Developing digital competence - learning, teaching and supporting use of information technology

By Jens Kaasbøll, University of Oslo

Table of contents

Chapter 1. Introduction .......................................................................................................... 5
  1.1. Motivation and target groups ....................................................................................... 5
  1.2. Aims ............................................................................................................................. 5
  1.3. Related areas ................................................................................................................ 5

Chapter 2. IT skills ................................................................................................................. 7
  2.1. Learning IT skills ......................................................................................................... 7
  2.2. Instructions sheets ........................................................................................................ 9
  2.3. Instruction videos ...................................................................................................... 13
  2.4. Training for skills ...................................................................................................... 15
  2.5. Assessing IT skills .................................................................................................... 16
  2.6. Summary .................................................................................................................... 16

Exercises ............................................................................................................................... 17

Chapter 3. Subject matter areas ........................................................................................ 21

Chapter 4. Understanding IT ................................................................................................ 31
  4.1. IT concepts ................................................................................................................ 31
  4.2. Learning IT concepts ................................................................................................. 37
  4.3. Functional models ...................................................................................................... 41
  4.4. Conceptual models .................................................................................................... 42
  4.5. Relationships between concepts ................................................................................ 47
  4.6. Videos providing functional and conceptual models ................................................ 50

© Jens Kaasbøll, 13th April, 2012
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7.</td>
<td>Learning failures</td>
<td>51</td>
</tr>
<tr>
<td>4.8.</td>
<td>Summary</td>
<td>55</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Understanding information</td>
<td>61</td>
</tr>
<tr>
<td>5.1.</td>
<td>Representation competence</td>
<td>61</td>
</tr>
<tr>
<td>5.2.</td>
<td>Correspondence competence</td>
<td>62</td>
</tr>
<tr>
<td>5.3.</td>
<td>A representation system – slide design</td>
<td>66</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>Understanding task support</td>
<td>72</td>
</tr>
<tr>
<td>6.1.</td>
<td>Levels of mastery of using computers in work tasks</td>
<td>72</td>
</tr>
<tr>
<td>6.2.</td>
<td>Usefulness</td>
<td>74</td>
</tr>
<tr>
<td>6.3.</td>
<td>The organisational level</td>
<td>76</td>
</tr>
<tr>
<td>6.4.</td>
<td>Summary</td>
<td>77</td>
</tr>
<tr>
<td>6.5.</td>
<td>Project</td>
<td>78</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Training for transfer</td>
<td>79</td>
</tr>
<tr>
<td>7.1.</td>
<td>Conceptual-practical training</td>
<td>80</td>
</tr>
<tr>
<td>7.2.</td>
<td>Learner activity</td>
<td>88</td>
</tr>
<tr>
<td>7.3.</td>
<td>Summary</td>
<td>89</td>
</tr>
<tr>
<td>Chapter 8</td>
<td>Learning problem solving</td>
<td>91</td>
</tr>
<tr>
<td>8.1.</td>
<td>Competence for learning</td>
<td>91</td>
</tr>
<tr>
<td>8.2.</td>
<td>Navigation: the fourth basic way of learning IT use</td>
<td>92</td>
</tr>
<tr>
<td>8.3.</td>
<td>Exploration</td>
<td>94</td>
</tr>
<tr>
<td>8.4.</td>
<td>Problem solving</td>
<td>96</td>
</tr>
<tr>
<td>8.5.</td>
<td>Fostering metacognitive competence</td>
<td>97</td>
</tr>
<tr>
<td>8.6.</td>
<td>Summary</td>
<td>100</td>
</tr>
<tr>
<td>Chapter 9</td>
<td>Evaluation of training</td>
<td>104</td>
</tr>
<tr>
<td>9.1.</td>
<td>Evaluation of reaction to training</td>
<td>105</td>
</tr>
<tr>
<td>9.2.</td>
<td>Evaluation of learning – assessing competence</td>
<td>106</td>
</tr>
</tbody>
</table>
9.3. Evaluation of behavioral change ................................................................. 109
9.4. Evaluation of result or outcome ................................................................. 110

Chapter 10. IT user competence standards ....................................................... 112
10.1. Standards and guidelines ....................................................................... 112
10.2. Tests ....................................................................................................... 113
10.3. IT competence levels ............................................................................. 118

Chapter 11. Super-users .................................................................................... 123
11.1. Roles ..................................................................................................... 123
11.2. Trainers ................................................................................................ 129
11.3. Organising training ............................................................................... 130
11.4. Summary ............................................................................................... 130
11.5. Super-users as leaders ......................................................................... 131

Chapter 12. IT support .................................................................................... 132
12.1. Support as boundary interaction ........................................................... 132
12.2. IT support versus super-users ............................................................... 134
12.3. Support quality ..................................................................................... 134
12.4. IT departments ..................................................................................... 136
Ten golden rules for improving IT users’ competence

1. Provide users with detailed instruction sheets or videos, also during training.
2. Train users so that they understand IT concepts.
3. Provide a variety of learning material.
4. Make sure users understand the usefulness of the IT.
5. Organise training at the same time as the system is installed.
6. Train users so that they can solve problems and learn on their own.
7. Identify, organise, authorise and cultivate superusers.
8. Include IT, information and use competence in support and training.
9. Provide a variety of support channels and frequency.
10. Train local groups of users, not only individuals.
Chapter 1. Introduction

1.1. Motivation and target groups

1.2. Aims

1.3. Related areas

informing users about introduction of new IT in advance will help user acceptance (Bondarouk, 2006).

© Jens Kaasbøll, 13th April, 2012
Part I. IT skills and learning

The learning aim of Part I is:

To be able to design learning activities and material for IT users such that they learn IT skills.
Chapter 2. IT skills

Users of technology need skills for applying it to meet their needs. A skill is a practical competence, indicating that a skilled person knows how to do it. Therefore skills are also called know-how. IT know-how will normally include some bodily skills like hand-eye coordination for mouse movement and the ability to push the keys on the phone with the thumb when writing a text message. The cognitive component of IT skill concerns the ability to utilise the functionality of the technology through selecting operations and entering data.

Understanding is complementary to skills, and it includes knowing why mechanisms work like they do or knowing whom to deal with. Understanding is also called theoretical competence, know-why or textbook knowledge, since it can be learnt from reading books.

Being able to enter numbers and formulas in a spreadsheet is an IT skill, since it involves doing things on the computer. The ability to tell that a spreadsheet is useful for calculations is not a skill, since telling does not involve the doing. The ability to tell about something demonstrates an understanding. Although a user who has spreadsheet skills often also has spreadsheet understanding, there is no automatic relation between the two. One person can be very skilled without much ability to tell about it, and another may have a profound understanding of IT, but being poor at using it.

IT users need IT skills, without which they would not be users. IT skills therefore constitute the main learning aim of IT training and material for learning use. While skills are fundamental, more advanced users excel in understanding IT and in problem solving and being able to help others. These more advanced levels of IT user competence will be addressed in the subsequent chapters.

2.1. Learning IT skills

Learning is development of competence that provides a relatively stable change of behaviour. As an example, Roberto used to carry out complex calculations on his phone. After having discovered spreadsheets, he uses these instead. Since his change of behaviour is lasting, Roberto has learnt a new IT skill. Roberto also tried setting up a relational database for managing is accounts, but he reverted to spreadsheets. Since he did not stick to this habit, learning might not have taken place.

Repetition

Repetition is a learning process for strengthening an already existing skill. Repeating behaviour many times normally leads to the learner being able to do it without conscious awareness. We say that the skill has become automated. Walking, running and biking are examples of skills which people would learn so that they are automated. We normally don’t pay attention to how to move the leg forward when walking. Correspondingly, typing on a
keyboard becomes automated after long practice, so that we can write words without considering which fingers to move in order to hit particular buttons.

This book will not address practicing the bodily skills like pushing the buttons on a QWERTY keyboard. The interested reader can find specific textbooks on such skills (Barnes, 1890). Rather, we will concentrate on the cognitive component, the know-how of operating software and IT gadgets.

**Imitation and instructions**

While one can become an efficient user of IT by repeating a sequence of operations, repetition does not necessarily extend our repertoire of skills. A way of learning new skills is *imitating* others’ behaviour.

Assume that Ali is watching Gabriela while she is pressing CTRL/C and CTRL/V for copy and paste. Ali has been doing this operation by means of menu choices quite often, but he has not seen the shortcut before. He sees that it is time efficient and therefore starts using the key combination thereafter. When Ali continues to use the key combination instead of the menus, a relatively stable change of his competence has taken place, meaning that he has learnt this skill.

It has for long been recognised that imitating others is an important way of learning IT use (Bannon, 1986). In general, similarity between the settings during imitation and repeating ease learning (Ormrod, 1995). Imitation has therefore a strong advantage as a trigger for learning, since what the learner observes is exactly the behaviour to be repeated.

Imitating others is not always successful for learning. Assume that Ali went back to his work after observing Gabriela pushing CTRL/S for Save as, and that he pushed the shortcut keys the first time he had wanted to save a file under another name. When having to repeat it again the next day, he had forgotten which keys to press. Then his change of behaviour is not stable, so no learning took place.

Ali was motivated to learn the copy and paste shortcut through observing that it saved time. Motivation is a strong factor for learning. If Ali is happy with his menu choice for copy and paste and does not see any advantage in keystrokes, he will not bother trying, and hence he will not learn it.

People’s ability to remember many details after one viewing one demonstration is very limited. Pushing two keys where the first key is the first letter of Copy and the second is the next key to the right was manageable for Ali. Our short term memory is limited to a few elements, so people will normally not remember a sequence of 10 operations after a demo.

---

1 In a behaviouristic learning literature, this way of learning is called ‘modeling,’ ORMROD, J. E. 1995. *Human Learning*, Englewood Cliffs, New Jersey, Merrill., but since the word ‘model’ is used for other purposes within IT, we use the term imitation for this learning process.
User training is most likely to be carried out as followed. Each student has a computer, and the teacher uses a video projector. The teacher instructs which keys to push on the computer by demonstrating it and projecting the screen. Instructions are guidelines which lead the user step by step through a procedure, demonstrating how to carry it out. The learners try to remember the keys and repeat the operations being carried out. Due to the low capacity of our short term memory, the learners often forget the steps.

A similar situation happens when an IT support person verbally instructs a user which buttons to push in order to solve a problem. The next time the problem occurs, the user has forgotten the steps.

Since a colleague or a trainer normally cannot stay around for repeated demonstrations until every learner has acquired the skill, other some media might be helpful.

### 2.2. Instructions sheets

Instructions can be written, video or audio recording, or a combination thereof. We will call the written documents instruction sheets. These can include graphics, and they can appear in various media, for example, in-line help in the software, web pages or printed documents.

For imitating written instructions, users need to recognise the IT application when reading them, so including screenshots is normally necessary. Screenshots are also needed for pointing exactly where to push a button or tick a box.

**Setting up a link**

1. Open the document.
2. Click where you want the table to appear, and then select **Insert → Object → OLE Object**
   
   Fill the dialogue box as follows.

   1. Select
   2. Find file
   3. Link file
   4. Confirm

   ![Insert OLE Object](image)

3. Check that the link is working by altering some numbers in the spread sheet and in the document select **Tools → Update → Update → Open Read Only**.

Figure 1. An instruction sheet for setting up a link from a document to a table in OpenOffice in Windows.

© Jens Kaasbøll, 13th April, 2012
Operating an IT device is a sequential series of actions, so instructions need to specify the sequence. Video, audio and text are sequential modes of expression, while illustrations are not. Illustrations therefore need an added sequencing, for example as in Figure 1.

Reinforcement from the system that the operation is successful is an important factor for learning (Ormrod, 1995). Digital devices often provide immediate feedback on the result of the operation. If the result is what the user wanted, the feedback is a reinforcement of learning. Sometimes, the system does not provide appropriate feedback, so that learning is not reinforced. The users should therefore do additional operations to control the output. The last instruction in Figure 1 is for checking that the operation has yielded the desired result.

Few users read manuals (Novick et al., 2007), and long texts are particularly unlikely to be read by most users, since they are more interested in doing than in reading (Carroll, 1990). Instructions should therefore be short. The instructions in Figure 1 are as short as possible and still complete. The example in Figure 2 has short and precise instructions for how to trigger the selection of fields. However, the instructions to the left for how to fill the fields are too wordy and are not broken up into steps.

![Inserting Merge fields](image)

**Figure 2. Instructions for mail merge (MS Office 2007)**

The instructions in Figure 2 also illustrate another common problem with screenshots. We are often interested in small portions of a large area. If the screenshot includes the whole window, the details in the area of interest become tiny and difficult to see and mark up with arrows. A
better solution in Figure 2 could be to extract the name and address area of the letter and blow it up, so that the reader can easily spot the exact position where to type Last_Name.

Choice of example will also influence the complexity of the sheets. The example should illustrate the normal execution of the operation, without including any other, disturbing data.

Studies of user experience reveal that they are not satisfied with the instructions provided for the software they use. Typical sources of dissatisfaction are that IT instructions are too basic, but also too difficult to imitate (Novick and Ward, 2006, Smart et al., 2001). The observation that some documentation is too basic may come from the fact that it is intended at the novice, and then the more proficient user finds it too detailed. The opposite may be the case when the documentation is too difficult.

The instructions in Figure 1 are intended for users who have some skills in navigating in the menus. Novice users might have had problems with finding the Insert menu option, since the instructions do not include a screenshot indicating where this menu is located.

![Save and open](image)

**Save and open**

1. Saving. Before quitting a program:
   1.1 Click the button.
   1.2 Click the down-arrow in the “Save in” box.
   1.3 Type the name you will give the file in the “File name:” box.
   1.4 Click the “Save” button

2. Opening. The next time you have opened the program and want to work on the same file:
   2.1 Click the button
   2.2 Click the down-arrow in the “Look in” box.
   2.3 Double-click on the file name.

Figure 3. Instruction sheet for novices (MS Word, 2003)

The instructions in Figure 3 are detailed concerning a basic operation, so it fits novices. A user who has acquired some IT skills would find these instructions too basic. One might argue that more advanced users would not look up how to save a file. However, if one wanted to know how to save with a special file format, searched for this and hit the instructions in Figure 3, then the instructions would turn out as too basic. On the other side, if the novice does not know where to find the and buttons, the instructions would be too difficult.

The example indicates that instructions have no obvious level of detail which fits all users. The computer scientist may launch the idea that the software should track the users’ skill level and present instructions accordingly. However, even keeping one version of user documentation correct and up to date seems to be too demanding for many IT vendors and in-
house software systems, so managing a set of different levels could easily lead to more chaos than improvement. A more realistic approach is to make some simple assertions about which functionality that will be used at different skill levels, and adjust the instructions accordingly. Table 1 provides recommendations for adjusting the instructions to skill levels.

Table 1. Skill levels and corresponding instruction design.

<table>
<thead>
<tr>
<th>Skill level</th>
<th>Operations</th>
<th>Presentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novice</td>
<td>Any basic</td>
<td>Screenshot and every detail.</td>
</tr>
<tr>
<td>Ordinary</td>
<td>Any basic</td>
<td>Brief mention</td>
</tr>
<tr>
<td></td>
<td>Menu selection for new operation</td>
<td>Textual navigation from main window to location.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E.g., Insert → Object → OLE Object</td>
</tr>
<tr>
<td></td>
<td>Unknown window</td>
<td>Screenshot for navigation</td>
</tr>
<tr>
<td></td>
<td>Several operations</td>
<td>Sequence</td>
</tr>
<tr>
<td>Advanced</td>
<td>Any ordinary or basic</td>
<td>Brief mention</td>
</tr>
<tr>
<td></td>
<td>Menu selection for new operation</td>
<td>Textual navigation from appropriate point to location.</td>
</tr>
<tr>
<td></td>
<td>Unknown window</td>
<td>Screenshot for navigation</td>
</tr>
<tr>
<td></td>
<td>Several operations</td>
<td>Sequence</td>
</tr>
</tbody>
</table>

Missing the skill level is not the only trouble that users report concerning instruction sheets. Another complaint is that user documentation is out of date (Novick and Ward, 2006). Outdated material was abundant when manuals were printed, and new software versions were distributed. Publishing instruction sheets on the web ease the updating.

**Tools for creating instruction sheets**

When creating instructions which include screenshots, one normally needs copying a portion of the screen and thereafter adding some graphics and text. The Print Screen key copies the whole screen, so for selecting an area, the image has to be cropped to the desired size by means of a software tool which can handle raster graphics.

Windows 2007 and later has a program called Snipping Tool which produces a copy of an area which the user can select. Ubuntu Linux has the option Applications → Accessories → Take Screenshot → Grab a selected area. In Mac OS X, Command-Shift-4 allows you to select an area of the screen and save it as a file, while Command-Control-Shift-4 saves it to the clipboard. For finding out how to make screenshots of mobile phones, search the web with the terms Screenshot and phone name.

Instructions can be presented in many media for instruction sheets; in-line help which appears in the software, a slide, a web page, a text document. If only one form of publication is relevant, the instructions should be made with an appropriate software for the medium, for example, Impress, PowerPoint or Prezi for slides. If the instructions are to be published in several media, the professional approach would be to store the instructions in a format from
which they can be extracted for any type of publication. DocBook is a such a format, which is intended for writing technical documentation.

2.3. Instruction videos

The previous section outlined four principles for instructions; sequence, recognisability, brevity and completeness. These principles hold for any medium. The contents and structure of an instruction video should therefore be similar to the sheet.

Generally, the written text in instruction sheets would be presented orally in a video, and the static screenshots would be replaced by a dynamic screen capture, showing mouse movements and characters being typed. Videos may also need some graphics like arrows or highlighting for drawing attention to specific parts of a window.

Examples of video instructions are abundant on the web. Two introductions to formulas in spreadsheets can be found on YouTube:

- 06 Google Spreadsheets Cell Formula pt 6 of 7 (mrwaynesclass, 2009)  
  http://www.youtube.com/watch?v=vZvtsNotlEo

- Creating formulas using cell ranges in an OpenOffice calc spreadsheet (COL CCNC, 2010)  http://www.youtube.com/watch?v=U7QlOpluAF0

The first video has replaced written text with sound, while the second one has kept the written instructions and has no sound.

Concerning the four principles for instructions, sequence is guaranteed, since the video is the medium. Second, the use of screenshots enables recognising the software. However, there is a large number of cells filled with data in the examples, so the learner needs to be able to disregard the cells which are irrelevant for the insertion of formulas. Excessive amounts of data or of interface details clutters the picture and makes it unnecessary hard to recognise the essentials. Third, both of these videos are short; around one minute. Users are more likely to watch a short video to the end than a long one, and most other instructional videos are longer than these. Fourth, the videos are complete in the sense that they cover all steps necessary to insert the formula.
When perceiving the world, people group together stimuli which are located closely together, which are similar, and which constitute shapes which we expect (Ormrod, 1995). The yellow callout in the silent video (COL CCNC, 2010) points to the rectangle that is dragged, see Figure 4, left part. Also, the yellow colour appears both in the callout and at the cursor position. This support the association between the callout and the rectangle in both location and similarity, so it supports out perception in two ways. The callout in the right part of Figure 4 is located far from the rectangle, so in this case, the association is only through the similarity in colour. This example is form a video, but the principles of closeness and similarity for achieving an association in perception is valid also for static illustrations.

**Tools for creating instruction videos**

Producing a video can be done in three steps:

1. Recording the screen and voice by means of a screen recording and video production software. The recording yields a series of frames, as illustrated in Figure 5. The series of frames is stored in the format of the software package.

![Figure 5](image.png)

*Figure 5. A series of frames for video production. Screen capture from Wink (Kumar, 2010).*

2. Editing the frames. Frames can be deleted and added from other recordings. Also graphics can be added in this step.

3. Rendering. The software produces a video file, which could be

   a. Animated vector graphics – Flash .swf can be played with Adobe Flash Player.
There are several softwares which can do the whole or parts of this process. Three examples are:

- Adobe Captivate is a commercial product with extensive functionality (Adobe, 2012). It runs on all platforms.
- KRUT is freeware and runs on all operating systems (Östby, 2012). However, it skips the editing step.
- Wink is freeware and can do the steps 1-3 above (Kumar, 2010). It runs on Linux and Windows.

2.4. Training for skills

It was noted above that learners quickly forget long series of operations. Having written or video instructions, users can look them up when necessary. While users seldom read manuals, they are twice as likely to look up in training material (Novick et al., 2009). Teachers in training sessions should therefore hand out instruction sheets or videos to the learners instead of instructing by means of a projector (Herskin, 2006). Then the users will have training material to look up in when they are back at work. Following an instruction sheet instead of the teacher at the projector also eases the learners’ practice during the course, since they can follow their own pace. During teacher instructions, some learners work slower than the teacher, such that they are left behind. One might object to the video on the same grounds; that the learner cannot follow its pace. However, videos can be paused and replayed indefinitely, in contrast to the teacher in front of a class.

Training in courses by means of instruction sheets also relieve the teacher from running around in the computer lab to help out those who forgot the instructions (Herskin, 2006). Nevertheless, some learners with insufficient digital literacy do not imitate the instruction sheets but asked their fellow students for help instead (Hadjerrouit, 2008). This might be a symptom of that written instructions are more abstract than live ones, such that novices should also imitate the trainer with projector or possible view a video, which is more concrete than a written sheet.

