Programmer</td>
<td>Thesis worker (Programmer)</td>
</tr>
<tr>
<td>Length:</td>
<td>About 45 min</td>
<td>About 35 min</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person:</th>
<th>Customer 5</th>
<th>Customer 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Lead Interaction Designer</td>
<td>Lead for framework integration</td>
</tr>
<tr>
<td>Length:</td>
<td>About 40 min</td>
<td>About 35 min</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person:</th>
<th>TAT 1</th>
<th>TAT 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Product owner</td>
<td>Software developer (support)</td>
</tr>
<tr>
<td>Length:</td>
<td>About 60 min</td>
<td>About 60 min</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person:</th>
<th>TAT 3</th>
<th>TAT 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Software developer (support)</td>
<td>Software developer/education (support)</td>
</tr>
<tr>
<td>Length:</td>
<td>About 55 min</td>
<td>About 45 min</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person:</th>
<th>TAT 5</th>
<th>TAT 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Software developer (support)</td>
<td>Software developer (support)</td>
</tr>
<tr>
<td>Length:</td>
<td>About 60 min</td>
<td>About 35 min</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person:</th>
<th>TAT 7</th>
<th>TAT 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Sales</td>
<td>Sales</td>
</tr>
<tr>
<td>Length:</td>
<td>About 35 min (complemented twice 5min each)</td>
<td>About 45 min</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person:</th>
<th>TAT 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Software developer (support)</td>
</tr>
<tr>
<td>Length:</td>
<td>About 45 min</td>
</tr>
</tbody>
</table>