Instructions sheets and videos need to be stored where users can find them when needed. Searching Google with “guide windows” yields more than one billion hits, and there are more than a million instruction videos for Linux on the web. The users’ challenge is to find the right one. Research has reported that users have trouble finding instruction sheets and other documentation when needed (Novick and Ward, 2006). This challenge will be addressed in Chapter 7.

When introducing new business specific software in an organisation, people need to learn the skills for using it. How to organise for user learning will be taken up in Error! Reference source not found.. In any case, instructions need to be produced and distributed. Knowing
that users have trouble searching for and finding relevant instructions, the best option is to place the instructions such that no search is necessary. The solution is to include instructions in the user interface of the software, so called context-sensitive or in-line help (Shneiderman and Plaisant, 2010).

2.5. **Assessing IT skills**

Upon completing user training, the teacher may want to know whether the users have learnt the skills aimed at. Since skills are demonstrated by doing and not by saying, tests of skills should be through practical exercises. Exercises like

- Summarise both rows and columns in the spreadsheet.
- Use styles consistently in the document.

are therefore appropriate for testing IT skills. The trainer needs to observe the performance of the learners on the IT device to judge whether they are at the wanted skill level. Alternatively, viewing the result produced by the trainees may be done, but such inspection does not capture the mistakes which the learners might have done on their way. The following question

- How do you summarise both rows and columns in a spreadsheet.

calls for an oral answer and not a demonstration of practical skills. The following question is even further from testing skills:

- What is a style in a text processor?

This question does not concern know-how at all, but rather knowing-that or understanding.

2.6. **Summary**

IT skills are strengthened through practice. Trainers and other people might speed up the learning of new skills through the learner imitating their behaviour. Users may also imitate instructions in documents or videos. We illustrate imitation in Figure 6.

![Figure 6. Imitation. The arrow denotes the learning process. The IT to be learnt above, and the instructions to guide learning below. Instructions may be provided by people, videos and documents.](image)

© Jens Kaasbøll, 13th April, 2012
A golden rule for user learning is:

1. **Provide users with detailed instruction sheets or videos, also during training.**

**Exercises**

1. Redo Figure 2 according to the principles of sequence, recognisability, brevity and completeness.

2. Does the Instruction sheet for novices (Figure 3) ensure that users gets feedback to reinforce learning?

   If not, how can it be added?

3. Assess these instruction videos according to the four principles:
   
   
   http://www.youtube.com/watch?v=WW2IDE4rPCc
   
   http://www.youtube.com/watch?v=1RNdQj74OvU&feature=related

4. Construct an instruction sheet (and example files if needed) for learning how to
   
   a. Insert a table of contents in a document
   
   b. Insert pictures into documents
   
   c. Insert graphs in spreadsheets
   
   d. Create combination charts in spreadsheets
   
   e. Create e-mail filters
   
   f. Upload pictures to flickr

5. Make instruction videos for the same cases

6. Find instruction sheets and videos for making bibliographies on the web. Zotero, EndNote and the Bibliography References in Word are relevant tools.

   a. How can they be improved?

   b. Make your own instructions for the same

7. From your experience, which computer operations do your friends, family or colleagues struggle with?

   a. What is difficult in doing these operations?

   b. Design an instruction sheet for the difficult part.

**Project**

8. Select a topic which you have observed that people are struggling with, and which requires understanding of a new concept. Choose the media for instruction, generate an example and make a sheet or video.
Pedagogical theory – behaviourism

Within a training and transfer view of user competence, the outcome of the learning process which takes place during training is our focus. This view of learning is in accordance with the behaviourist approach, where learning is considered a relatively stable change of the potential for action. That means, after learning, the learners should be able to do things which they could not do before, and that this ability is not a random change. Being able to do something does not necessarily imply that it is done, since required conditions like time and money might not be present. The behaviourists only consider observable behaviour, meaning that what goes on in people’s head is outside the area of interest.

The typical way of regarding learning in the behaviourist perspective is that a person is presented with a stimulus from the environment, for example Arja’s computer displaying a spread sheet table and a document. Thereafter Arja responds to the stimulus, for instance by importing by a link. If this response was different from the previous ones, and also that Arja continues with this response more times when she is presented with the same stimulus, she has learnt a new behaviour.

After a response, the person can receive a new stimulus, which can reinforce the learning, for example that the numbers in the document are updated according to changes in the spread sheet. If seeing this makes her more inclined to import by link the next time she sees a spread sheet table and a document, then the updating constitutes a reinforcement for her learning.

Positive reinforcement like appraisals and negative reinforcements like the disappearance of an error message will both strengthen learning. Immediate reinforcements are better than delayed ones, and informative better than uninformative ones. For example, observing the change of numbers is an informative reinforcement. Repeated reinforcements at variable intervals are also more effective than only one reinforcement or repetitions at regular intervals.

Punishments weakens learning and are the opposite of reinforcements. For instance, after printing a file, the computer displays the message Illegal user action! Printer damaged! would for most users constitute a punishment, making them less likely to try the print command again. Also, lack of any new stimulus after response weakens learning, for example that the computer does not provide any feedback after the user issues a print command.
Part II. IT understanding and learning

The learning aim of Part II is:

| To be able to design learning activities and material for IT users such that their competence is brought up to a level of understanding. |

The difference between skills and understanding, which was introduced in the Conceptual-practical training model, will be addressed in this part of the book. This will be done for three subject matter areas which will be identified for IT competence: tasks, IT and information.

Part II will take on a learning view of developing IT user competence, considering learning as processes towards higher levels of mastery in the three subject matter areas. Learning takes place anywhere and anytime, not only in courses or other activities aimed specifically at learning. However, a more thorough view of learning will also help refining the training designs from Part I. Also, the competence areas and levels constitute a background for design of learning material and organisation; the latter will be covered in Part III.

Pedagogical theory – constructivism

The constructivist perspective on learning is based on the assumption that new skills and understanding is based on what we already know. For example, if we see a button in a new program which has the same name as in the previous program, we assume that it does the same operation. This implies that when we are presented with something new, we always associate it with something familiar. Since new understanding is based on the already existing one, we construct our own knowledge; we do not copy the teacher’s understanding.

People are active and communicating learners, and learning takes place in interactions with the environment, including fellow learners, teachers, computers and books. A culture which encourages questioning is therefore promoting learning.

In order to accommodate for the learners’ construction, the teacher needs to know the learners’ starting level, so that teaching can be directed towards what is needed to build from that level.

This part of the book will identify levels of competence, which the teacher can use for assessing the learners’ levels. Also, we will see in the section on learning failures how learning can go wrong when starting from a basis which is not aligned with the material to be learnt.

The theoretical approaches to learning are illustrated in Figure 7.
Pedagogical theories

Behaviourism
Part I
• Observable behaviour
• Learning is a relatively stable change of behaviour

Constructivism
Part II
• New competence based on what we already know
• We construct our own knowledge

Situated learning theory
Part III

Levels of competence

Gagné-Briggs principles of instruction
Kirkpatrick’s model of evaluation

Figure 7. Pedagogical theories and their application in this textbook.
Chapter 3. Subject matter areas

The learning aim of this chapter is being able to identify the three subject matter areas in topics to be taught.

‘Competence’ broadly denotes abilities related to work, while ‘knowledge’ does not signify any particular aim of application. Since this book is about abilities for using computers, ‘competence’ will be preferred as the basic concept. But keep in mind that computers also are being used outside of work. Therefore, the interest here is competence for some activity rather than for participating in working life. We will use the terms ‘IT use competence’ or ‘IT user competence’ interchangeably, and the IT will also encompass communication technologies and mean the same as ICT.

During training, the subject matter to be learnt has been found having greater implications than other factors on teaching practice, including class size and level (Stodolsky, 1988). Consequently, finding the subject matter areas of IT use competence is necessary, the next chapters will show us that each of them have individual steps of learning.

Competence for using IT shows us the purpose of the competence, but not it’s content. Although it is obvious that IT constitute at least some of what the competence is about, we will also see that there are other subject matter areas of the IT competence.

Concepts like ‘software knowledge,’ ‘computer literacy,’ ‘information literacy,’ and ‘digital literacy’ have been coined to capture the essence of what IT use competence should constitute. We do find streams of such work in three different academic areas: computer and information systems, information and library science, and research on IT in schools.

In the research area of information systems, (Sein et al., 1998) proposed a six step model for user competence. The three lower steps of their model concerned learning to use the functionality of the software. The three upper ones concerned the connection between technology and the organisation where the IT was to be used. This way of including the use of IT in the organisation seems, in addition to the technological, to be a subject matter area.

As a next step, Coulson et.al. (2003) also added a learning-to-learn step on the road to learn a computer system. Their experiment showed that learners who received training on explicit conceptual models of ERP systems became better articulating ERP concepts. Since companies often send someone for training and let them train others, an improved grasp of the conceptual model of the system could help those becoming better trainers.

In the information systems literature, another take on IT user competence used the categorisation of cognitive, skills, and affective competence for studying methods of measuring the user competence (Marcolin et al., 2000). Skills relate to the lower levels of the model of (Sein et al., 1998), while cognitive competence concerns the higher ones. The affective competence, which Marcolin et.al. studied, was self-efficacy. This division of
competence into cognitive, skills and affective aspects is a general one, also found in (Gagné and Briggs, 1974), and it does not bring us closer to specific subject matter areas of competence for IT use.

‘Knowledge domain areas’ are also brought in by (Marcolin et al., 2000) as a dimension in their classification of user competence. Word processors and spread sheets constitute the knowledge domain areas of their study. These knowledge domain areas only concern the type of software used. Considering the endless supply of new software, this does not bring us closer to a general characterisation of user competence.

Moving to the information science, they have used the term ‘information literacy.’ Literacy came with the transition from oral to written culture, starting up more than 5000 years ago in the small with the cuneiform in Iraq (Walker, 1987). So the political and scientific discussion about literacy has been around for centuries.

Library scientists have seen the need for defining the competence for users of libraries catalogues and classifications. In information and library sciences, we can find almost any conceivable explanation of what information literacy really is (Buschman, 2009). The web with its search engines has been a trigger for new discussions about information literacy, and the Association of College and Research Libraries came up with a competence standard for higher education (2000):

An information literate individual is able to:

- Determine the extent of information needed
- Access the needed information effectively and efficiently
- Evaluate information and its sources critically
- Incorporate selected information into one’s knowledge base
- Use information effectively to accomplish a specific purpose
- Understand the economic, legal, and social issues surrounding the use of information, and access and use information ethically and legally

The standard also divide these topics into levels suitable as learning goals for children at various ages.

In contrast to the IS literature, this definition does not include the information technology, but it includes the ability to access and evaluate information. Similar to the IS literature, the ability to use information is included in this standard.

Switching to the school context, the first attempt at characterizing IT user competence seem to be Luehrmann’s speech from 1972, “Should the Computer Teach the Student, or Vice-Versa?” (Luehrman, 1980). He advocates for pupils learning programming as well as using computers for analysing social or ecological data, text and music and create graphics. The issue of students’ computer literacy has been debated, where some have advocated that
students need learning programming, while others have minimised the need for school children to learn about computers. With the expansion of information technology into other gadgets, the phrases ‘digital literacy’ and ‘new literacies’ have replaced ‘computer literacy.’

Leu et.al. (2004) define ‘new literacies’ in this way:

*The new literacies of the Internet and other ICTs include the skills, strategies, and dispositions necessary to successfully use and adapt to the rapidly changing information and communication technologies and contexts that continuously emerge in our world and influence all areas of our personal and professional lives. These new literacies allow us to use the Internet and other ICTs to identify important questions, locate information, critically evaluate the usefulness of that information, synthesize information to answer those questions, and then communicate the answers to others.*

Leu et.al. (2004) also emphasize that the IT has brought literacy from text comprehension to understanding an expanding system of all kinds of signs, including interactive elements, pictures and animation. They also point to that the technology is changing so rapidly that the changes of literacies are limited by our abilities to adapt (Leu Jr. et al., 2004), so new technologies frequently redefine what it means to be literate. Hence, they include competence in using the technology in their conception of ‘new literacies.’ The multiplicity of the concept refers to that meaning is represented in various media, and that it is used in different contexts. Lankshear and Knobel (2008) summarised 35 years of discussion about IT in schools, stating that instead of considering digital literacy as a single skill, there are myriads of practices where IT is involved, and mastering each of these requires a specific ‘digital literacy.’

Like the IS literature, the school setting scholars also include the technology and the way it is used in practices in their conceptions about IT user competence. Also, they share the concern for the ability to express and evaluate information with the information scientists. The lessons from these three research areas of information technology competence thus points to that IT users need competence in three subject matter areas:

- **Information.** Systems of signs of any type, including text, numbers, images, videos, etc., and the way a part of the world is represented by signs.
- **Information technology.** The functionality and user interface of software and hardware, including paper, in which the information is stored and processed.
- **Tasks support,** for coupling the tasks in which the IT is used with the functionality Similar three-partitioning are found in some recent studies of IT user competence.

In a study of a health management information system, (Kaasbøll et al., 2010) pointed out the representation of the domain, the information technology, and the management practice as the three competence areas for a management information system. Representation of domain means the information which represents something outside itself, being health in the case of health management information system. The management practice is the tasks in which the system is going to be used.
Puri (2007) studied the sharing of maps between GIS personnel and indigenous inhabitants of the mapped area in India. He identified three aspects which were crucial for both groups to achieve a shared understanding of the maps. First, the contents of the map, or ‘the scope of knowledge embedded in the map,’ which corresponds to the information subject matter area, as expressed above. The second aspect is the information technology, and the third one is the practices which go into the utilization of the object (Puri, 2007), corresponding to the task area. The joint understanding of these three aspects enabled cooperation between the two groups.

This three-partitioning also corresponds to what is found in modern information systems development methods. The tasks in users' practice are modelled by use cases, the information is structured in a class model, and the technology by a handful of other formal models.

Reviewing the conceptual-practical training in Chapter 2, the usefulness belongs to the subject matter area of tasks, while the skills and understanding concern the information technology area. The information area was less important in the example presented, so it was omitted there in order to reduce the complexity in the introduction.

**Example 1 – Bank account**

Information. The banking information is representing Aziza’s account and the transfers to and from her account. The representation of a transfer would contain a date, an id, the id of the sender or receiver, an amount, etc. When paying, she needs to know that the receiver id corresponds to the intended receiver.

IT. Being a competent user, Aziza knows that there is a computer in the bank, possibly extended to a web browser, a card payment terminal, and an ATM, where she can withdraw cash.

Task. The system supports the paying and the withdrawal tasks of Aziza’s practice, and she needs to know when to pay someone and how much cash which is appropriate to withdraw.

**Example 2 – Health information system**
Figure 8. An example of health information.

Information. A health information system contains data about health activities, for instance, the number of babies immunized last month or the number of malaria cases treated. These numbers represent the domain, being the health of the population in the area. Health competence includes, for instance, knowing that epidemics may vary seasonally, but may not change dramatically from one year to another unless war or natural disaster. The malaria information shown for four districts in Figure 8 shows a normal seasonal variation, except that the competent health professional Bobby will notice that the figure for October, Riverside is an incorrect representation of the domain.

IT. The numbers can be stored in a computer system, for instance a spreadsheet, so Bobby needs to have the skills to enter numbers, navigate and generate graphs like the one above. Also, knowing that the graph is updated when a number is changed in the table is useful competence when working with spreadsheets.

Task. The systems are used in health management, for example, when considering where to distribute malaria nets. Bobby needs to know how to utilize the data in his planning.
Figure 9. The three subject matter areas for computer users, illustrated by a health management system.

Figure 9 shows the three areas of competence which computer users need to master.

**Example 3 – e-mail**

Information. The data in one of the messages in a mailbox represents a sender, receivers, a subject and a body. The senders and receivers are entities of a particular type, represented by information with the format `localpart@hostname`. Knowing how the addresses are constructed may enable a successful guess of an unknown address to a known person. The subject and body represent the contents which the sender wants the receiver to get. Knowing that a subject “Email Award Notification of $750,000” means that someone wants to tap into your bank account is also part of the useful competence of e-mail contents.

IT. Knowing how to view incoming messages and composing new messages constitute a basic technological competence on e-mailing. More advanced competence involves mailboxes and their structure, moving messages, filtering, storage of addresses, remote servers and local mailboxes, etc.

Task. Communication is the main task. Knowing some advantages of e-mail make people prefer e-mailing when sending the same, large message to many people, and avoid it when negotiating a delicate matter within a small group. However, e-mail is also exploited for non-
communicative tasks like making a backup of your file by e-mailing it to yourself and leaving the attachment on the mail-server.

**Example 4 – Word processor**

The file generated by a word processor can be understood in two different ways, ending up in two different triangles of information-IT-tasks.

First, consider the writing of a letter about the weather. The written text will represent the weather, implying that the text is information. The word processor’s functionality for accepting, storing and displaying a sequence of symbols constitutes the information technology in use, and the task is letter-writing, see Figure 10, left part.

Second, this software has no operations for processing weather information nor for any other domains which we could imagine writing about. Rather, its operation concerns the formatting of any text, regardless of its contents. In order to do this, the text processor stores information about the text format, like the size of margins and the font chosen, and its functionality deals with changing the formatting. The tasks for which formatting is used deal with designing the letter so that it is pleasing for the intended receiver, see Figure 10, right part.

![Figure 10. The two types of data in a text file and its corresponding domains, IT and tasks.](image)

The reason for this double characteristic lies in the data that the word processor is storing. First, the written text itself is data, and second, the formatting data is about the visual design of the text. Being about two different domains, the text and the formatting data constitute two different subject matter areas of information for the user. Consequently, there are also two
different technology and task areas. The two technology areas are found amongst the functionality within the text processor, and most of its functionality deals with formatting, see the comparison in Table 2.

Table 2. The two ways of interpreting a text processor according to the subject matter areas.

<table>
<thead>
<tr>
<th></th>
<th>Contents</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>Text</td>
<td>Text formats</td>
</tr>
<tr>
<td>IT</td>
<td>Functionality for entering, deleting and moving text</td>
<td>Functionality for layout and design</td>
</tr>
<tr>
<td>Task</td>
<td>Writing and editing</td>
<td>Formatting</td>
</tr>
</tbody>
</table>

**Exercises**

1. What are the three subject matter areas in the following examples?
   - A budget
   - A ticket for the cinema
   - Buying a sweater on-line
   - A music database
   - Facebook

2. Compare a file made with a presentation program with the text processor example above. Do you find a separation which is similar to contents and formats? If so, what are the three subject matter areas for the contents and for the format-data?

3. Make an illustration similar to those of Figure 9 and Figure 10 of the task, IT, information and domain for the following examples:
   - Cinema ticket system
   - Music database
   - Library search system

4. Consider the two screenshots of software below. In which ways do these programs indicate their domains, information, functionality in the IT, and tasks? Both programs can be downloaded for free.
5. When people struggle with interpreting the results from a web search engine, in which of the three domains do they have trouble?

6. The last time you got stuck with IT, in which area was this, and how did you find out what to do?

7. When you have been asked by others to help out on some IT related topic, in which of the three areas have they had trouble?

**Project**

8. Determine the subject matter areas of the topic for instruction you designed in the project exercise in Chapter 2.
Chapter 4. Understanding IT

The learning aim of this chapter is to be able to determine the IT concepts in topics to be taught, and to identify levels of understanding of these concepts.

For the novice user, IT may look confusing, and different devices and software packages may present general principles and concepts in idiosyncratic ways. The first part of this chapter will identify some IT concepts and principles which should be recognisable for users and which appear in all software. Thereafter, levels of understanding such concepts will be explained, and this constitutes the background for designing how a sequence of concepts to be learnt should be composed.

4.1. IT concepts

Information technology is characterised by a quick turnover of new software versions, information systems and hardware gadgets. Users therefore need to upgrade their competence often, so they need to constantly learn about the technology and probably also about the two other subject matter areas described above. This implies that IT user competence also includes the competence for learning about IT, which includes learning about information, learning about IT, and learning about tasks.

From the educational sciences, we know that understanding ease transfer of skills to new situations (Bransford, 2000). For example, a computer user who has understood the concept of text flow, and that text flows from one column to another, but not between cells in a table, would be more likely to choose the right kind of text structuring tool in a new software tool.

Table 3. The aspects of a concept.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Explanation</th>
<th>Example - styles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose - Why</td>
<td>The usefulness</td>
<td>Provides uniform layout throughout a document</td>
</tr>
<tr>
<td>Functionality – What</td>
<td>The transformation which it causes.</td>
<td>When applying a style to a paragraph, the paragraph becomes formatted as specified in the style. When modifying a style, all paragraphs adhering to the style are changed accordingly.</td>
</tr>
<tr>
<td>Contents (data) - What</td>
<td>The constituents</td>
<td>A style is a collection of all formatting instructions for a paragraph.</td>
</tr>
</tbody>
</table>
Internal structure  
(data structure) –  
What  

How the constituents are organised.

Formatting instructions are divided into character level, the whole paragraph layout, and special items like bullets.

External structure  
(data structure) –  
What  

How the instantiations of it relates to instantiations of other concepts

All paragraphs belong to a style.

Relation to other concepts  

Similarities and distinctions to other concepts.

Style and master slides determine layout of portions of a file. Styles apply to paragraphs, while master slides apply to slides.

The description of a concept would normally include several of the aspects mentioned in Table 3. The aspect termed ‘Why’ refers to usefulness and task support, the ‘What’ concerns data and functionality. Skill level aspects are omitted, so that Where the concept is located on the interface and How to operate it are excluded from the theoretical presentations which address understanding.

These aspects are to be included when presenting a novel concept. Including concrete examples is necessary for most people to understand the concept. A mix mode of presentation, including speaking, illustration and written text is normally better than only presenting it in one way.

Computers and other IT technologies are constructed on the basis of a few principles. Throughout half a century, more principles have been introduced for easing the design of software, and these principles have also appeared in user applications. Some basic concepts for user programs will be introduced.

**Data and data structures**

**Types: Units of data with accompanying operations**

When requested to explain a word processor, Astor says:

*You can write there and print, and change and print again. And then you can change the layout of pages and letters.*

Astor is talking about what he is doing with the program and not about the structure of the file which he has produced. A common observation is that Astor talks about software in the same way as most people, and when asked about the data in the file they have produced, few can tell anything about how a text document is structured. This leaves a limited understanding, since a basic principle in computing is that the data type determines the accompanying operations.

For example, a paragraph in a text processor document has operations like justifying, centring, aligning left or right; a cell in a spread sheet can be changed from number to date to text; and a file in an operating system can be copied and deleted.
Considering that users need to select operations which fulfil the needs of their tasks, they need to store data in a type which is appropriate for the processing they want. When needing to generate a table of data, they should choose a spreadsheets when having to do calculations, while creating a table in a text processor will be a better option when formatting is more important.

The domain to be represented and the tasks to be carried out should determine the type of data selected. When scanning a document, users may choose whether to create a picture file or a format that allows also for storing characters which can be optically recognized by the computer application. The tasks to be carried out later might decide which option to choose; character recognition allows for searching through the text in the document, while a picture can be changed in contrast, colour, etc. Further, the picture format tiff is an uncompressed representation, leaving the picture exactly as scanned, while storing in the jpeg format compresses the file with some loss of detail, but at 5% of the storage space. This knowledge about two ways of representing may also be useful for working with scanning.

In general, the *type-instance* relation is known from programming. At the operating system level, it corresponds to that a file (instance) is of a specific format (type). In a text processor, a paragraph (instance) is of a specific style (type). While users normally would not change the definition of a particular file format, they may change the definition of a style in a text processor. When doing so, the paragraphs of that style will change accordingly, so that the type-instance relation is conserved, and the changes of the type has a cascading effect.

Data being an instance of one type may be transferred to another type by *export* and *import* functionality. Some properties of data are normally lost during transfers.

Types can be in a *generalisation-specialisation relationship*, meaning that the specialised type inherits properties from the general one. For example, the Heading 2 style in a text processor can be built on the Heading 1 style and inherits the paragraph formatting from the Heading 1. Whether changes on Heading 1 are propagated to Heading 2 and all its paragraph instances may vary according to the brand of the text processor.

**Structures of data units**

In line with (Norman, 1983), we will say that *conceptual models* are structures, concepts and principles of data and functionality and their purpose generated by the designer of a system.

Data units of the same or of different types are organised into larger structures, commonly by sequential order, hierarchies, and coordinates. The letters in a text document are sequentially ordered, files in the operating system have hierarchical organisation, while cells in spreadsheets are organised by 2-dimensional coordinates. The structures allow creating wholes which are combined from parts, corresponding to the aggregation type of relationships.

Again, there are operations which can manipulate the structures, like inserting a new column in the spreadsheet or moving a folder from one place in the hierarchy of the operating system to another.
Figure 11 presents a conceptual model of the data structure of a file generated with a presentation program. It depicts that the file contents is broken down into smaller and smaller parts. Also, that there are master slides which determines the layout and design of the slides and their contents, and that the master slides also can be parts of larger structures, called templates. The model is an example of how type-instance and data structures are mixed in a file with a more complex construction.

The model in Figure 11 is presented with a notation taken from computer science. It is therefore inappropriate as a means of communication with users, since they are not acquainted with the abstract notation utilized here. We will see in chapter “Developing material for learning” how such models can be adapted to fit users’ experience.

Computers allow for creating links, references, shortcuts or whatever they are called in order to achieve two effects:

**Functional dependency**, meaning that when data is changed where stored, the changes are also accessible from where the link to the data goes. This principle ensures that data is stored and hence updated one place, so that inconsistencies are prevented.
Breaking the structure, in the sense that from one place in a file, there is a reference which brings the user to anywhere else in the file or to another file, regardless of hierarchical, grid, sequential or other orders. Type-instance and generalisation-specialisation relationships may all be considered variants of the functional dependency principle, where changes at one place cascade to those places which are linked to it and are supposed to be influenced by it.

Meta data

The properties of a file, which often can be found by right-clicking on the file symbol in the operating system, are data about the file, and this aboutness relation is normally called meta. Some of the meta data are functionally dependent on the contents of the file and its production process, while other can be set, like the access restrictions.

Access rights

Access rights can be ordered from very little to the maximum:

Knowing the existence, which can be called Write-only when you enter the data yourself and Notification of existence when others have stored the data. When writing your password, you cannot read it, and when sending off an e-mail, you cannot read the copy which ends up in the receiver's inbox. There is no access after the writing.

When searching for a book or paper on the web, you may come across the title at a web site, but in order to see the full text, you have to pay a fee. Without paying you are in the situation that you know the existence of the data, but you have no access to its contents.

Read only. The huge majority of pages on the web are only for reading, and you cannot change them. You may download the source code and change your own copy of the page, but those changes will not be stored at the original web page.

Append. A blog would allow you to add text, but not to change what others have written.

Change. When buying a book from a web-shop, you write an order, which is stored in the database of the bookshop, where you are appending data. When you change your address, you are also deleting the old address and inserting a new one, so you have the change access to your personal data. When updating an article on Wikipedia, you also change what others have written, although your change access might have to be approved by an editor.

Delete means not just changing the contents of a file or deleting all its contents, but deleting the file itself. Most users have Delete rights to the data files on their own computer, digital camera, mp3 player or other personal devices. Normally, on shared data, only the owner can delete.

© Jens Kaasbøll, 13th April, 2012
Access rights are associations between users and data. Often users are divided into groups, each of which as specific access rights. For example, there are internal users with change rights, external members who can append, and non-registered users who only can read.

**The layered architecture**

As seen in the previous chapter, the contents and structure of a document constitute one way of regarding word processors, while the formatting enables another view. This can be expressed by saying that the document can be separated in one contents and structure layer and one format layer, and that each of these layers can be changed independently of the other. In web page design, the structure and contents can be coded with html, while the layout can be set by Cascading Style Sheets.

Users who mix up the two layers are likely to do more work when changes have to be made than those who keep layout or format separate from structure and contents. For example, users who add a blank paragraph in order to achieve a format effect, namely larger space between paragraphs with text, will have to change each paragraph. Paragraph formats, including space above and below, can be set by the styles, which is a formatting tool. When changing the style, all paragraphs of that style are updated.

In general, all data can be viewed and manipulated at many layers. For example, if a file is suspected to contain a virus, it can be opened by a Notepad or similar editor, which treats all data as characters, thus internal codes and user data are viewed as being of the same type.

While the deeper layers of the computer software is normally left for the programmers to deal with, having some insight into layers of the internet protocol may, for example, help users understand where connection problems reside.

The hardware layer has some principles which users need to cope with, since they have to grasp the difference between input and output. Some may also have understood that for example, the memory chip in a digital camera is also a general storage for data, so they can use it as a backup for their files when on vacation.

Of philosophical interest is the principle that data and programs are stored in the same way, so that programs can have other programs as their data and operate on them (von Neumann, 1945). This implies that the computer can change its own way of operation, and this property distinguishes computers from all other technology.

**Sequencing**

Computers and other digital devices operate through sequences of operations. Seen from the user point of view, each application running on a computer carries out its own sequence of operations called a *process*. Several programs can run their processes at the same time, without interfering with each other, so their processes run in parallel. For example, we can write in a word processor and import music from a CD to the computer at the same time.
Correspondingly, a Facebook wall can contain several processes, each having a topic which people write comments to. In contrast to running a word processor and a CD import at your own computer simultaneously, Facebook involves several people, each with their own computer, contributing to a joint communication process.

We achieve a continuous communication process when all participants are present at the same time, also called \textit{synchronous} communication. When writing on the Facebook wall, there can be time delays between the comments, so the communication process is discrete or \textit{asynchronous}.

Concerning the number of readers, we distinguish between \textit{point to point} and \textit{mass communication}, the latter is for everyone to be a receiver, or to have read-only rights. Coupling the two modes with the number of receivers, we get a matrix like Table 4, where examples of technology are included.

\textbf{Table 4. Modes of communication and number of receivers with examples of technology.}

<table>
<thead>
<tr>
<th></th>
<th>Point-to-point</th>
<th>Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{Synchronous}</td>
<td>Chat</td>
<td>TV</td>
</tr>
<tr>
<td>\textit{Asynchronous}</td>
<td>E-mail</td>
<td>Web</td>
</tr>
</tbody>
</table>

Third, there is a difference between whether the communication process is initiated by the sender, a \textit{push}, or by the receiver, a \textit{pull}. Sending off an e-mail is a push, while downloading a web page is a pull operation. In the synchronous point-to-point communication, the initiative switches and the communication process becomes \textit{interactive}.

Forth, the type of data transmitted makes us select appropriate software and determines how we sense the data and express ourselves. Being synchronous, point-to-point and interactive, the phone requires hearing and speaking, while the textual chat depends on seeing and typing.

Some software, like a learning management system and Facebook, utilize combinations of communication functions, each according to the four dimensions above.

When booking a ticket at the movies, the user initiates an asynchronous communication process with the database at the movie theatre’s computer. Simultaneously, other customers can run similar processes against the same server. In this case, each communication process has a definite starting point when the user identifies the seats she will book. Thereafter, only this process can access the data for these seats until the seats are booked, the user decides not to book them, or the user has spent too long time, so the process is ended by the server.

\textbf{4.2. Learning IT concepts}

When using IT, people learn skills, but practice does not necessarily promote understanding. While skills are associated with doing, understanding requires the ability to express and communicate in talking or writing about the subject matter areas. We will here describe the
learning trajectory when starting out with skills and ending up in understanding through levels of mastery. Such levels can be used by the teacher to assess the learner’s competence at any stage, so that the teacher can stimulate the learner in a way which the learner can relate to, and which also points to the next stage of mastery.

Categorising increasing levels of competence has been done for the purposes of setting educational goals and for characterising learners’ actual performance. Bloom (1956) suggested a general taxonomy for advancing levels of cognitive competence in the 1950s, and it is still used for describing learning goals. It has five levels, as shown in Table 5a, and it assumes that a learner should reach a lower level before advancing to an upper one. Also taxonomies for the affective and psychomotor domains were developed (Hmelo-Silver et al., 2007).

Table 5. a) Bloom’s taxonomy of cognitive competence, and b) Dreyfus and Dreyfus’s model of skill acquisition.

<table>
<thead>
<tr>
<th>Synthesise</th>
<th>Expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyse</td>
<td>Proficient</td>
</tr>
<tr>
<td>Apply</td>
<td>Competent</td>
</tr>
<tr>
<td>Explain</td>
<td>Advanced beginner</td>
</tr>
<tr>
<td>Repeat</td>
<td>Novice</td>
</tr>
</tbody>
</table>

Dreyfus and Dreyfus (1986) suggested a five stage model for skills acquisition, see Table 5b. It characterises how practitioners acquire skills over years of collecting experience, something which modifies the novice’s behaviour according to the rules of the textbook to an expert who acts intuitively based on a huge amount of examples.

While the Bloom taxonomy concerns the learning of theoretical material, expressed in language or some formalism, the Dreyfus and Dreyfus model addresses the refinement of practical skills. Concerning use of computers, we have seen that an important learning challenge is neither of these, but rather the change from skills to understanding. This change is characterised by first being able to do something, and thereafter being able to express it. For example, after having saved files a few times and listened to explanations, the learner may be able to say what it means, where files are stored, and why we do it. Since previous models do not address the change from skills to understanding, this chapter will bring a specific model of levels of mastery for IT concepts, and the following chapters will do the same for the two other areas of competence.

Skills can be carried out perfectly without being able to explain how we are doing it, like keeping the balance on a bicycle. Skills can thus be tacit competence, also called know-how. On the other hand, understanding requires the ability of telling others what we do, which is called explicit competence, or know-that.

Research has been carried out on programmers’ learning of IT concepts. Users also have to learn IT concepts in order to understand, although on a simpler level. Therefore, we will adapt
a model developed in computer science education research for characterising also users’ basic learning of IT concepts (Aharoni, 2000). These steps have also been found amongst users’ learning (Stamatova and Kaasbøll, 2007). It identifies three steps of competence with two learning processes in between for the learning of IT concepts.

**IT skill.** The first step is being able to perform some action, for example, referring to another cell in a spreadsheet by typing a formula in one cell and clicking in another. At this initial level of mastery, the user can only carry out the operation and possibly also express each push of a button, but without being able to explain anything.

The learning process of **interpretation** takes the learner to the next step:

**IT functional understanding.** Step 2 is when the learner can refer to the input and output of this action without actually carrying it out. In the example, the user would tell that in order to have one cell refer to another, one has to get the coordinates of the other cell into the formula. It encompasses at least the functionality aspect of a concept, see Table 3.

The learning process of **reflection** takes the learner to the next step:

**IT conceptual understanding.** The final achievement is when the learner can refer to the concept as an object of its own and use it when talking about other phenomena. The concept would then have become reified. The spreadsheet user could say that cell-referencing is an ingredient in formulas.

Learners’ level of mastery can be found by observing what they do and say. Assume that they are supposed to learn the concept of a master slide and how changes in the master slide can affect all the slides in a presentation file. Learners who have completed the sequence of computer operations that demonstrates the skill can be asked to express the concept. Edith may respond like this:

*I went to view and slide master and then changed the font size, and went back to the normal view.*

Edith is repeating the steps which she carried out without expressing the initial state and the outcome of the operation, so she has not reached the second level of functional understanding yet. George might say:

*I wanted to change the font size of all the slides, so I changed it at the master slide. Then it will change all the slides.*

George has expressed the starting state of a font size, which is different from what he wants. Further, he mentions that the master slide can do all the desirable changes, which is expressing the function of the operation. George is therefore at the functional understanding

---

2 The cited articles use the terms action – process – object to denote the three levels. In order to fit with the skills-understanding pair and avoid confusing level 2 with the learning process, other terms were used here.
level. By mentioning font size and slides in his explanation, he demonstrates that for these two concepts, he is at the third level of understanding.

The learner Dmitri demonstrates that he is at the conceptual level of master slide by saying:

Master slides control the appearance of the normal slides. When you change the master, all the others will change too. It’s like the page layout in the text processor, which also changes every page.

By comparing master slide with page layout, Dmitri refers to master slides as an entity of its own. Since Dmitri has grasped the master slide idea and also seen its resemblance with page layout in a text processor, he is ready for learning a more general concept through analogical reasoning.

User understanding is termed mental models. In addition to the relationships between tasks, functionality and concepts which reflection brings about, mental models are normally also considered to include the way the interface is operated (Westbrook, 2006). This latter part which links understanding and skills will be the results of the learning processes navigation, see Section 8.2.

Interpretations brings the user from a skill level to a functional understanding of a specific aspect of the IT. Observing the output of computer operations and the verbal and iconic signs at the user interface may be sufficient for an interpretation to take place. When an instructor is close, the user might seek help for understanding what is happening. Explanations from documentation or other people could help the user to a functional understanding.

Reflection builds on at least a functional understanding and is a cognitive process where the outcome is a new connection between structures, concepts, principles, or other mental models. Reflection belongs in the realm of understanding or know-that, as illustrated in Figure 12.
While some users generate adequate mental models on their own, slow learners are particularly bad at doing this (Furuta, 2000). The result is that they will have serious trouble understanding the next concept which builds on the one they have not understood, making them even poorer learners. Explanations and conceptual models are therefore essential for those who do not easily find out on their own what goes on beneath the surface of the computer.

### 4.3. Functional models

While instructions were presenting buttons and menu choices, functional models intend to build an understanding about input, operation and output. For example, assume that Hayley wonders what happens with the original file when converting it. Since she is afraid that it will be lost after conversion, she does not do the operation. The functional model presented in Figure 13 illustrates that the original file is kept, so viewing this, Hayley might obtain an adequate functional understanding of file conversion.
Figure 13 shows a graphical functional model of the conversion, including input, operation and output. The input and output are recognisable, while the operation is abstracted from the appearance in the user interface. Including button push would have changed the figure such that it also would become an instruction. Since functional model aim at understanding, interaction details are omitted. The input and output could have been expressed more abstractly too, making the illustration simpler but less recognisable.

**HTTP 404**

The **404** or **Not Found** error message is a [HTTP standard response code](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes) indicating that the client was able to communicate with the server, but the server could not find what was requested. A 404 error should not be confused with "server not found" or similar errors, in which a connection to the destination server could not be made at all. A 404 error indicates that the requested resource may be available again in the future.

Figure 14. A functional model from the Wikipedia.

Functional models are also needed when the output does not appear as expected. Figure 14 illustrates a functional model of a frequently appearing error. In order to understand this functional model, the user should already have understood the concepts client and server. If not, the model of network connection in Figure 15 can be used for explanation. Further, if not knowing what “HTTP standard response code” is, the user should understand that this can be ignored. There might be other functional models which fits the novice user better and still others intended for the advanced ones.

### 4.4. Conceptual models

We build conceptual models when shifting focus from the operations to the data and structures of IT.

**The layered architecture**

In order to make sense of the functional model in Figure 14, an elaborate conceptual model of web connectivity might be needed. The layered architecture of the computer is a key model to understand the network infrastructure and how the connection between computers is set up. The OSI model deals with 7 layers including hardware, while the Internet Protocol deviates from this model in some respect.

In order to see to which extent the layered architecture brings useful understanding for the user, let us try to imagine a typical situation when users may get exposed to the network infrastructure. When failing to get access to an existing web page, there are several places where the connection could be broken.

- Wrong web address (Uniform Resource Locator, URL)
• Local setting. The IP address or DNS server address is wrong, or in the e-mail client, the in or out-server are incorrect.
• Local antenna or other hardware.
• Local link. Too weak signal or broken Local Area Network (LAN).
• Wireless transmitter.
• The user connection of the Internet Service Provider (ISP).
• The external internet gateway of the Internet Service Provider.
• The server from which the web page is transmitted is broken.

The first reason, the URL, is normally visible in the browser and is also typed by users, so understanding how URLs and also e-mail addresses are constructed should be a part of user competence.

The IP address is software setting at the user's computer, and this does not appear in application software, so in order to provide a useful model of networking, we can say that it belongs to the software network infrastructure.

Antennas, cables, radio signals and transmitters are recognisable hardware devises and properties, so dividing the user's model into hardware and a software layer would be feasible, and it would contribute to understanding at least one useful distinction in the communication infrastructure.

The network further away from the user cannot be seen or experienced directly, so distinguishing hardware from software there would be meaningless. Another distinction can be experienced, however. When managing to log in successfully at the ISP with a password or network key, user experience that the connection with the internet is established. Nevertheless, the web pages may not appear, due to a broken gateway between the ISP and the internet. So a user model might distinguish between the user side of the ISP and its internet side.

The last bullet point in the list above may be experienced due to that most other web pages are accessible but the one specifically looked for.

A conceptual model with connections horizontally and layers vertically is shown in Figure 15.
Data structures — Recognisable elements

When using an application, some features and principles are easily recognisable at the interface, for example that the cells in a spreadsheet are organised in a grid, and that the text in a document has a specific layout. The sequence of operations, typically whether to choose data before operation or vice versa, may not be displayed, but they are experienced through the users' actions, so we obtain an immediate impression.

Other features are less prominent. Examples of hidden features are that the caption of an illustration in a text document does not belong to the main text flow, and that behind a number in a spreadsheet cell could be a formula which refers to many other cells. In the word processor, there is no intuitive way to see where one text flow starts and another one ends. It might be possible to view the non-printing characters, but these do not necessarily tell us about the text flows or many other properties of the document, like the paragraph and character styles.

When the user interface does not show the hidden features, they should be made explicit through a conceptual model. The written text is a one-dimensional sequence, while structures in the computer often are of other kinds. Since many hidden aspects are structural, a combination of language and graphics would normally be a better option than just one of them.

Creating useful graphical models is partly arts & crafts, but there are also principles to consider. The books by Edward Tufte constitute a comprehensive introduction to the area (Tufte, 1990, Tufte, 2011). Marti Hearst (2003) has made a tutorial on graphical elements and how people experience them, while Rosling (2006) provides a video of visualisation of numbers and statistics. These authors deal mostly with the information subject matter of IT competence. In the following, specific considerations for visualising the interior functioning and structure of software are presented.

Any presentation of what goes on in the interior of the computer should be based on the current competence of the users, including the users' understanding of concepts, experience with operating the software and their background for understanding the notation used.
In order to aid understanding, and not making it more difficult, graphical representations need to be
- simple, in the sense that they contain few \((7\pm2)\) elements
- recognisable, so that each element provides immediate meaning

Figure 16 is a conceptual model of the file system, which is its internal structure in the categorisation in Table 3. It is simple, but is made with a notation which is not recognisable by most users. On the other hand, maximum recognisability is sought in Figure 17, and this visualisation of the same conceptual model also aims at providing a general model of the structure of the file system. Figure 17 also uses examples instead of the general categories in Figure 16, bringing it closer to user experience but making the illustration larger and less simple. There is often a trade-off between simplicity and recognisability.

![Figure 16. Abstract model of the file system](image-url)
Figure 17. Recognisable model of the file system.

Figure 16 is a model of how the file system can be conceived under the surface, while Figure 17 is mainly a surface model with an additional graphical element for showing the under-the-surface connection. The user interface of the Windows file system provides a reasonably comprehensive view of the data structure when viewed in the Explore mode shown here.

For applications which aim at the What You See Is What You Get (WYSIWYG) principle, the data shown at the interface is supposed to mimic the printed copy, and then there is little room for also displaying the underlying structures. The view of a text file can include a table, which may be a copy of the spreadsheet or brought into the view because the user has set up a link to the spreadsheet file. Despite identical visual results, the underlying data structure will differ. In such situations, the illustration should depict the two data structures in the hard disk of the computer, and that they appear in the same way at the interface, see Figure 38 (p.84) and Figure 39.

The elements of Figure 38 and Figure 39 exploit the everyday experience of the layered architecture of the computer. The hardware, the data within the computer, and the visible copy of these data in way which the screen allows.

In the previous illustrations, blue arrows are used for denoting reference, link, pointer, shortcut, or whatever particular name is used for the mechanism for achieving functional dependency. Often, there is more than one type of entity and relationship to illustrate, and then the corresponding symbols have to be differentiated. Figure 11 (p.34) shows how the character fonts are used for separating the two kinds of entities in the model. Also, the two
kinds of relationships are differentiated, and since the arrow and the diamond could mean anything for users, the relationships are also labelled. Even with these labels, the model is at a high abstraction level which might be better suited for teachers of IT than for users.

Exploiting the more easily recognisable interface and adding the type-instance relations at an example may produce a visualisation like Figure 18. It addresses the external relations of the style concept, see Table 3. Adding the consists-of relationships here would have made the illustration far from simple.

Figure 18. Style-paragraph relations.

4.5. **Relationships between concepts**

While data structures can be illustrated by user interface elements, relationships between abstract concepts like data-types and functional dependency have no obvious visible clues.

For supporting conceptual understanding, we can illustrate how a concept resembles and is distinct from other concepts.

Figure 19 shows that two concepts are similar by being specialisations of a more general concept and how they differ. This graphics could be accompanied with an example in order to
make the model more recognisable. A more comprehensive illustration of relations between concepts is found in Figure 21.

Consider Fadhili, who writes his e-mail address in the URL field in his browser, hoping that this will bring him to his e-mail service. He might have mixed up two concepts; e-mail address and web page address. A possible way of clarifying the distinction between two concepts is comparing them in a table, for example as in Table 6. The categories in such a table can be extracted from those in Table 3 (p. 31).

**Table 6. A model for distinguishing two concepts.**

<table>
<thead>
<tr>
<th>Web page address – URL – Uniform Resource Locator</th>
<th>e-mail address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td><a href="http://www.google.com">www.google.com</a></td>
</tr>
<tr>
<td>Purpose</td>
<td><a href="mailto:fadhili@provider.co.ke">fadhili@provider.co.ke</a></td>
</tr>
<tr>
<td>Where</td>
<td>Locate a web page</td>
</tr>
<tr>
<td></td>
<td>Identify your inbox as the sender or receiver of an e-mail</td>
</tr>
<tr>
<td></td>
<td>Address field of browser</td>
</tr>
<tr>
<td></td>
<td>From field or To field in e-mails you send</td>
</tr>
</tbody>
</table>

We also might have to explain to Fadhili that his inbox is not a web page, and that he should rather find out the URL of his e-mail service.

**Homonyms**

Assume that Rose has an eager photographer and that she is familiar with copying to and from CDs. Then Oliver tells her that Deamon Tools is a very useful program for working with images, and that it handles DVDs well. Rose downloads it for testing it on some of her photos, but she cannot get it to open the jpg-files. The next time she meets Oliver, she tells him that the software did not work, and that she would rather stick to the image editing software which she is using.

Rose’s trouble is based in that the word ‘image’ is a homonym in the digital world. Rose means a digital photo while Oliver talks about a disk image. Again, a table could be used for discriminating between the concepts.

**Learning a sequence of concepts**

Assume that you are going to learn making table of contents in a text document. Generating the table presupposes that the items to be included have been styled with the appropriate heading styles, so you have to learn about styles and heading styles before the table of contents.

Table of contents creates a list of links to the sections in the document, and you have recently learnt about footnotes. Assuming now that the learners have understood both table of contents and footnotes, the teacher seizes the opportunity of also teach the concept of references, which is a more general category of all links in documents, and thereafter cross-references. The sequence of learning is illustrated in Figure 20.
Figure 20. A concept sequence diagram. The arrow means that the concept from where the arrow is departing must be learnt before the concept the arrow head.

IT conceptual understanding means that the learner can use the concept while talking about other concepts. This implies that heading styles should be understood at the conceptual level while learning table of concepts.

Assume that Victor after having learnt Table of contents can explain:

*We can make a table of contents by first styling the headlines that we want to include, and then do the Insert table of contents where we want the table.*

*Footnotes are like we see them in books. We make them by Insert footnote.*

He explains these two concepts partly by referring to how they are created, which points to the skill level. Also, he mentions the inputs needed for table of contents, and he compares footnotes with those in books, which indicates a functional understanding. All in all, Victor is somewhat above the skills level, but not quite at the functional understanding.

At this point in time, the teacher presents the more general idea of references and how footnotes and table of contents are two different types of references. After some clarifying questions, Victor may now be able to say:

*Footnotes and table of contents are references. That means that they point to where the text is in the document.*

With this statement, Victor no longer explains the concepts of footnotes and table of contents with mentioning how they are made. His last sentence points to a functional understanding. Also, he uses the two concepts when explaining the more general one of references, which is a hallmark of IT conceptual understanding, but since he does not mention what a reference is, we cannot be sure that he has reached this level.

Normally, people learn in a process form the concrete to the abstract, corresponding to interpreting observations to arrive at a functional understanding. For learners to understand the more general concept of references, most people will need to see a couple of different examples of references first, like footnotes and table of contents. Then, they can understand the similarities between these examples and thereafter grasp the more general concept. After having learnt the more general concept, learning more types of references will be easier. Therefore, going from references to cross-references is a feasible teaching sequence.

When designing a sequence of teaching for IT concepts, a directed graph of how they relate as shown in Figure 20 would be needed. It might be appropriate to teach the concepts which
belong to one operation at the same time, for example fields, tags and merge when teaching mail-merge.

While it is obvious that you need to learn about heading styles before table of contents, cells before formulas in spreadsheets, and fields before mail-merge, there is no definite point where the more general concepts should be introduced, and also, which levels of abstraction that are feasible. Figure 21 provides a brief summary of IT concepts which fall in under the functional dependency category. The example above about references suggests that the bottom level in the figure might be appropriate as starting points for user learning.

![Figure 21. IT concepts from abstract at the top to closer to user experience at the bottom.](image)

### 4.6. Videos providing functional and conceptual models

The contents and illustrations for functional and conceptual models should be the same in documents and videos. There are a few guidelines particularly for videos, however.

Our brains have a very limited short term memory, making it impossible to make sense of many stimuli occurring concurrently. However, sight and hearing are operating in parallel, and we can combine visual and audio impressions and make sense of the combination. This is what videos can exploit.

For instance, when presenting a conceptual model like Figure 21 in a video, the graphics should constitute the visuals, and a voice should tell about it (Clark, 2007). With a large figure like that, there should also be a finger or a colour blob marking the area which is talked about at the moment.

Adding written text in the picture for presenting the graphics would be a mistake. That would overload visual capacity while not utilising our hearing.

It has also been found that video presentations are more effective when the viewer feels like there is a conversation going on. In order to strengthen this impression, there should be a real voice, and not a computer generated one (Clark, 2007). The conversation impression is also strengthened if a person is visible on the screen for periods, although this could be a simple, animated figure (Clark, 2007).
Like any external source for learning, videos need to be as short as possible, meaning that also explanations and conceptual models need to be presented without additional disturbance. An example is normally needed, and this should therefore also be to the point without additional features.

A simple sequence for a video providing a functional or conceptual model could be:

1. Picture of a person presenting the concept in one or two sentences.
2. Picture of graphics with a finger or colour spot. The voice of the person.
3. Picture of a person repeating the concept in one sentence.

4.7. Learning failures

Learners often stumble on their paths to understanding, and we often see that the concepts they develop do not correspond to those of the teacher. That does not necessarily mean that one is better than another, but some ways of understanding may be inadequate for certain purposes.

**Discrimination error**

Consider the novice learner Herbert, who has just learnt to open programs by clicking at the symbol at the bottom of the screen, for example the Explorer symbol in Windows. Then he learns to close programs by clicking at the $\times$ in the upper right corner of the program window, and he observes that the window disappears.

Thereafter, he learns that windows can be minimised by clicking at the underscore character in the upper right corner, and he observes again that the window disappears. So now Herbert is confident that there are two ways of opening and closing programs. He does not recognise the difference between Figure 22a and b, and whenever he pushes one of them, the Explorer window opens with a search engine.

![Figure 22. The symbols on the bottom of the screen for a) starting Explorer and b) resuming it after minimising](image)

The lack of ability to discriminate between two different stimuli like these is called a discrimination error. Herbert’s initial mental model of the open-close operations can be illustrated with the diagram in Figure 23.
After learning about minimising, Herbert’s mental model might have been altered slightly to what see in Figure 24.

Two factors leads Herbert into this understanding of opening and closing programs. First, the observable difference of the result when minimising or closing is small, and unlike the illustration in Figure 22, these symbols are not displayed at the same time, making comparisons more difficult. Second, Herbert has proceeded from closing to minimising without being aware of another conceptual distinction, the one between windows and programs. Not knowing that a program can be running even if it has no window open makes it impossible to grasp the idea of minimising. So Herbert has skipped learning one concept which was necessary for understanding the following one.

A conceptual model which includes the distinction between program and window is shown in Figure 25.
Interference between IT concepts

Regardless of this achievement, people also systematically make mistakes, for example where the cell-reference concept interferes with Copy-Paste.

Assume that people working with spreadsheets for producing graphs calculate, by formulas and cell-referencing, one row of numbers B3-M3 based on the 1-2 rows, like in Figure 26.
Rows 5 and 6 constitute the bases for calculating B7-M7. The aim is to make one graph where rows 3 and 7 are compared. A common way of doing this is first to make a matrix of the numbers which are going to be displayed in the graph. People can then copy row B3-M3 into B11-M11 and row 7 into row 12, giving a matrix of B11-M12 from which the graph can be produced. Trouble appears if they have to correct B2, since B11 has no cell-reference to B2.

Knowing that the row 11 should contain the same numbers as the row 3, the users choose the copy command, instead of typing the formula $=B3$ into B11 and copy this formula to the right. The latter option would have kept the cell-reference from B2 through B3 and B11 into the graph. What is observed is that the copy-paste concept interferes with cell-referencing, since the effect of the reference in this case should be a copy of the value without any other calculation. An explanation of this commonly seen practice might be that the copy-paste is more firmly understood than the cell-referencing principle.

Interference happen when we have to choose between two concepts having some similar properties, like copy-paste and the equal-to-formula in this case, and we do not consider the consequences of choosing one over the other.

If no corrections of the underlying data in B1-M2 and B5-M6 take place after the graphs are produced, copy-paste and the equal-to-formula would both be adequate.

**Interference between IT concepts and other concepts**

When learning word processors, the learners are taught that styles apply to paragraphs. When being asked about what a paragraph is, Lura responds:

_That is a sequence of sentences dealing with one idea._
Lura is completely right when it comes to the grammatical rules, but her language understanding may unfortunately interfere with her ability to pick up computer principles. Consider the example in Figure 27. Seen from Lura’s perspective, the text in the middle has one headline and one paragraph, so that styles apply only to the paragraph, which corresponds to a paragraph in the grammar world view. Seen from the word processor’s side, all text are paragraphs, and the separation between one paragraph and the next one is marked by pushing the ‘enter’ button.

**Figure 27. Two different meanings of the term ‘paragraph.’**

Without clarifying the distinction between the two senses of paragraph, Lura would believe that styles will not apply to headlines.

### 4.8. Summary

Skills are necessary for using IT, but understanding is the basis for learning new skills. Functional understanding means being able to explain that an operation transforms an input state to a result. More elaborate, mental models of what a concept means is necessary for using this concept as a basis for learning new ones.

The interpretation learning which leads to functional understanding can be supported by functional models given by people or in documents or videos. Correspondingly, the reflection which lead to conceptual understanding is supported by conceptual models.

Slow learners are in most need of functional and conceptual models. These should therefore be presented in training and be available when users need them at work.
**Exercises**

1. Make a model of the structure of
   - A word processing file
   - A spread sheet file
   - The file system of the operating system
   - A relational database

   similar to the model of a presentation program file in Figure 11.

2. In order to produce a four pages folder to be printed, what are the pros and cons of using a text processor or a presentation program?

3. Consider GoogleDocs. Find type-instance relations, generalisation-specialisation relations, meta data, and access rights.

4. Consider the file system. Find examples of how general computer concepts are used there. Specifically, how is functional dependency implemented in the file system?

5. How is the separation between contents and formatting organised in
   - Spread sheets
   - Word processors
   - Web page authoring tools

6. Draw a model of a Bluetooth connection between a computer and a mobile phone, including the transfer of pictures and phone numbers between the two devices.

7. Four students should work together on this exercise.

   Pick a relatively difficult IT user concept, and ask your fellow students to explain it. Note down what they say as precisely as possible. On which level of IT conceptual understanding are your fellow students?

   One student asking questions, one responding, and the two others raking notes. When two students take notes simultaneously, together they will better capture what the student in the other pair is saying.

8. When requested to explain graphs in spreadsheets, we get the following responses:

   Gloria:

   *Graphs are drawings of numbers. They show us the numbers so that they are easier to compare.*

   Jussi:

   *The graphs are linked to the numbers, so when I change a number, the graph will also change.*
Yma:

*We make graphs by selecting the numbers to be charted, and then choose the graph type. We can change the format of the graph afterwards.*

At which levels of skills or understanding of IT concepts are Gloria, Jussi and Yma?

9. When asked about also displaying percentages, Pablo says:

*I know how to calculate these numbers. When I type them in one row, I can calculate the percentage in the row below.*

Which level of IT concept competence does Pablo demonstrate for this use of spread sheets?

10. Find other concepts/principles/ideas in the computer which are similar to
    a. Cross references
    b. Properties of a file in the file system
    c. Master slides

11. For each of these cases, to which other general principles from computers would you relate them? Out of the possible ones, which principle would you use for explaining to users?

12. For the following concepts, write two explanations in plain text, one as a functional and the other as a conceptual model.
    - Master slide
    - Web search engine
    - Bibliography
    - Spread sheet formula
    - File type

13. For the same concepts, draw concept sequence diagrams like Figure 20 showing which concepts must be prerequisite for these, which possible concepts do they include, and which ones could follow these?

14. For the same concepts,
    a. Identify an example from the world outside the computer which resembles the computer concept.
    b. What is the similarity?
    c. How does the computer concept differ?
    d. Express the similarity and difference on one slide, including a drawing.

15. When asked to explain tables in word processors, Zohran says:

*You use tables for dividing text into columns, like in a newspaper. Then the text can start in one column and continue in another.*
What type of misunderstanding does Zohran have? Discrimination errors or interference?

16. Have you observed cases where your friends have misunderstood concepts? Or have you done it yourself? What kinds of misunderstanding?

17. Open a spread sheet file with Google Spreadsheet, Microsoft Excel and OpenOffice Calc. Can the software display the precedents and descendants of formulas? This function is often called ‘trace.’

For which subject matter areas (information, IT, tasks) do the traces support the user’s understanding?

18. Consider Figure 15 (p.44) and Table 4 (p.37). Which basic learning processes do they target?

19. Which learning process do the following illustrations aim at; imitation, interpretation or reflection?

a. How MS Excel Is Structured by Microsoft TechNet
b. Mail Merge by Clement Khalika and Eddons Munthari

<table>
<thead>
<tr>
<th></th>
<th>Section Break</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Section 1</td>
</tr>
<tr>
<td>2</td>
<td>Section 2</td>
</tr>
<tr>
<td>3</td>
<td>Section Break</td>
</tr>
<tr>
<td>4</td>
<td>Section 3</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-1-</td>
</tr>
<tr>
<td>7</td>
<td>-2-</td>
</tr>
<tr>
<td>8</td>
<td>-3-</td>
</tr>
<tr>
<td>9</td>
<td>-4-</td>
</tr>
<tr>
<td>10</td>
<td>-5-</td>
</tr>
</tbody>
</table>


c. Page numbering by Chipiliro Awali and Muhabi Chisi

d. Table of Contents by Christina Ussein and Edward Kambwiri

20. Which learning processes do the following tutorials aim at? Do they follow the guidelines for video design? Consider both the various elements in the tutorials and the totals:
21. What kind of learning failure did Rose (p. 48) do? Make a table which distinguishes the two meanings of ‘image.’

22. Make a concept sequence diagram for the setting up a link case from Chapter 2.

**Project**

23. Which concepts do your topic build on? Which new concepts do you introduce? In which order are they introduced? Make a sequence diagram like Figure 20.

24. Analyse the concepts by means of the aspects in Table 3 (p. 31). Make graphical illustrations.

25. Can the concepts be confused with other concepts? If so, make tables distinguishing them.
Chapter 5. Understanding information

The learning aim of this chapter is to be able to determine the representation system and correspondence issues in topics to be taught, and to identify levels of understanding of representation system and correspondence. Also, the chapter aims at providing skills in the representation system used for slide design.

There is no end to the types of domains which can be represented by information, and each of them has their own structure. Regardless of all differences, we are dealing with a domain, a system of representation, and the correspondence between domain and the information which is expressed in the system of representation. For instance, the domain can be health, the information consists of numbers, and the correspondence can be expressed like “there were 178 cases of malaria in the community.”

For all representations, we can identify ideal types of competence which only concern the information or which only concern the correspondence between information and domain. We will explain this through an example.

Consider the data in Figure 28, where the number of malaria cases for four areas are represented in a line graph. The data has been typed in some computer system, which has generated the graph.

Figure 28. Information representing health issues in four areas in a district.

5.1. Representation competence

Reproduction skills. Emi has typed the data and made the computer system produce the graph. She knows nothing about the domain represented here, namely malaria epidemics. Also, she is ignorant about criteria for data quality like consistency and completeness, so when she checks the output from the system with her data source, she checks all numbers and does not consider the data for October in Riverside in particular. Emi has reproduction skills.
**Representation system skills.** Ibrahim is also ignorant about malaria epidemics, but he is a statistician, so he notices the trends of less malaria during April-October. He also sees that the data for October in Riverside is an outlier, so he checks that he has actually typed this number in accordance with the source he has received from the health personnel. Since he is fluent with the type of representations used but not necessarily with the domain being represented, we say that he has representation system skills.

Representation competence concerns the system of representation, and not the domain being represented or the correspondence between these. Astrud, who speaks Portuguese, has the representation system skills when typing a report in Portuguese from a script, since she can correct the grammar and the spelling. However, since she does not know English, she would not be able to spot misspellings in the original when typing an English manuscript, so she would only have the reproduction skills when typing English with a word processor.

**Interpretation** of representations takes us from skills to understanding:

**Representation system understanding.** If Astrud in addition to correcting grammatical errors also can explain what the grammatical rule says, she understands the structure of the Portuguese language, or more general, the representation system. Similarly, Ibrahim would also be at this level if he could explain the statistics of outliers, referring to standard deviations, etc.

### 5.2. **Correspondence competence**

Correspondence competence deals with the ability to make the information match the reality to be represented.

**Validation skills.** Janine knows the malaria epidemics, so she is familiar with the trends of less malaria in the dry season. Seeing the line graph, she explains:

> During the dry season, there are fewer mosquitoes, hence less malaria. That is why the number of malaria cases are low during March to October, and high during the rest of the year.

Janine knows that the variation in malaria incidents for Riverside from September to November is impossible, so she identifies the October data as an error. She could correct it by entering a number for October in between the values for September and November, but then she would utilise representation system skills. Validation skills would require that the error is rectified by aligning the information with reality. In this case, that would require a recount of the patients being treated for malaria in Riverside in October.

In case of information which signifies activities or transactions to take place, the correspondence between information and reality would be achieved by changing the reality according to the information.

**Validation understanding.** Again, the understanding level requires the ability to express the idea of validation. Instead of recounting, Angelique contacts those who have produced the
data source, discusses the mistake and arrives at a plausible figure. Being able to bring the representation in line with the reality through expressing her concern about the clash, Angelique has the validation understanding.

We will take a quite different example, which is uploading a video onto YouTube. Assuming that you are familiar with uploading files, YouTube does not surprise you, so your analogy with previous applications works out. After uploading has completed, some fields to fill are displayed, see Figure 29.

![Figure 29. a) Fields to fill about a video which has been uploaded. b) The available categories to select.](image)

The fields to be filled need validation skills, so that the information entered corresponds with the video. Also representation system skills are needed, so that the title and the description are expressed according to grammar and style. Selecting Category is done from a menu of fixed
choices, so for this selection the reproduction skill of selecting a menu choice is needed in addition to the validation skill of selecting the category which fits the contents of the video.

Title and Description might be obvious, but what is a Tag, and how does it differ from Title, Description and Category? And what does Sharing options mean? Tags and Sharing options seem to be information about videos, thus representing the domain of this web site. However, the person who has never uploaded a video before may not even be at the level of validation skills for these fields.

Tags are not just information about videos; ‘tags’ is also a computer concept. Interpretation and reflection on IT concepts could be used for characterising learning of any concept which is implemented in a computer. The skill of tagging would be the ability to write something in the Tag field. Functional understanding would be the ability to say that initially, the video has no tag, and after having typing something, is has a tag. At the level of IT understanding, the user should be able to use the tag concept when talking about other aspects of videos, for example saying that tags are used to alter the video appearance and format.

The example illustrates that for characterising levels of mastery of information which represents domains; also the learning of IT concepts model may apply in addition to the representation competence and the correspondence competence.

Correspondence competence as described above assumes that there is straightforward connection between information and a phenomenon in the world which the information represents. For many reasons, this assumption has limited validity.

Consider the Indian tourist Ravi, who during his visit to Egypt publishes the picture in Figure 30 with the caption in his Facebook albums, so his friends at home can watch. Ravi assumes that his friends recognize that the construction in the picture is actually not Taj Mahal, with which they are familiar, but the pyramids in Egypt. Further, Ravi wants to say that the pyramids are as magnificent as the Taj, and he hopes that the caption will communicate this. In general, we use a word which has a conventional meaning in a different way, and hope that the context will be sufficient for the reader to get the intended message anyhow. This example is from leisure life, but we find plenty of examples of metaphors and irony in business also. Metaphors require validation skills, not for adjusting the information or the reality, but for adjusting the information into something that makes sense compared to reality.
When the figure in the accounting system does not match the one on the receipt, it is unlikely that the accountant intended the reader to interpret anything else than the figures recorded. It is either an intended or unintended mistake; the former would constitute a lie. Mistakes or lies which are out of the normal range might be found through representation systems skills or validation skills, but when the errors are smaller, routinely comparing the information against reality is the only way of finding out.

Normally, the author of a novel intends the book to be a complete lie, in the sense that there is no corresponding reality to the information provided in the book. Since the reader is aware of this, nobody gets confused. In some cases, readers may start discussing a fictional universe which the books refer to, in which case we can say that there is a correspondence; not between information and reality, but between information and an imagined world. Then we can consider correspondence competence in such a setting also.

Most databases are intended to contain accurate representations, and in general they may actually do. However, quantities of data are always prone to small errors and incompleteness, which is acceptable as long as the essence is correct. For example, an address list may state that a person lives in New YorK, and the post office might send the parcel to New York. Assuming that the I was a typo for Y, the redundancy in the name enables the reader with sufficient representation systems skills and validation skills to correct the error.

A painting of a person is a representation of this person, informing the viewer of the person. Completely abstract, or nonfigurative art does not represent anything, so it is not informative, hence it is not information. Similarly, absolute music, for instance Beethoven’s 7th symphony, is purely sound, without anything intended to be represented by it, not even in the title of the work or in the individual movements. The title is just informing about the form of music and how it is to be played, not about anything outside it. However, painting and music could be used for representing something, so the system of colours and tones could still be found in the
art; hence representation competence would be relevant. When The Beatles is performing “Yesterday,” the lyric is informing us, possibly in the same way as the fiction author does.

Assuming that correspondence is relevant, it may be breached in three ways. Information could be incomplete, inaccurate, and out-dated. Validation skills are needed for all these three aspects in order to obtain complete, accurate and current information.

5.3. A representation system – slide design

There is a large number of representation systems in daily use. In addition to spoken and written natural language, there are musical scores, dance notation, chemical formulas, knitting codes, and programming languages, to mention a few. We will present some principles and concepts for a representation system which is also useful in user training, namely slides. Being able to explain principles for slide design corresponds to the representation system level of understanding for slides, or ‘slide design understanding’ for short.

Slide design depends on other ways of representing the world; it brings in the written language and all possible kinds of illustrations. Using already known representation systems makes slide design appealing, since people do not consider that designing slides require any additional competence. However, this is also the pitfall of slide design; people use PowerPoint or Impress as if it were a word processor with page organisation of the text and easy manipulation of the format and figures. We use the outputs of presentation programs for accompanying a presentation or for displaying a slide show on its own, and both of these differ from the purpose of a written text, which is meant to be read at the readers’ speed. Also, each of the two purposes of a slide show has its own design rules, and we will in this section address the design of slides which accompany an oral presentation. We will come back to design guidelines for design of independent presentations like slide shows in Error!

Reference source not found.

A main reason for using slides in a presentation is that it allows for more than one way of presenting the material at once. Since some people learn better by hearing, others through reading and still others through seeing a figure, all three groups in the audience can be satisfied at the same time. Moreover, most people learn even better through a combination of impressions, so that presenting in oral, written text and figures at the same time is advantageous for all the audience. This brings us to the first guideline:

1. Combine text and illustrations.

When we are listening to a presenter at the same time as reading the text on the slide, we would easily loose out on one or the other. In order to minimize this fall out, we would normally write text which reads very fast, and this would be seven words in one line. This line should present the essence of what the presenter is saying in a few sentences, which could constitute a paragraph if written. Copying full sentences from a textbook or a web page onto a slide, which some presenters are doing, is therefore a dysfunctional way of using slides. The slide in Figure 31 illustrates the result of copying full sentences into a slide. In order for the
audience to grasp the message, the presenter has to read aloud the full text. The main message of this slide is rewritten in Figure 32. The text on the slide can be read in five seconds, and the design illustrates the process of producing the letters. After pointing to the four elements in this slide for explaining the essence of mail merge, the presenter can subsequently tell about how the list of recipients is stored and other details from the full text.

### Mail merge

- **Mail merge** is a software function describing the production of multiple (and potentially large numbers of) documents from a single template form and a structured data source. This helps to create personalized letters and pre-addressed envelopes or mailing labels for mass mailings from a word processing document which contains fixed text, which will be the same in each output document, and variables, which act as placeholders that are replaced by text from the data source. The data source is typically a spreadsheet or a database which has a field or column matching each variable in the template. When the mail merge is run, the word processing system creates an output document for each row in the database, using the fixed text exactly as it appears in the template, but substituting the data variables in the template with the values from the matching columns.

**Figure 31. An inappropriate slide design. Text from Babylon Online Dictionary.**

**Figure 32. One point per line. Graphics illustrating a process.**

In some cases, for instance when presenting a quote of a couple of sentences, we need to display the full text on the slide. To avoid fall out in such a case, the presenter should read the quote in full, so that the oral and visual impressions are synchronised. In general the guideline is:

2. Write each point on one line.

Simplicity is also an advantage concerning illustrations. They should display the essential of the point and avoid disturbing details. If the point of the illustration is to show the reality, a photo is appropriate, but unnecessary surroundings should be cut off to avoid distracting...
details. If the point is of a more abstract character, a drawing is better suited for communicating the essentials and avoiding the disturbances. In summary:

3. Keep illustrations as simple as possible

Text and figures displayed on a screen may look large for the presenter, but the audience in the back of the classroom may have trouble reading the text. To ensure legibility, use minimum 18 points font size and sans-serif typeface (Figure 33), since these are clearer when displayed on projectors than the serif fonts.

![Figure 33. Typefaces with and without serifs.](image)

Slides are also often printed as handouts and reduced in size to accommodate more slides on one page. Font size 14 on the original will then become tiny and difficult to read for the long-sighted, while the near sighted have trouble reading 14 point size on the screen. The conclusion is:

4. Minimum 18 point font size with a sans serif font.

The other factor which affects legibility is the contrast between the text and the background. Black on white or white on black are safe, but nearly all other combinations are reducing legibility. Also backgrounds which can be chosen in a commercial presentation program may hamper legibility. The yellow marker colour is the only background colour which actually improves legibility, and therefore it should be used for emphasizing. Black letters on a light blue background may help dyslectics, and this combination is also fine in general. Visibility of figures also require sufficient contrast, and even if the contrast looks good on a screen, a projector might require a larger difference between light and dark in order to deliver easy to see pictures. So,

5. Keep contrast between text/graphics and background close to black versus white.

More thorough introductions to slide design can be found in (Duarte, 2008) and (Reynolds, 2010). Many guidelines can also be found on the web, for example at SlideShare.

**Exercises**

1. Identify elements of Reproduction skills, Representation system skills and Representation understanding for the representation systems
2. Provide examples of Validation skills for the information in
   - Credit cards
   - Cinema ticket system
   - Music database
   - eBay or another e-auction

3. Consider the following pie chart and the comments to it. Characterise the comments according to their levels of representation and correspondence competence if possible.

   Domenico:
   *So the biggest proportion here is the total. That is like it should be.*

   Julia:
   *Oh, that is a terribly high failure rate. What can we do about it?*

   Kylie:
   *This diagram is completely rubbish. What was the purpose?*

4. Make your fellow students explain when the following events will take place the next time:
   - Leap day (29th February)
   - Vernal equinox
   - Easter
   - Eid ul-Fitr (Id al-Fitr)
   - One of the Hindu New Year’s days (Gudi Padwa, Ugadi, Vishu, Bihu)
Characterise the responses according to their levels of representation and correspondence competence

5. Qing is an accountant, paying expenses and salaries in a construction company, saying:

*I always get approval from somebody for paying expenses. The other day, I noticed an invoice of 3 million, which is way above our normal payments, so I checked with the issuing company.*

At which levels of information competence is Qing? Consider both her representation and her correspondence competence.

6. For the case of video uploading on YouTube, write explanations of

- Tags
- Sharing options

at the levels of representation system understanding and validation understanding.

7. Assess the following slides according to the guidelines given in this chapter.

**Introduction**

- The change must be compatible with existing cultures, and the organizational needs, Otherwise user of the new implemented information system may resist its use.
- Introduction of a new information system requires institutionalization of a new kind of culture.
- An implementer of an IS must be aware of the types of change, the impact of change to the user and the organization .
- In these case where user support was not adequately provided there might be resistance towards a new implemented IS use
8. Compare the slide design guidelines in this chapter with other guidelines which can be found on the web. See e.g.

- TechRepublic  

- EllenFinkelstein.com  

**Project**

9. Identify the information competence which you need to include in your training. Also, determine the information competence, which your teaching will build on. Which level of representation system and correspondence competence should be the learning goal of your session?

Define the information competence to be learnt and the targeted level of learning. Present your definitions on a slide.
Chapter 6. Understanding task support

The learning aim of this chapter is to be able to determine the tasks for which the IT taught will be used, and to identify levels of understanding of using IT to support tasks.

6.1. Levels of mastery of using computers in work tasks

Sein et. al. (1998) and Coulson et. al. (2003) have proposed levels of knowledge of software use, where an important distinction goes between competence of the software and of its use. Their distinction is similar to the distinction in this book between competence on IT and on using computers in tasks. As mention in Chapter 3, they do not consider the competence in the information area.

The previous literature considers that a basic competence is knowing how to use IT in one’s work tasks, and a more advanced competence is to see what IT does for the organisation. The Committee on Information Technology Literacy (1999) also considers the societal impact as part of IT competence. One dimension of competence on IT use is therefore its scope, extending from the individual through the organisation and into the societal level.

Sein et al (1998) distinguished between the ability to apply an IT system and to see what else the system can do. This will be expanded here to include the ability to suggest changes of the IT to fit work, organisation and society.

The competence of using computers will have both an individual and social dimension. The mastery will be considered at three levels:

Skills for use in tasks. The skills of using IT to support activities, both individual action and social interaction.

Understanding current situation. The ability to explain how IT supports activities and other consequences IT has for own tasks, the organisation and the society. This includes understanding of why an application is useful or not, which seems to be the most important factor affecting learning and using the application (Davis, 1989).

Understanding possible changes. The ability to explain what IT can do, how IT systems can be changed, and how tasks can be changed in the future. This level also includes the ability to predict changes of IT and of how IT, organisation and society might affect each other.

We will look at some examples of people at different levels of mastery. When asked about the impact of IT in her workplace, Kirsten replies:

I inform customers about new services and change of terms, and I remind them about appointments. When doing this, I look up the customer’s history in our computer system and then write them letters in Word. Thereafter I send the letters off in Thunderbird.
Kirsten is telling about her own work and is providing some information on which IT tools she is using for three tasks. She is not showing any motivation for using the IT tools in the sense that she can point explicitly to their usefulness, for example by comparing these tools with other solutions. Her reference to “our computer system” is rather vague, and she is not telling anything about other consequences of IT for her work or the organisation. Although she is able to tell a little bit on how IT supports her activities, she is not at the Understanding level yet.

Leonard, being asked the same question, replies:

*The corporate database means a lot to my work and to the organisation as a whole. Now I enter all incoming mail in the system, and if it is on paper, I scan it. That means that I can search everything that is there and also that people in the claims payment department has the information at once. Delays due to waiting for papers to be transferred or finding the case in the archive have been eliminated. Besides, it gives us the up to date information on how we are doing, so that there is no longer any argument with management on productivity measurements. I see this as a win-win-win situation for us, management and customers.*

This explanation is telling little about Leonard’s skills, but it demonstrates that he has understood consequences for his own work, for colleagues in other departments, for management, and for customers. He demonstrates an Understanding of the current situation and compares it to the previous system. His enthusiasm of the usefulness of the corporate database demonstrates a clear motivation for using the system.

A study of users’ perceptions of manuals found that they want documentation to include more than how to carry out a specific task by means of a software package (Scott, 2006). They also want the manual to include how the task which the software supports relate to other tasks carried out by themselves of colleagues.

Melly says:

*After we got the basic patient information in the computer system, I don’t have to waste my time waiting for the patient record any longer. I look forward to the time when the record in the computer is complete with x-rays and attached documents, but we can do most of what we need by the diagnoses and lab info which is there now. Also, when we receive a patient from Jakarta Central, they also send us the patient info, so that before the patient is here, we know what to do.*

*The main trouble with this system has been the security. You have to log in here and there, and after 20 minutes of idle time, you are logged off. If a nurse has logged on in the meantime, we were stuck, and had to find her to log off before we could access the data. We discussed this with the IT guys several times, but they said it was a policy decision such that medical data should not be spread to those who have no rights to see it. However, in my opinion, it is more important that the medical staff that needs the information gets it than that others are denied access. That means that we give*
priority to providing the right medical treatment rather than protecting the patient’s privacy. The latter will never cure their illnesses. So since the IT people did not help us out, we found out that all staff in the department should have the same password. Thereafter we have had no trouble opening the system when needed.

Melly demonstrates understanding of how IT systems affect her and the organisation, how the systems can and should be changed, and even the ability to work around the rules set up. She is also arguing about a societal issue like the access to data versus privacy.

6.2. Usefulness

Understanding current situation mean to understand how IT supports activities and other consequences IT has for own tasks, the organisation and the society.

One of the few, well documented connections within use of IT in organisations is the Technology Acceptance Model (TAM). In its original form, it says that the usefulness of a technology is the strongest factor concerning whether the technology will be used, while its ease of use and learning is of less importance (Davis, 1989). TAM predicts that if a computer software is experienced as useful by the users, they will use it, even if they have to put effort into learning it. On the opposite side, a system which is easy to learn and use will not be used if the users do not experience that it is useful for their tasks.

The model is illustrated in Figure 34, where the bold arrow indicates a connection that is stronger than the other one.

![Figure 34. The Technology Acceptance Model, original version, adapted from (Davis, 1989).](image)

An example: At a time when computers were not everywhere, a hospital installed a computerised encyclopaedia for nurses in one ward, where they could find information on care procedures, medical explanations, guidelines, etc. The nurses had previously experienced that their questions were not always answered in a polite way, and that looking ignorant in front of superiors was a bad experience. Therefore, they quickly adopted the system to avoid having to ask doctors or administrators for help. The system only had one terminal, and after a
while, this terminal was moved to another ward 5 minutes walk away. Despite this extra time, they continued using it. In order to reduce disturbances, they organised a buddy system, so that one nurse collected the questions and walk over to the other ward, while the others took over her tasks during the half hour needed. Thus, they added lots of additional effort in order to achieve the usefulness which they experienced.

Measuring the result of use of IT systems in organisations in general has shown impossible. Normally, you cannot isolate the costs of technology implementation, and you cannot isolate their effects. Expenses are interwoven with the costs of learning and changing work processes, and correspondingly, the products and services produced by an organisation depends on a package of factors, including competence, infrastructure, and the market. TAM therefore measures the degree of success in the time the technology is used, and this kind of measurement has over the years become a standard for measuring technology acceptance and success.

Later on, TAM has been refined with more factors, and a combined model looks like Figure 35.

![Figure 35. The revised Technology Acceptance Model, adapted from (Venkatesh et al., 2003).](image)

When colleagues use the system or your boss tells you to use it, the social influence is increased. Facilitating conditions concern accessibility, including network connection, electricity, printers, etc.

When evaluating proposed IT systems, the model tells us about four factors to consider. Beware that it is the prospective users’ opinions which matter. If outside consultants do not see the point in a software package, while the users do so, the system will probably persist.

© Jens Kaasbøll, 13th April, 2012
On the other hand, if the consultant thinks that some information produced by the system will be very valuable for the organisation, while those working there do not share that opinion, we cannot expect that they will take the effort of learning and using a new system.

In addition to the four factors, which seem to be relatively stable, other factors will moderate the picture. For a young man, the usefulness will be more important than for others, while for an elderly lady with little IT experience, ease of learning will count more than for others. Such moderating factors may depend on the local culture, and the studies behind TAM are mainly carried out in North America.

Based on TAM, learners who have not understood the usefulness of a specific IT will be less likely to learn its operation. Teaching usefulness should therefore precede teaching skills.

6.3. The organisational level

Nusrat is working on designing dams by means of a geographical information system. His reply is:

*When I get the right data from the surveyors, I can plan exactly where to place the dam and how to construct it, so that we can control the river. However, the data is often inconsistent. They are making too many mistakes out there. With correct data, we can get the dam built straight away. But now I have waited for six months to get this right, and I have called them several times lately to remind them to send me accurate data. I clearly see the need for automatic transmission of data from their instruments through wireless modems to my computer, so that we can avoid all these errors, so I proposed this equipment for the next year’s budget.*

Nusrat has understood how he can design a dam by means of his GIS, and the six months waiting for data indicates that his motivation to use the system is high. He sees possible outcomes of what he is doing, and he has plans for improving the system. In this way Nusrat is at the level of Understanding possible changes of IT for his work. However, his attitude towards the surveyors also points to some lack of understanding organisational issues. Nusrat is the one benefitting from the data entered, while others are doing the data entry job. He has seen the usefulness of the system, while the surveyors have not, so they are less likely to make the extra effort. Also Nusrat’s conviction that once he has designed the dam, it will be built, points in the direction that his competence of IT for his own job is at the level of Understanding possible changes, while when it comes to organisational issues, his understanding is limited.

Since application of IT in work practices, for leisure, education or any other activity is a multifaceted endeavour, most of us might be like Nusrat; we might understand some issues very well while being ignorant of other. The three level model of externalisation of IT use might therefore have to be qualified concerning various aspects.

During an implementation of an enterprise resource planning system, most users avoided learning it in the beginning and left the data entry to a few super-users (Boudreau and Robey,
The majority did not see the usefulness for their job, and they were not convinced that it was an advantage for the company. Learning it became a burden which added to their normal, busy day. Usefulness according to the Technology Acceptance Model is an individual experience, so usefulness for the organisation does not imply immediate learning.

Research points to that when we work together with tasks which depend on each other, training is more important for our adoption of IT and information systems than when adopting a single user application (Reynolds, 2010, Sharma and Yetton, 2007). So, for people to use an enterprise wide information system appropriately, training is more essential than when downloading an app to a phone, even when the technical challenge is at the same level.

6.4. Summary

People become motivated to learn new IT when they understand its usefulness for the tasks they are carrying out. Also understanding how the technology and the information are embedded in the organisation is advantageous for user learning.

4. Make sure users understand the usefulness of the IT.

Exercises

1. Arja is a teller in a store. When asked about how she uses the computer, in her work, she responds:

   I scan the goods, then tell the customer the total, receive money, and return the change, which I read from the computer.

   On which level of competence for using IT in tasks is Arja?

2. Arja’s boss is Shakira. She is telling this about the IT system in the business:

   I use the system to supervise our sales, and in particular I look out for new trends amongst our customers. When I see that some brands or types change more than the seasonal variations, I change the place of the goods in the store, organise a sale or start a campaign. Without the computer system, we would have seen the trends much too late compared to our competitors. Also, I observe the salespeople’s achievements, and I bring this information to the weekly pep talks. I wish that we would get direct computer links to our main suppliers, so that we could avoid all the operations of preparing an order and sending it off.

   On which level of IT user competence is Shakira? Consider all three subject matter areas.

3. Consider a reasonably complex IT application in an organisation. Ask your fellow students to explain it. Note down what they say as precisely as possible. On which level of IT user competence is your fellow student for this topic? Consider all three subject matter areas.

4. Grudin (1994) argues that one challenge to introduction of groupware is that those who get the benefits are not necessarily the same people as those who have to put
additional work into the systems, including data entry. This argument might also hold for other collaborative IT systems. What are the challenges for learning in such cases, and how can these be overcome?

5. Find two or three examples for usefulness of the following subjects:
   a. Table of contents in a document,
   b. Line graphs in spread sheets,
   c. URLs as displayed in hits from search engines.

6.5. **Project**

6. Find at least two purposes which can motivate learning your selected topic.

*Write a question which triggers the audience to find other purposes.*
Chapter 7.  Training for transfer

The learning aim of this chapter is to be able to design a training session which maximises the possibility for learning both skills and understanding.

Training is the organised activity where one or more trainers teach a group of trainees. The aims, contents, time schedule and classroom are determined in advance, normally by the teachers with little influence from the learners.

The need for training when implementing information systems was emphasized at the end of the previous chapter. Likewise, training seems to be essential for developing adequate mental models. For instance, 80% of Greek high school students equals Internet and www (Papastergiou, 2005). The students’ models are simplistic and utilitarian, rather than structural. Those students who had been presented conceptual models in school held more sophisticated, structural models.

When trying to manipulate a system with unknown behaviour patterns and no external help sources, the good learners generated mental models of its operation, often close to a functional understanding (Furuta, 2000). The few accounts which the poor learners came up with indicated skill level rather than understanding, e.g.

*I am turning the left knob to the right (clockwise).*

A control group, who were provided with the theory of the system, carried out the task in a significantly shorter time than those who had no external help (Furuta, 2000). This implicates that poor learners are more in need of theory than those who generate mental models more easily. Another study also concluded that training is more efficient than self exploration (Simon and Werner, 1996).

Some project managers may believe that after users are trained, they will master the application and use it. This assumption is in general false, for several reasons. First, the users might not find the application useful, and therefore not use it even of they know how to, as discussed in the previous chapter. Second, they might not have learnt what the teacher intended, and third, even if they became skilled during the training course, they might have

![Transfer model of training](image)

*Figure 36. A transfer model of training. Competence is learnt in a course and transferred to work or other settings.*
forgotten it when times come for applying it. The problems of transfer of competence from courses to the activity where the competence is to be used have long been acknowledged.

Seen from the training perspective, learners have some competence when starting up, and hopefully a bit more when completing. This new competence is what the learners will bring back to the tasks. We can illustrate the process as in Figure 36.

Timing of training is possibly the most important factor to transfer success. If training is provided weeks before the IT system is put into operation, the users will forget much of their new competence, so the possibility for transfer has decreased seriously. (Finnegan, 1996, Karuppan and Karuppan, 2008). On the other hand, if training is provided long after a new system is installed, the users may say:

*Why training this late? The system has been up for ages, but nobody knows how to use it. The management is not giving appropriate attention to this change.*

Users will bring this negative attitude to the training session and blame the trainer for the delay and mismanagement. Less motivation for learning and a bad atmosphere will impact the course.

The golden rule for user training is therefore:

**5. Organise training at the same time as the system is installed.**

The trainer needs to identify the start and ending level of competence for the training. Determining the beginning level serves two purposes. The trainer will know where to start, and also, the trainer can relate new topics to what the learners are familiar with already.

Understanding is in general known to enhance transfer of competence from courses to work (Bransford, 2000). The benefit of conceptual knowledge for transfer has also been demonstrated in research on computer training (Yi and Davis, 2003). Improved understanding also helps users remember longer during periods of not applying a software (Karuppan and Karuppan, 2008). We will therefore adopt a training method which also targets understanding.

### 7.1. Conceptual-practical training

Based on research on teaching and learning, Gagné and Briggs (1974) proposed a nine step model which in general would be the best design of a training session which aim at both skills and understanding, see Table 7.

**Table 7. Gagné-Briggs sequence of instructional events.**

<table>
<thead>
<tr>
<th>Instructional event</th>
<th>Purpose for the learner</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Gain attention</td>
<td>Focus on trainer</td>
</tr>
<tr>
<td>2. Inform the learner of the objective</td>
<td>Focus on the learning goal and the usefulness of this goal</td>
</tr>
</tbody>
</table>
### Instructional event

<table>
<thead>
<tr>
<th>Instructional event</th>
<th>Purpose for the learner</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Stimulate recall of prerequisite competence</td>
<td>Retrieve prior learning</td>
</tr>
<tr>
<td>4. Present material</td>
<td>Perceive essential elements</td>
</tr>
<tr>
<td>5. Provide guidance to the learner</td>
<td>Perceive how the elements can be operationalised</td>
</tr>
<tr>
<td>6. Elicit performance</td>
<td>Develop basic skills through own actions</td>
</tr>
<tr>
<td>7. Provide feedback</td>
<td>Reinforce adequate behaviour and correct the inadequate</td>
</tr>
<tr>
<td>8. Assess performance</td>
<td>Rehearsals for enhancing skills</td>
</tr>
<tr>
<td>9. Enhance retention and transfer</td>
<td>Combine usefulness, essential elements, and skills</td>
</tr>
</tbody>
</table>

Although this sequence is designed to be generally optimal, the actual subject to be learnt, the learners and the setting, may call for different events and sequences. Concerning IT user training, the only instance of instructional development documented is found in (Herskin, 2006). Although Herskin’s book does not provide references, the sequence of instruction presented follows the Gagné-Briggs model.

Herskin’s model has been tested (Hadjerrouit, 2008), and the learners responded:

*It helps understanding.*

Similar approaches aiming at teaching skills and understanding of IT have also been found effective. (Bhavnani et al., 2008). In addition to teaching in the US, training for skills and understanding was also carried out by another researcher in a disadvantaged community in South Africa, where few learners were familiar with computers (Marsh, 2007). These learners also benefitted from the teaching method. For example 39% of the students who were taught about split window in Word for viewing several relevant parts of a document at the same time also applied the same strategy in Excel, without this being mentioned in the Excel teaching. The students transferred the competence nevertheless.

The Gagné-Briggs model separates five different types of capabilities, intellectual skill, cognitive strategy, information, attitude and motor skill. Correspondingly it shows us five different types of each of the nine instructional events. Cognitive strategy concerns problem solving, and this will be addressed in Chapter 8.

This book will not address motor skills. Use of IT is primarily an intellectual skill. Therefore the sequence of instruction presented in this chapter will mainly address the intellectual side. Also the attitude towards the information technology is important for the ability to transfer intellectual skills from courses to work. This will be included specifically in events no 2 and 9.

Information capacity in Gagné-Briggs’s sense concerns the learning of facts. Knowing some facts is also useful for transfer. The learning of simple facts will be included in the skills
practice in event no 6. Information competence, in the sense of the ability to interpret and present information, was considered in Chapter 5.

Table 8 shows a sequence of instructions for teaching one IT topic to a class. The sequence should last 30-60 minutes, and the practicals, events 6-8, should make up for at least half of the time. Event 1 necessarily comes first, and event 5 links the theory to the hands-on activity. Therefore this should be the last event in the introduction. The contents of events 2-4 may change order, depending on the teaching topic. Event 6 introduces the practical training, while the invocation of events 7 and 8 is determined by the individual or small group initiative and progress.


<table>
<thead>
<tr>
<th>Instructional event</th>
<th>Teaching use of IT</th>
<th>Example – Setting up a link</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Gain attention</strong></td>
<td>Any quick action which draws the learners’ attention to the teacher. Not necessarily IT related.</td>
<td>Clapping hands.</td>
</tr>
<tr>
<td><strong>2. Inform the learner of the objective</strong></td>
<td>C1. Explain the IT concept / idea / principle to be learnt. U1. Explain the usefulness of this idea for the tasks in which the IT is going to be used.</td>
<td>C1. Explain import spreadsheet by link into a document. Explain that when data needs to be changed, updating the spreadsheet is enough, inconsistencies are avoided. U1. Present an example where a report under preparation contains a table of figures and calculations, and that the figures are not complete yet.</td>
</tr>
<tr>
<td><strong>3. Stimulate recall of prerequisite competence</strong></td>
<td>C2. Compare the new IT idea / principle / concept with previous ones. C3. Compare also with experience from phenomena outside IT.</td>
<td>C2. Present how setting up a link differs from copy and paste, how it is similar to hyperlinks, and C3. How it differs from the copier machine, and asks learners for other comparisons.</td>
</tr>
<tr>
<td><strong>4. Present material</strong></td>
<td>C4. Explain the principles of how the data is stored and processed by the computer</td>
<td>C4. Explain how setting up a link is stored in the hard disk.</td>
</tr>
<tr>
<td><strong>5. Provide guidance to the learner</strong></td>
<td>C5. Present steps with observable results in the procedure for carrying out the operation.</td>
<td>C5. Present the steps in the procedure for importing by link.</td>
</tr>
<tr>
<td>Instructional event</td>
<td>Teaching use of IT</td>
<td>Example – Setting up a link</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>6. Elicit performance</td>
<td>S1. Develop basic skills through own actions</td>
<td>S1. Give the learners an instruction sheet or video with the detailed prescription of how to carry out the operation on the computer.</td>
</tr>
<tr>
<td>8. Assess performance</td>
<td>S3. Observe learners’ progress of task. C6. Elicit their understanding by asking about the state of their data. S4. Ask them to repeat if necessary, possibly with small variations.</td>
<td>S3. Check whether the learners have identified the data and created a link. C6. Ask about where the data is located and which links they have created. S4. Ask them to repeat the operation if not carried out correctly.</td>
</tr>
<tr>
<td>9. Enhance retention and transfer</td>
<td>C7. Continue assessment by asking the learners to express the concepts taught. Repeat and extend S5. skills, C8. data storage and processing, and C9. Concepts. C10. Combine the IT concepts with similar ideas / principles / concepts. U2. Repeat the usefulness with reference to tasks.</td>
<td>C7. Ask the learners to express copy-paste vs. setting up a link. S5. Repeat the steps in the procedure and ask about other ways of doing it. C8. Repeat how the data is organised in the hard disk, and ask about how this can be observed. C9. Repeat that setting up a link is similar to hyperlinks, and C10. Introduce the principle of functional dependency as a more general concept. Ask for other instances of this concept. U2. Repeat that inconsistencies can be avoided and ask about work tasks where this may be advantageous.</td>
</tr>
</tbody>
</table>

Compared to the Gagné-Briggs model, the IT training requires the attention of more elements in the events. Understanding the usefulness is an important motivator for learning. This is introduced in event 2 as U1 and repeated in event 9, see U2. Likewise, understanding the IT concepts is essential and often non-trivial. Therefore this is treated nine times in six of the events. Teaching for learning skills is the third type of element here, S1 – S5, mainly addressed in the middle section, the hands-on part.

Understanding the usefulness and the concept, increases the chances of transferring competence from course to work. The clue for transfer of skills lies in the instruction sheet or videos in event 6. When the learners bring the instruction sheets back to their workplace, they...
may be able to operate the software by means of the same instructions at work as in the course.

Training material for the example indicated in the rightmost column of Table 8 will be spelled out in the sequel. The material is annotated by double encircled codes U1, etc., in order to show which element in the instruction sequence the material is targeting.

**Importing spread sheet into document**

- **Aim**
  - Able to choose copy-paste or setting up a link
- **Problem**
  - You are to write a report on the health for last year
  - The numbers are not complete yet
  - You have to start writing nevertheless
  - Numbers in a table in a spread sheet
  - Table should be part of a document
  - What happens if data in the table needs to be changed?

**Figure 37. First slide for learning setting up a link. Emphasis on usefulness.**

This introduction opens with presenting a case for which the operation can be useful, U1.

**Figure 38. Repeating known principle. Illustrating the hidden data on the hard disk.**
Instead of presenting the new concept first, this introduction starts out with what may go wrong with the operation already known by the learners. It also illustrates the relation between what is shown on the screen and the how the data is stored on the hard disk, see Figure 38. The essence in this case is understanding how a new concept relates to one which is previously known, the copy-paste. The results of both operations look the same on the screen, but how it is stored on the hard disk differs. For supporting this understanding of hidden data structures, the relation to the visible is illustrated in Figure 39.

In order to provide a concrete illustration of importing by a link, the teacher could present the dynamics that the link offers. The teacher could display an already prepared solution on a projector, and demonstrate that when changing a number in the spread sheet, the document is changed accordingly. The point of this demonstration is to show the benefits of the result; not to learn how to do it. The ‘how-to’ is for the practical part, therefore the teacher is not to tell which buttons to push.

**Setting up a link**

- One table on the hard disk
- Link
- Changes in spread sheet propagated to the document

![Figure 39. The way setting up a link is stored in the hard disk.](image)

© Jens Kaasbøll, 13th April, 2012
Procedures

Copy-paste
1. Open both files
2. Copy the table and paste it into the document
3. Check by changing the spreadsheet

setting up a link
1. Open both files
2. Insert the table into the document such that the link to the spreadsheet is kept
3. Check by changing the spreadsheet

Figure 40. The procedure for the operations at the computer.

The procedure for carrying out the operations is divided into three steps (Figure 40). The separation follows the fixed rule that after a step, the learner may observe on the computer whether it has been successfully carried out. This will ensure the learner if being on the right track. If not, asking the teacher may be appropriate.

After the copy operation, the computer does not provide observable feedback about what has been copied. That is why copying alone does not constitute any step in the procedure.

During introduction and summary, instructions for particular software installations are avoided in all written materials and presentations. There are three reasons. First, the introduction and summary should focus on understanding. Second, the strict separation allows learners using different versions of software to be taught, as long as the teacher has instruction sheets for all versions. Third, a modularised design of training material is achieved. When a new software version appears, the material for introduction and summary can be kept untouched.

Being able to view the instruction videos or sheets at the same time as the software to be learnt is easier than hiding them behind each other. The teacher should therefore tell the learners to reserve one part of the screen for the instructions. Reminding the trainees to stop and start the video according to their own speed of working is also useful for easing learning.

When learners work on their own, the teacher can move around freely in the classroom to help out when needed. Instructing by means of the projector means that the teacher has to rush in between the screen and the learners to help out those who are stuck. Experience shows that working with instruction sheets or videos relieves the teacher from most of the requests, since the learners have material to look at. The teacher is gaining time to respond to the questions from the learners.

The slides for the summary focus on three topics. The concept of setting up a link, how this concept is a special type of a more general type, and its usefulness.

© Jens Kaasbøll, 13th April, 2012
Even if the teacher observes that the learners carry out the operations in a satisfying way, this does not mean that they have understood the concepts. The summary should therefore continue the assessment, by asking the students to express the concept they have been working with, for instance as C7 in Figure 41. If the targeted number of students do not respond properly, a thorough repetition of the concepts is necessary.

Figure 41. Repeating the difference between copy-paste and setting up a link in the summary.

After assuring that the learners gained a sufficient understanding, the topic can be taken to a more abstract level. Setting up a link is compared to formulas in spreadsheets supposed to be known to the learners. Both are illustrated as examples of the functional dependency principle, Figure 42. Understanding similarities and more general concepts ease the learning of new concepts, which is why we should aim at more general understanding. If the learners did not reach the appropriate level of understanding of setting up a link, they will not be prepared for the more general topic of functional dependency. In this case, the introduction of the general topic should be skipped.

Functional dependency is introduced directly after the import-by-link concept, skipping the data link level in Figure 21. Since explaining functional dependency might be easier with data link examples than with type-instance, it might be better at a later stage, after the learners are familiarised with type-instance, to introduce the difference between type-instance and data links, and the common functional dependency principle.
Functional dependency

- setting up a link
  Changes in spreadsheet → changes in document
- Table of contents
  Pages inserted → Table of contents updated
- Formulas in spreadsheet
  Arguments changed → Result updated
- Link from display to data storage
  - Data changed → Changes propagated to display

Figure 42. How setting up a link is a special instance of the functional dependency principle.

In order to improve the likelihood of transfer, tying the learnt concept to the users’ future practice is repeated. Therefore, asking the learners about where setting up a link can be useful or disadvantageous concludes the summary, see Figure 43.

Other use?

Are there other types of data than tables where setting up a link is an advantage?

When do we choose Copy-Paste anyhow?

Figure 43. Last slide in summary, triggering discussion about usefulness.

7.2. Learner activity

The Conceptual-practical training model mainly addresses the contents of the session and the role of the teacher. Also the learners’ activity is important for transfer, so we will point to ways to organise this.
During the introduction, making the learners to talk about cases where they have added spreadsheet tables in documents, serve two purposes. First, it triggers the learners’ attention to the current topic, and second, it informs the teacher about the learners’ competence in this area and thus how to address the explanations.

During the practical, obviously the learners are busy on their computers. This is necessary for developing the initial skill. Since talking while doing contributes to understanding, it might be better if the learners were discussing while running the computer. This can be achieved through having two learners at each computer. One is operating the keyboard and mouse, while the other is making comments, asking questions, checking the documentation etc. Then the pair swaps roles, so that both do the hands-on exercise. A laboratory study showed that learners operating in pairs outperformed individuals in their conceptual understanding. The pairs could explain more about how the software operated and they performed better in exercises which introduced some novel elements (Lim et al., 1997).

Operation in pairs is known from computer science education as ‘pair programming.’ Several studies have been carried out, and pair programming has been proved more effective for bringing more students through exams than the individual programming (McDowell et al., 2006). The lesson from learning of programming addresses learning goals of understanding and problem solving.

Motor skills were not considered, neither in the user learning nor the computer science cases. It would be reasonable to believe that learners, still struggling with the mouse and keyboard, should be practicing this as much as possible, while pair learning fits better for those above this level.

7.3. Summary
The Conceptual-practical training model emphasizes teaching for understanding of concepts and of usefulness as essential for being able to transfer competence from training to work. Also including instruction sheets or videos in training is beneficial for transfer. The need for organising training when new systems are introduced is also introduced as the fifth Golden Rule:

1. Provide users with detailed instruction sheets or videos, also during training.
2. Train users so that they understand IT concepts.
3. Provide a variety of learning material.
4. Make sure users understand the usefulness of the IT.
5. Organise training at the same time as the system is installed.
Exercises
1. Extend the case on setting up a link with two more examples for illustrating the usefulness of the concept.

2. Extend the case on import by link with two more exercises for the practical training, so that those who complete quickly can move on to the next one. Each of the additional exercises should include a small, new challenge.

3. The following slides are intended to cover the introduction part of a training session on indexes in documents, steps 1-5 in the Conceptual-practical training for IT users. Assess this introduction according to the principles presented in this chapter.

INDEX CREATION

PREREQUISITE KNOWLEDGE
– Familiarity with Microsoft Word Processor
  a) Main menu
  b) Tool bar
– Ability to navigate through a word document

OBJECTIVES
By the end of the lesson, learners should be able to;
• Mark index entries
• Create an index

Definitions
• An index is a directory arranged alphabetically to guide on the location of key words or phrases in the document.
• Key words are core items that surrounds and best explain the main topic

Mark index entries
• Highlight the key word or phrase then,
  - On the menu bar, go to “Reference” then “Mark entry” and on the other window “Mark”
  - Click where you want your index to appear.

Create an index
• After all entries have been marked, then go to “Index group icon” and click “Insert index”
  - This window will show, then do
    – Check Right align page numbers
    – Formats (select one)
    – Select type, Columns and Language
    – Accept all by clicking “OK”

Project
4. Develop a complete conceptual-practical training session for your chosen topic. Test it at your fellow students.
Chapter 8. Learning problem solving

The learning aim of this chapter is to be able to design activities through which people can become better explorers, problem solvers and learners of IT.

Information technology is characterised by a quick turnover of new software versions, information systems and hardware gadgets. Users therefore need to upgrade their competence often, meaning they have to learn about the technology and also about information and how IT supports tasks. This implies that IT user competence also includes the competence for learning about IT, which includes learning about information, technology, and tasks.

This chapter will first summarise the competence that improves the conditions for learning in Section 8.1.

In the previous chapters, the learning process for IT competence has been presented as a movement from skills to understanding. Proficient IT users seem to engage also in other learning processes, like playful exploration of the technology, experimenting with its functionality and finding solutions to problems by searching the internet or calling friends. Most users would need to be introduced to these learning processes in order to engage in them on their own at a later stage. This chapter will therefore present these processes such that they can be introduced by the IT teacher.

The learning processes described are applicable to all three subject matter areas of information, IT and task support, but may require quite different approaches to training. Learning task support will to a large extent take place on the job (Boudreau and Robey, 2005), while understanding information and IT might require the study of functional and conceptual models (Kaasbøll et al., 2010).

8.1. Competence for learning

Learning is based on what we already know (Bransford, 2000). When learning a new operation, we adapt our existing competence to the new situation.

When learning a new piece of software, people bring their experience of doing the tasks without computers or with other software. Knowing that text can be duplicated through a copier, they might type their text in the text processor, print it, multiply it in a copier machine and send one copy to each receiver. After having also learnt about electronic mail, they might reconsider this as a way of distributing text.

Novice users learning to use a text processor may have been used to creating one long sequence of text from page to page, and they have read newspapers where the text is divided into columns. If they get the task of organising text into two columns like in the newspaper, they may choose inserting a table with one row and two columns, since this gives the same appearance, even though they manually have to move text from one column to the next one.
Users come with blurred concepts which do not match the principles of the software package, or they have developed more precise understanding from using one piece of software and have trouble when their expectations based on this concept are not met when using another program. For example, users often confuse

- aligning text with the tab key in a text processor
- columns in a text processor
- tables in a text processor
- tables in a spread sheet

A poor learner will not be aware of the possible mismatch between preconceptions and novel technology. On the other side, a person with learning competence in this respect will be open for all options of whether aspects of a concept fit in a new environment or not.

Pedagogical literature also points out other conditions for learning, and understanding usefulness is one of these (Bransford, 2000), as also mentioned in the previous chapter.

Literature on learning also emphasize that understanding and facts ease transfer of skills to new situations (Bransford, 2000). A computer user who has understood the concept of text flow, and that text flows from one column to another, but not between cells in a table, would be more likely to choose the right kind of tool in the next piece of software. Understanding the three subject matter areas presented in the previous chapters therefore ease learning of new IT concepts, information and use of IT in tasks.

Managing a new text processor would also be eased if a user has learnt trivial facts like that Times New Roman and Arial are fonts, because then it is easier to recognise that where these words appear, the functionality concerns fonts. Facts like vendors, file formats, and knowledge of people to ask for help would constitute parts of the learning competence for computer users.

8.2. Navigation: the fourth basic way of learning IT use

The previous chapters have described learning as processes influencing our skills or understanding. We will now regard the improvement of skills as one basic way of learning IT use with the two variations of repetition and imitation. Two other basic ways of learning IT use is interpretation and reflection. Interpretation takes the user from skills to understanding, and reflection improves the mental models of understanding, as illustrated in Figure 44.
The fourth basic learning process to be introduced brings the user from understanding back to skills. *Navigation* means finding out how certain functionality can be triggered at the user interface. Knowing that the IT has certain functionality but not knowing how to trigger it, users try and err, or find *directions* amongst documentation or from other users. A direction departs from functionality and points to how to trigger it in the user interface. Such help text might continue as instructions.

Directions provide the link between task or functionality and how to trigger it. They can be as simple as Figure 45, which is the start of an instruction of how to customise styles.

Explanations and directions deal with the link between conceptual models and the user interface. This commonality is also manifested in a joint problem of terminology.

When designers invent a term for a functionality, only one or two in ten users would use the same term (Furnas et al., 1987). This causes both problems when looking for the interface object which will trigger the wanted functionality, and likewise, messages popping up on the screen may be written with unintelligible terms for most users. Also, chances of success when searching documentation decreases. This terminology trouble has been confirmed by research. Many of the participants in a study of problematic use episodes were not able to find the functionality which they knew existed (Novick and Ward, 2006). When turning to the documentation, they reported additional trouble, since they also could not find the right place...
in the documentation. Often, their search terms did not match the keywords in the documentation.

Guidelines for error messages emphasize user-centred phrasing (Shneiderman and Plaisant, 2010). Considering the users’ variety of terminology, this is easier said than done, which Figure 14 also illustrates.

In order to overcome the terminology problem, explanations and directions generated by a community of many users may secure that a user’s search term hits one of the other users’ help. This is the web solution, which was the most frequently used type of help amongst 107 users (Martin et al., 2005). Half the users had problems with finding what they were looking for, and also half of them had trouble interpreting the documentation found.

For business internal systems, the www may have little to offer. Instead, user questions and responses can be made available for searching also inside the interface of the system, so that the threshold for use is as low as possible.

8.3. Exploration

Kids have fun with exploring new devices, and they play together and discover. Nerds do the same with IT, developing skills at high level, and they are learning oriented in the IT domain. The learning oriented users actively explore the technology, trying to find better ways of using a program for a certain task, and play around with it in order to see what it can do. This is a strong learning competence. The active explorers have a tendency to become local champion, whom others ask for help and who push for new computer applications.

An example from a study of user competence illustrates the difference between users who are good and bad learners. Youssou tells about his learning oriented brother:

*My brother is truly amazing. For myself, if something doesn’t work I might try it again once but the majority of the time I will just ‘give up’. My brother sees these ‘failures’ as challenges to be met and conquered. He delights in the fact that he never has to stop learning because there will always be a new challenge to conquer. He loves the fact the information technology is such a dynamic field that it is always changing, improving and making new breakthroughs.* (Phelps et al., 2001)

The performance oriented users stick to one way of using a program when they have learnt that way, even though there might be easier ways. They refrain from pushing a button which they have not touched before, due to being anxious for making a mistake or loosing data. The anxiety can be regarded as a negative computer learning competence.

Phelps et al. (2001) provide an example of this type of learner too, we call her Ofra:

*If something goes wrong when I am using the computer I freak out and panic, but when I see these people use the computer they seem to be able to work it out on their own. It is obvious to me that I learn differently to them when it comes to information technology.*
A person may be performance oriented in one aspect of life, while learning oriented in another. The stereotypical image of a computer nerd is that he has learnt everything about the computer, but socially, he sticks to what he knows, which is chatting with other nerds. Likewise, the elderly social worker is fabulous in dealing with people, but she has computer paranoia.

The willingness to explore was found to be the most influential characteristic in a study where people were asked to characterise highly competent information systems users (Eschenbrenner, 2010). These users

...try to use IS to its fullest potential ... are not afraid to explore new things.

Learning oriented users seem to be exploring the technology and solving problems. Competence for problem solving and exploration of information, IT and its task support is therefore regarded as a high level of user competence and as a characteristic of good learners of IT.

Figure 46 illustrates the range of learning processes for learning oriented and performance oriented users.

Figure 46. Exploration - a composite learning process. While learning oriented users explore, performance oriented users repeat what they are familiar with.

In a field trial, computers were set up so that children in poor communities in India could play with them without any instruction (Mitra et al., 2005). Observations showed that children in groups explored the system. Seen from the individual child, the other children were external sources for help of all kinds. Regarding each group as an entity, they were left on their own without any external sources of help. In competence tests, the children were asked about the meaning of icons, and a steady progress was demonstrated over nine months. The children

© Jens Kaasbøll, 13th April, 2012
had developed IT skills and also exploration skills. Due to the type of tests carried out, we do not know their understanding of the technology.

Even if learning oriented users have the ability to explore, they might choose not to. In a field study of user learning of software, exploring for the sole purpose of learning constituted the exception (Rieman, 1996). This was the case even if some of these users were computer scientists. Reasons given were that exploring was unproductive, and that they had too much else to do.

8.4.  Problem solving

When problems of use appear, we are thrown into an urge for action, either through learning our way through it or getting others to solve it. In the latter case, learning has meagre conditions.

We will distinguish between two kinds of problem solving. Experimentation takes place when we start wondering whether IT can do something that we would like it to do, while trouble shooting appears when the technology does not respond as expected. The starting points differ, but in both cases, we may learn how to solve the problem through composite learning processes like exploration.

Experimentation

Exploration is learning while playing around with IT without other purposes than learning and possibly having fun.

Experimentation is a planned action of problem solving, starting at the understanding level with a hypothesis of the type “It can do this, but I want it to do that. Can it? How?” Then the test is carried out, and the result is compared to the initial question. The experiment will include a composite learning process through the full circle back to the understanding level.

Advanced users experiment a lot. They test the limits of software, for example “This field is for numbers. Will it take text also?” or “Can the scanner transfer data directly to the phone, or do I have to use the computer as a receiver?”

Experimentation may fail, in the sense that we do not solve the problem. We might nevertheless have learnt something about the unsuccessful actions.

Inadequate mental models was one of the top three causes of failure in experimentation without external sources as well as when referring to them in a study of users (Novick et al., 2009).

Trouble shooting

Experimentation was triggered from understanding. Trouble shooting is triggered from practice, when noticing that the IT does not do as expected. We might get help from a colleague to interpret or search the web for an explanation, and in both cases we bring the issue to the understanding level. Through reflection, we might find a possible solution, which
has to be tested. Trouble shooting may include the same composed learning process as experimentation, although the purpose differs.

In a lab study of troubleshooting, users’ strategies for finding errors in spread sheets were evaluated (Subrahmaniyan et al., 2008), and the following main strategies were identified:

- Dataflow. Follow formula dependencies. This included interpretation of the formulas and reflection on their correctness. Arrows between formulas and their cell arguments were displayed. Male participants succeeded with this strategy, females did not.

- Testing. Trying different values to evaluate the results, including the whole cycle of experimentation. Again, males were more successful in testing.

- Code inspection. Users examined formulas and compared the through interpretation and reflection. Some used lists of formulas to check off those inspected. The reflective code inspection was the preferred strategy for successful females. Others changed the formulas without reflection and interpreted the result, often introducing more bugs in the formulas. Unreflective code changes with testing were an unsuccessful strategy, also preferred by females.

- Specification checking. Comparing formulas with the specified result. This was the only strategy which included external sources.

In summary, the study points to the necessity to reflect when troubleshooting errors in spreadsheet formulas. Playful exploration was not productive (Subrahmaniyan et al., 2008).

8.5. Fostering metacognitive competence

Research has demonstrated that training is more efficient than self exploration (Simon and Werner, 1996). However, training is normally not available when wanted, so users need to learn without a teacher in most cases. A field study of user learning emphasized experimentation as the main strategy for learning (Rieman, 1996), since users prefer learning new features of software when needed. External sources were used in the majority of cases.

Knowing that kids explore their environment and their toys by playing, and that animals also learn through playing, one could assume that there is no need for teaching people the process of exploring. Sadly, this assumption is wrong, as the case of Ofra illustrates. Users therefore need to learn the competencies of exploration, experimentation and trouble shooting. Since these competencies are about learning, we can call them meta-learning, and the terms problem solving competence and metacognitive competence are often used. Since this competence is needed for efficient learning, it is included as the fourth area of IT user competence, see Figure 47. The learning processes themselves are included as the fourth area of competence. Teachers should try to improve the learners’ problem solving competence.
Figure 47. The learning processes themselves are included as the fourth area of competence.

In order to explore, the users need to feel confident; not being afraid of the technology (Eschenbrenner, 2010), and this concerns performance oriented learners like Ofra in particular. A trainer could try convincing them that when things do not work, it is not because they have destroyed the computer. Also, reminding them that there is normally an undo operation which can bring them back to where they were could calm their nerves.

Trainers and support personnel can improve learners’ metacognitive competence in several ways: by telling about the benefits, by demonstrating experimentation, making users do it on their own and in small groups under supervision, and by discussing it afterwards. A video demonstrating the steps might also be of assistance to some. A study of effectiveness of problem solving compared three strategies (Andrade et al., 2009):

- Experimentation without referring to external sources
- Only consulting external sources
- Switching between experimentation and consulting external sources

The latter was the most efficient, implying that trainers should teach this practice.

A setting that encourages exploration needs to be created. In a course, the trainer could try triggering exploration by prompts like

- Find out what this program does. The first reasonable response is awarded.
• First row: left button. Second row: right button. After five minutes, you explain to the other row what the buttons do.

• Now, we will try to make this program do something that it was not designed for. Use the spreadsheet application to draw a face.

Some IT lends itself easier to exploration, for instance the Lego robots, while an enterprise resource planning system seems to have control as its main purpose. An option is exploring its applicability to other tasks than its purpose, like the spreadsheet for drawing a face. The teacher could e.g. say

• Find other tasks where you could apply this program.

A presentation program could be the best alternative for novices when starting learning IT. It allows for exploration after a quick introduction, users can write and make illustrations, it has built in design elements to be composed, and users can include their own pictures. Kids also find excitement in presenting what they have made, and there is no wrong answer. It would possibly work equally well for adults.

Practicing exploration constitute one step for users to adopt it as a learning activity. Another step would be to discuss exploration as a way of learning, so that users also reflect upon learning strategies for IT. The teacher could trigger discussions on what users learnt during an exploration, and point to specific activities which seemed particularly productive in the class.

A trainer might try triggering exploration by saying “Find out what this program is doing!” The learners may well end up in experimentation nevertheless, discussing hypotheses about what the program can do, testing them, and comparing the results with the hypothesis. Such a systematic approach to learning has the same components as the standard hypothetico-deductive research method. Trainers could encourage it by questions like

• Recall that in the presentation program, we could control the layout of slides with the master slide. How does the text processor compare on this functionality?

• This is a task which you might encounter at work. Find out whether the IT can support this!

The latter request brings the course activities closer to challenges to be encountered after training. It is also an example of problem based learning where the teacher has described a reasonably realistic situation which the learners recognise. When the challenge is complex, the learners will have to go through many cycles of experimentation. For efficient learning, the trainer needs to provide lots of support during such problem based exercises (Hmelo-Silver et al., 2007).

---

3 Edgar Bostrøm, personal communication.
Observing precisely what happens, comparing with beliefs and testing unclear results are parts of such competence. Also, having the ability to explore cooperatively or draw on external resources is crucial in all four learning processes in the cycle.

Finding the solutions to problems is no guarantee for learning at all. The problem may appear in a slightly different way the next time, so that the interpretation of it into a known category does not work. The user may not remember where to find the information needed and how to search for it. Also, even if the user understands the issue, the way of implementing it in the system might have been forgotten.

During interpretation, the opportunities for misunderstanding are ample.

_This worked last time I did it, why did the computer do something else now?_ says Ksenija. Simultaneously, the trainer has noticed that this time she hit another button than the last time, while she is convinced that she repeated exactly the same typing. The trainer’s response could be to ask Ksenija starting over again and re-typing. If the computer performs as she expects this time, the trainer could bring up the issue of observing precisely what one is doing.

Another failure in the interpretation stage is avoiding observing the computer response precisely. The support personnel will often ask the user for the contents of error messages, and if these are lost, reproducing the situation is difficult. Therefore, taking complete and precise notes of what is happening when errors occur should be taught during IT training. This includes saving screen-shots and intermediate results.

Also, the teacher can bring up Frequently Asked Questions lists from the technology taught and the learners can try searching for it on the web. A way of getting help when an error message appears is to type the message in a search engine and try reading a hit. The “404 error” gives 8 million hits and the first one is from Wikipedia, see Figure 14.

A laboratory study where learners were given tasks to be done but no instructions of how to do them, compared individuals and pairs. The results showed that the pairs developed a better understanding of how the software operated and they performed better in exercises which introduced some novel elements (Lim et al., 1997).

Advanced users who help out others learn from troubleshooting their friends’ computers (Poole et al., 2009):

_I just fixed things and learned at the same time....Actually, I remember feeling excited when I first helped someone out._

### 8.6. Summary

Users who are better able to learn on their own become more proficient and require less attention for support services. Improved understanding increases the chances of learning on your own.
We call the ability to explore, experiment and trouble shoot metacognitive or problem solving competence. Explicit training in Metacognition would improve users’ learning abilities.

6. **Train users so that they can solve problems and learn on their own.**

**Exercises**

1. In which ways can directions be provided at the user interface? Provide examples from software or gadgets which you have used.

2. A user is struggling with changing a one column document into two columns. What would you tell him such that he both solves the problem and becomes a better problem solver?

3. Read the conversations below between a user and a support person helping out. Which types of learning processes (basic and combined) did the user and the supporter go through? Did the supporter provide any explicit training for problem solving? How could the supporter have provided more training for problem solving?

   **a. A phone conversation – mending the purchase order**

   Kyung: Hi. I am editing a purchase order, and then I need to enter a higher amount than I typed yesterday. But I can’t get the system to change the number, it seems locked. How can I fix it?

   Lena: You see, the auditor regulations state that once you have logged out, you can’t change the data that you have entered. You can’t even delete the item that’s there. So what you can do is insert a new line with the additional amount, such that the two add up to what you want.

   Kyung: Aha. Very strange and awkward, but a smart solution. Does it work if the original amount is too high also?

   Lena: Yes, then you type what you need to subtract with a minus sign in front.

   Kyung: Makes sense. Thanks.

   **b. Phone conversation – new version**

   Wanda: I cannot open the document from headquarters any longer.

   Emeline: Does it concern all documents, or just the last one you tried?

   Wanda: I don’t know.

   Emeline: Try open one you have opened before.

   Wanda: OK … now I try the one I received … It opens.

   Emeline: OK, then I think you need to upgrade to version 3 of Editor. You see, headquarters have started saving their files only in version 3 formats now.
c. **Phone conversation – Table of contents**

Samba: Hi, I wonder whether Editor 3 can create an index.

Lola: Sure, First you mark those words you want to include. Then select Special Sections and Index.

Samba: I’ve seen that, but I want an index of the headlines of the document.

Lola: Oh, but then you probably want a Table of Contents and not an Index.

Samba: Oh, so that is what it’s called.

Lola: There are tutorials on how to operate the Editor 3 on our web site. Go to Software and then Tutorials and you will find a long list.

Samba: Software … Tutorials … Oh yes, I can see that. Editor 3 … and Table of Contents down there at T, yes. Thanks. … And here it opens. … It looks simple, I’ll try.

d. **Co-located – Web server**

Ivana: I don’t see the mail page like I used to.

Edwin: Which page?

Ivana: The yahoo mail.

Edwin: OK. We have had some trouble with the DNS server, so let me reset it. (Takes over the keyboard and fixes the problem.) Now it will work fine. Hopefully we will get the server up and running again this afternoon. (Leaves.)

e. **Co-located – Synchronising phone and PC**

Rubén: I wonder how I can synchronise the contacts in my phone with those in the computer.
Toots: Well, I don’t know. Actually, in many cases, also we, who are IT people, have to search for solutions on the web. So let’s try searching together, and then we might find a way. Which brand is your phone?

Rubén: It’s a SonSang Nokola model

Toots: OK. Let’s use that and “synchronise contacts” as search terms.

…

The first hit is from SonSang. I would rather look in the second one, which seems to be a similar request posted on an e-mail list. The responses in such lists tend to be written by users so that you don’t have to be an engineer to understand.

4. Assume that you are going to train a class about using a bibliography software for writing citations and references, including information, IT and task support competence. Provide an assignment aiming at learning problem solving.

Problem solving is more efficient when switching between trying on your own and consulting external resources. Find external resources on the web which may be helpful for each of these three areas. How can you trigger learners to find such resources?
Chapter 9. Evaluation of training

The learning aim of this chapter is to be able to design appropriate evaluations of training.

Organisations evaluate their activities to find out whether these are worth the cost or whether they can be done in a better way the next time. In general, there are four ways of evaluating training (Kirkpatrick, 1959, Kirkpatrick, 1975, Kirkpatrick and Kirkpatrick, 2006)

1. **Reaction.** Reaction is the participants’ opinion of the course. It is normally gauged during the training session. For example, the reaction can be found through a questionnaire to the participants asking their opinion of the course and the teacher.

2. **Learning.** This is an assessment of what the participants learnt during the course. An exam assesses learning, but in order to evaluate training, the exam should be administered also before the training, so that the difference in competence before and after can be found.

3. **Behaviour.** An investigation of people’s use of the IT when back at work or in other activities. For example, ask the participants about which functionality in a software system they use two months after the course.

4. **Result.** This is a measurement of changes in organisational performance, for example the number of clients which can be taken care of by means of new IT learnt in the course.

Number 3 and 4 evaluate the transfer of competence from course to work. The four ways are ordered in time as shown in Figure 48. The timing of four ways of evaluating training.

A combination of evaluations is normally better than a single one. For example, an evaluation of behavioral change might find that some of the intended effects of the IT had not come...
about. Then an evaluation of reaction to training might point to particular weaknesses in the teaching. It might as well happen that evaluation of learning concludes that the staff has learnt what was intended, but the evaluation of result did not demonstrate any change. This would mean that the training is fine, while problems have arisen in the implementation of changes in the organisation.

The levels do not imply any causal relationships between reaction, learning, behavioral change and result (Holton III, 1996). Learner motivation will influence the learning process and enabling factors in the organisation have strong impacts on change and results.

9.1. Evaluation of reaction to training

Learners’ reactions can be observed in the class as spontaneous statements concerning anything taking place there. The trainer might hear

Now I got it right.
I don’t see the point of this topic.
Good I got to know you, so that I can ask you later.
The lunch was delicious.

Statement 1 concerns the learning process, while no 2 is about the motivation for some of the course contents. The third one hints at an important organisational issue, namely that the learner has met somebody whom can be approached for help during when transfer to the work is supposed to take place. The fourth statement concerns the course environment. Knowing the learners’ reactions to any of these topics may be relevant when revising courses.

A teacher observing learners constitutes no systematic approach to evaluation, however. If observation is wanted, an independent person will be able to observe both the learners and the trainers systematically. If no additional person is available, interviews or questionnaires are alternative ways of gauging the learners’ opinions. Questionnaires have the advantage of anonymous responses.

Questions in interviews or questionnaires could address any of the topics in the examples above. Since learning and transfer might depend on several factors, and since there can be large individual differences, getting acquainted with other users might be as important as motivation and course environment.

When evaluating the learners’ reaction to the course contents, all three subject matter areas of IT competence can be addressed, since there may be a need for adjusting one or more of them. Concerning task support, questions like

Did the course address your needs in your job? If some needs were not met, which ones?
could be used. The closed question addressing the needs overall could be responded to on a scale from 1 to 6. The latter, open question can provide knowledge on specific task support to include.

Participants’ reaction to the IT and information contents of the training could be addressed by asking

Did the training provide a sufficient background for understanding the IT system? And for using it?

Did the course explain the data in the system sufficiently?
The latter question should be rephrased, targeted to the information taught, for example

Did the course explain the account types thoroughly enough?

if an accounting system was to be taught. Again, scales could be used in the response, and the closed questions could be followed up by open ended ones as in the task support case.

In a case study of evaluation of user training, the users responded that the hands-on exercises with real world data were useful for keeping their motivation (Mahapatra and Lai, 2005). This response was useful for the trainers, who became even more focused on crafting training material to fit the background and expectations of the learner group.

9.2. Evaluation of learning – assessing competence

Assessing IT competence is an activity which can take place for many reasons in various settings.

• An organisation wants their staff to be at a certain level, so they organise a test for everybody to take.

• When planning a course for a group of participants, we want to know their competence, so that the course can start at the appropriate level.

• When a course plan has been settled, we screen the possible participants, so that those at a too low or high level are channeled into other training.

• Employers test the IT competence of job applicants.

• Applicants document their competence through completing a certified test.

• A school administers an exam in their IT class.

• For evaluating a training course, we assess IT use competence prior to the course and afterwards. The difference between the two levels will tell us what the course has contributed with, like Kirkpatrick’s model state (Kirkpatrick, 1975).

Levels of IT competence have been described as skills or understanding in Chapter 4-8, and tests of competence can be arranged accordingly. Basically, skills are assessed through
practical tests, and understanding with written or oral questions and answers. A test can assess IT, information or task support competence or combinations thereof. Telling the accountant Rigo, who sits in front of a computer to

Print the spreadsheet.

would test his IT skills. Assuming that a reimbursement claim is registered in a database,

Check the reimbursement claim.
is a task which can be given him to test his IT, information and task support skills. In order to test his understanding, the following types of questions could do:

- What is a spreadsheet program? (IT understanding)
- What is debit and credit? (Information understanding)
- What is the purpose of double-entry bookkeeping? (Task support understanding)

People can express competence which is only at the level of skills also, like Kirsten talking about her sequence of tasks in section 6.1. Asking questions like

Which menu choices and buttons do you use for creating a numbered list which starts at the number 3?

does not require a response at the level of understanding; see Edith’s response in Section 4.2.

Assessing task support competence

Questions concerning task competence could be open ended. Focusing on a software tool:

Note down a task in your job where you use or could use spreadsheets. What is the advantage of using a spreadsheet in this task?

Taking the task as the point of departure, we could design a test for the level of understanding possible changes:

You are organising a sports event. For which tasks can IT be helpful, and which IT hardware and software would you use?

Skills test:

Create a spread sheet for currency conversion.

Multiple choice questions could also be IT focused:

Which of the following tasks can you use a spreadsheet for when planning a new house?

a. Draw the floor plan.
b. Compare the cost of different floor covers.
c. Find the formulas for areas of rooms and walls.
d. Write the contract for the architect.

Or based on the task:

You are planning a new house. Which of the following statements are correct?
a. The contract can be written with Adobe Reader.
b. Tables in text processors can be used for comparing colours.
c. I can communicate with the architect through sharing a folder on Dropbox.
d. The floor plan can be drawn with a spreadsheet.

Questions concerning task support and information should target the learners’ experience, so that they are familiar with the background of the question. When testing for an organisation, addressing work tasks in the questions would be appropriate.

Assessing problem solving competence

Competence at the level of understanding is valuable for further learning. In addition, problem solving (metacognitive) competence is useful. Open ended questions for checking experimentation competence could be:

Find out what goes right and wrong when copying from a pdf document and pasting into a text processor.

Here is a new application. Find out what it does. Multiple choice questions could assess the part of experimentation that involves reflection. A test could be:

You have attached a file to an e-mail to Bob. Then you discover that the file contains some statements which you do not want Bob to read, so you make some changes in the file and save it. You wonder whether Bob will get the changed file. Which of the alternatives below will give you the answer you need? Beware that the alternative should also tell us how to do it the next time.

a. Ask Bob about the name of the file he received.
b. Check which file that disappeared from your disk.
c. Remove Bob from the list of receivers and enter yourself instead.
d. After sending, check the mailbox containing sent messages.
e. Send the file to yourself from another e-mail account.
f. Remove the attachment and then re-attach the file before sending.

The flip side of such questions is that they do not necessarily test experimentation competence. If Manu, who answers, is very familiar with the e-mailing, he might answer correctly because he knows a lot about his e-mail service, and not because he is clever at setting up experiments.

IT users can also learn through troubleshooting. A general way of checking users’ ideas about troubleshooting is asking about repetitions:

You observe that the computer responded in a way that you did not intend. You repeat it, and this time it worked out. What can the reason be?

a. The computer regained momentum.
b. There was a virus the first time.
c. You made a typing mistake the first time.
d. The quantum mechanical circuit at the mother board kicked in.

e. The hard disk crashed.

Asking questions like the ones presented for experimentation is also possible, having the same drawbacks, in the sense that we cannot always know whether we are testing the users’ troubleshooting competence or the mastery of the particular example.

In addition to questionnaires, problem solving competence also lends itself to observation. The teacher can observe how the learner handles the software and looks up resources for help. For a pair of learners, also their conversations can be observed. Observations of problem solving come out with significantly different results than questionnaires to the same users (Novick et al., 2007). Assuming that the difference is due to poor memorisation of details of actions, observations seem to give a more correct account of problem solving.

Teachers normally observe learners’ activities in order to adjust the training on the spot. Such evaluation is called \textit{formative}, and it often helps improving the quality of teaching there and then. This chapter concerns the \textit{summative} evaluation, which aims at improving the course the next time it is carried out. However, the observations which teachers carry out informally on the learners’ achievements may also be included in the assessment of learning.

\section*{9.3. Evaluation of behavioral change}

Evaluation of reaction to training and of learning can be carried out in any training course, independently of what the learners are going to do afterwards. Evaluation of behavioral change is intended for company specific courses, where it is possible to approach the course participants some weeks or a few months after the course has ended. If the participants use IT to support their tasks in a different way from what they did before, a behavioral change has taken place. Finding the type and extent of such change is what evaluation of behavioral change aim at.

When the IT to be learnt is a software package on a server, users’ operations can be logged. Data on changes in user behaviour can therefore be found by analysing such logs. Details on individuals’ use of specific functionality can be summarised with much less effort than asking users. Like any surveillance, employees should be informed that their behaviour on a computer system is logged. Some countries or trades may have regulations or agreements concerning surveillance systems.

Logs do normally not tell us all we want to know, however. They provide statistics, but not the reasons why the staff use specific functionality or avoid it. Again, observations, interviews and questionnaires constitute possible ways of investigation. Observation takes more time and provides more detail per individual, while questionnaires reaches a larger number of users with less detailed response. Individual and group interviews lay in between the two.

When designing questions aiming at finding reasons, the factors from the Technology Acceptance Model (Section 6.2, p.74) can be a starting point. Usefulness, ease of use, social pressure, facilitating conditions and combinations of these are likely reasons for the use or non use of a specific functionality. For example, assume that the logs have demonstrated that
the functionality for search for similar cases in the client system is used to a very little extent. A relevant question in a questionnaire to the users could be:

Regarding the “Search for similar cases” in the client system, rank your agreement with the following statements on a scale from 1 to 6 (1=Disagree completely, 6=Agree completely):

☐ I find this search very useful in my work.

☐ This search is easy to use.

☐ The majority of my peers use this search.

☐ The computer system responds quickly on this search.

A low score on usefulness may have two reasons. Either, this has not been taught properly during training, so that the users have not understood its usefulness, or the functionality is of minor value. If users say that it is difficult to use, the training might be to blame. If they say that it is easy, the reason for low use may be that it is actually of little value, Peer pressure and technical conditions should also be taken into account.

9.4. Evaluation of result or outcome

The fourth level of Kirkpatrick’s model (Kirkpatrick, 1975) is evaluating whether the introduction of the IT tool fulfill its goals, and how the training has contributed to the result. While it may exist simple ways of measuring organisational performance, like the bottom line in a company, drawing inferences from changes in results back to training courses are often difficult due to a high number of intervening variables. For example, the business may observe that the number of clients served after introduction of the new client information system has risen by 20%. The reason for the change may be that the market has increased, that clients are served better, that a new system has been introduced through successful training, or that the staff has managed to get the new system working despite poor training.

A more feasible way is selecting some outcome which is closer to what was trained. Instead of the number of clients, we could for example measure the time for serving each client. This relieves us from dealing with markets or other external factors as possible explanations. Still, the possible impact of training on efficiency of client handling has to be established.

In an evaluation of training of an information system for reporting health statistics, it was possible to count the number of data items filled at any point in time and also perform some automatic comparisons to judge accuracy (Ngoma et al., 2008). In order to gauge the training result, the completeness and accuracy were first measured in several sites, some of which would be trained. After training was completed, the completeness and accuracy were measured again in the same sites, and the results compared, see Figure 49. The sites which...
were not trained constituted a control group. If the sites without training had improved as much as those with training, the training would not have had any effect. The untrained improved by 10% while the trained with 50%, leading to the conclusion that training was effective.

**No training** | **Trained**
--- | ---
Incompleteness | Inaccuracy |
Incompleteness | Inaccuracy |
0 % | 5 % |
10 % | 15 % |
15 % | 20 % |
20 % | 25 % |

**Figure 49.** Measurement of results of training. The “No training” sites constituted a control group.

**Exercises**

1. A retail shop is introducing bar code scanners for the cashiers. They all receive a one hour introductory course where the purpose of the system and the information in the bar codes are taught. How would you evaluate the training?

2. A few years thereafter, the shop converts to self checkout. The customers scan the bar codes and pay by credit card. Some cashiers are retained for helping out and controlling. Which training would you provide for the cashiers? How would you evaluate it?

3. You are the trainer in a company whose business is conducting user training for general software packages at several levels of competence. Your customers are mainly organisations sending batches of 3 – 20 of their employees to courses. How would you evaluate your training? Try finding an appropriate evaluation method at each of the four levels

**Project**

Suggest ways of evaluating your training at each of the four levels. Out of these four, which one will give most insight for improving the training?
Chapter 10. IT user competence standards

The subject matter areas of IT user competence were discussed in Chapter 3, and problem solving competence was added as a fourth area in Chapter 8. This book has also stressed the need for understanding in these four areas in addition to the skills. Understanding plus problem solving competence enable users to continue their learning when new technology is introduced and expended.

The range of technology, information and task support which users have needed has evolved over the years. In the 1980’s, the file system, individual office applications and possible business information systems constituted the typical collection of IT for users to master. The 90’s brought local networks and the Internet, with servers, browsers and e-mail added to the standard repertoire. During the last ten years, Web 2.0, mobile phones, digital cameras, music players and a number of other personal gadgets have lead to a diversification of modes of interaction as well as hardware. Business systems have moved into the browsers or migrated to enterprise resource planning software.

The continuous expansion of IT applications disables any stable description of the range of IT, information and task support competence. However, some comprehensive guidelines for IT user competence have been developed, either for the general public, for special occupations or pupils at school.

10.1. Standards and guidelines

An approach to the latter is the FITness (Fluency with IT) report, which describes a comprehensive set of skills, concepts and capabilities, see Figure 50 (Committee on Information Technology Literacy, 1999). This set of competencies includes the four subject matter areas, although task support is less specified. Contrary to many textbooks on software use, it addresses concepts and principles. FITness go even a step further, by including programming and algorithms, which is considered beyond IT user competence as advocated in this book.

Most organisations depend on their employees being capable of operating business critical systems. For example, the cashier needs to be able to check out goods and register payment, police officers need to know how to use the communication equipment, and the air traffic controller must be fluent in the IT system mapping the flights. In the latter case, and in other high risk tasks like handling surgical equipment and nuclear power plant control, the operators might have to be certified. A detailed specification of the competence, including information technology, task support and information competence, will be required for constructing certification tests.
Standards are operationalised through curricula and tests. Competency tests are used in level 2 evaluation of training, see Section 9.2. General tests of competencies are presented below.

### 10.2. Tests

Both commercial and other organisations have developed IT user competency tests, see (Covello, 2010) for an overview. Three major ones are:

- **Educational Testing Service** is a US based, non-profit organisation, known for its Test of English as a Foreign Language (TOEFL). They offer the iSkills Assessment, which measures IT literacy (Educational Testing Service, 2011)

- **Certiport** is a commercial actor, also providing courses and tests for software professionals. (Certiport Inc., 2011)

---

© Jens Kaasbøll, 13th April, 2012
- ECDL Foundation is a non-profit organisation providing the European Computer Driving License, also known as International Computer Driving License (ICDL). It was founded in 1995 by the Council of European Professional Informatics Societies in order to improve digital literacy across Europe. Later, it has gone intercontinental, and 11 million people have conducted tests given in 41 languages. (ECDL Foundation, 2011)

We will look at some sample questions to see how the tests are constructed. The ECDL is divided into 13 modules, mainly according to software types. In addition, there are three general modules:

- Concepts of ICT
- IT Security
- Project Planning

About Module 1, the ECDL / ICDL Sample Part-Tests (ECDL / ICDL, 2009 Module 1, p 1-2) says:

Module 1 *Concepts of Information and Communication Technology (ICT)* requires the candidate to understand the main concepts of ICT at a general level, and to know about the different parts of a computer.

The candidate shall be able to:
- Understand what hardware is, know about factors that affect computer performance and know about peripheral devices.
- Understand what software is and give examples of common applications software and operating system software.
- Understand how information networks are used within computing, and be aware of the different options to connect to the Internet.
- Understand what Information and Communication Technology (ICT) is and give examples of its practical applications in everyday life.
- Understand health and safety and environmental issues in relation to using computers.
- Recognize important security issues associated with using computers.
- Recognize important legal issues in relation to copyright and data protection associated with using computers.

Assume that we constructed open ended test questions for these learning goals, like:

What is the Internet?

Karl responds:

*A network through which we access all places in the world*

Karl is describing a function of the Internet, so he is at the IT functional understanding level. His understanding may be limited, since he does not specify the different types of functionalities, like the www, email, chat, etc. The ECDL has multiple choice questions for testing understanding (ECDL / ICDL, 2009 Samle Part-Test 1.2, p 3):
Which one of the following statements about the Internet is TRUE?

| a. The Internet is a global network that links many computer networks together. |
| b. The Internet is a private company network. |
| c. The Internet is a visual representation of linked documents. |
| d. The Internet is a network operating system. |

The statements a-d describes the Internet at the IT conceptual level. Given that Karl responded like above, he would most likely tick the a alternative, so his test result would show that he understands the Internet at the IT conceptual level. Constructing a multiple choice test for distinguishing between levels of understanding is difficult. If the following option was included:

e. The Internet is a network allowing for retrieval of data from remote computers.

both a and e would have been correct responses, and Karl could have selected any of these.

Concerning softwares, the spreadsheet module is selected as an example (ECDL / ICDL, 2009 Module 4, p 1):

Module 4 Spreadsheets requires the candidate to understand the concept of spreadsheets and to demonstrate an ability to use a spreadsheet to produce accurate work outputs. The candidate shall be able to:
- Work with spreadsheets and save them in different file formats.
- Choose built-in options such as the Help function within the application to enhance productivity.
- Enter data into cells and use good practice in creating lists. Select, sort and copy, move and delete data.
- Edit rows and columns in a worksheet. Copy, move, delete and appropriately rename worksheets.
- Create mathematical and logical formulas using standard spreadsheet functions. Use good practice in formula creation and recognize error values in formulas.
- Format numbers and text content in a spreadsheet.
- Choose, create and format charts to communicate information meaningfully.
- Adjust spreadsheet page settings and check and correct spreadsheet content before finally printing spreadsheets.

The learning goal specifies a series of skills, which are described in some detail. The “concept of spreadsheet” is not explained, so the understanding part of the goal is unclear. The tests are mainly of the practical kind, foe example (ECDL / ICDL, 2009):

Enter a formula in cell F5 with an absolute cell reference for one cell only that divides the content of cell E5 by the content of cell E11. Copy the formula in cell F5 to the cell range F6:F10.

So the goal of skills seems to correspond to the test type. An open ended question which addresses understanding is also included (ECDL / ICDL, 2009):

Which of the two cells F4 or F5 displays good practice in totaling a cell range? Enter your answer in cell B14.
Answers to open ended questions like this one can be assessed right or wrong or according to a scale, for example skill – functional understanding – conceptual understanding.

Responses to multiple choice tests are easy to assess. Assessing whether the candidate has written a correct formula in a spreadsheet also requires only a quick view. Reading, interpreting and grading an open ended answer is much more tedious.

ECDL’s division of IT competence into softwares hinders questions which relate concepts from two IT tools. For example, the following question could not be included:

What is the similarity between master slides in presentation programs and styles in text processors?
   a. They guide the printer.
   b. They provide information for the table of contents.
   c. They enable coherent formatting of the file.
   d. They enable import of slides into word processors.

Also differences between concepts could have been included if the tests could span more applications, for example:

What is the difference between tables and column layout in a text processor?
   a. Tables are imported from a spreadsheet, while column layout is generated within the text processor.
   b. Tables are only found in spreadsheets.
   c. Column layout is the vertical sequence of cells in a table.
   d. Tables are composed of separate cells of text, while column layout means that the text is displayed in sequential, vertical stripes.

The Instant Digital Competence Assessment (iDCA) is a recent test aimed at 14-18 year olds (Calvani et al., 2012). It is organised in the three dimensions technology, cognitive and ethics, instead of the organisation according to IT applications found in the ECDL. Technology corresponds to the IT subject matter area. The cognitive dimension addresses management and evaluation of data, which belong to the information area. Ethics covers general principles and constrains for IT and information use and is a part of the task support subject matter area. Since iDCA is not compartmentalised into software products, it could cater for the two questions above.

iDCA consists of multiple choice questions and does not address skills by asking the respondents to carry out operations on the computer. Its technological area addresses problem solving.

Competence tests versus self-reporting

Performance on competence tests have been compared with students’ self-reporting of their competence level. The latter was gauged by users responding to statements like:

- My spreadsheet skills are good.
• I am a more experienced spreadsheet user than most of my peers.

• I feel competent to use a range of applications.

• I feel comfortable opening and saving spreadsheet files.

The respondents would agree or disagree on a scale.

Most studies conclude that there is no correlation between how people self-report their level of IT competence and how they perform in tests (Larres et al., 2003, Merritt et al., 2005, Sieber, 2009, van Vliet and Kletke, 1994, Ballantine et al., 2007, Sink et al., 2008, Grant et al., 2008). Low performing users overestimate their capabilities. This result is also in accordance with findings from other areas (Boud and Falchikov, 1989). Although there are studies which have found a correspondence between self-reporting and test results (Hakkarainen et al., 2000), self reporting seems generally unreliable.

A consequence of users overrating their competence is that trainers and educators who rely on self-reporting assume a higher entry level than what is the case. For example, in a study of 173 college students 75% perceived their word processing proficiency as high and 20% as average (Grant et al., 2008). In the skills test, questions were grouped as basic, moderate and advanced. Table 9 shows the ten tasks which the researchers had characterised as moderately difficult. Tasks with correctness rank 1-7 are operations on the main document text flow, so no understanding of the data structure of document files is necessary. Tables and headers (rank 8-10) introduce independent text flows, requiring the students to alter their mental model of a document as a single sequence of characters to a multi sequence model. The majority of students seem to be stuck in the single text flow mental model, even though they characterise themselves as average or highly proficient.

Table 9. Performance of college students in the US on word processing tasks (Grant et al., 2008)

<table>
<thead>
<tr>
<th>Moderate tasks</th>
<th>Correct performance</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count words</td>
<td>91%</td>
<td>1</td>
</tr>
<tr>
<td>Add bullets</td>
<td>88%</td>
<td>2</td>
</tr>
<tr>
<td>Highlight text</td>
<td>82%</td>
<td>3</td>
</tr>
<tr>
<td>Find and replace text</td>
<td>60%</td>
<td>4</td>
</tr>
<tr>
<td>Use the Thesaurus</td>
<td>57%</td>
<td>5</td>
</tr>
<tr>
<td>Insert a date</td>
<td>54%</td>
<td>6</td>
</tr>
<tr>
<td>Justify a paragraph</td>
<td>47%</td>
<td>7</td>
</tr>
<tr>
<td>Enter data in a Word table</td>
<td>33%</td>
<td>8</td>
</tr>
<tr>
<td>Insert rows in a table</td>
<td>27%</td>
<td>9</td>
</tr>
<tr>
<td>Create a document header</td>
<td>8%</td>
<td>10</td>
</tr>
</tbody>
</table>

Although the competency tests do not distinguish clearly between a skill and understanding level, this test indicates that college students have a limited IT understanding. They might base their high self confidence on their skills in getting a document produced.
10.3. IT competence levels

This book has described three levels of individual user competency: skills, understanding and problem solving in the four subject matter areas. Results of measurements of IT user skills worldwide follow their own ways of grading competence.

In an iDCA study in Italy, the teenagers scored higher on trouble shooting than understanding IT concepts (Calvani et al., 2012). This indicates that people learn simple trouble shooting skills before they acquire understanding of many IT concepts.

An international survey of digital reading competence at school level 5 concerned the pupils’ ability to navigate and find appropriate web pages efficiently (OECD, 2011). Also, they were assessed on their skills in evaluating the information retrieved. The study therefore mainly addressed the information subject matter area. Interestingly, South Korean children outperformed the students from the other countries, including New Zealand and Australia, Japan, European and South American countries in this rank. Africa and North America were not represented. While a common opinion may be that people in the newly industrialised countries in Asia are well versed in electronics, while the European children are more literate in the original sense, this OECD study only partly supports such a view. Korean students perform better in digital than in print reading, while the opposite is true in Eastern Europe and South America.

Girls outperform boys in both digital and print reading (OECD, 2011). The same is found in a study of college students in the US (Hignite et al., 2009). An ICT literacy test amongst 6 and 10 year old children in Australia included a range of tasks typical for the age groups. Technological competence was necessary in addition to task support (MCEECDYA, 2010). Also in these areas of competence girls performed better than boys. A test of high school students in China with iDCA showed no performance difference between the sexes (Li and Ranieri, 2010), while boys performed better than girls with the same test in Italy (Calvani et al., 2012).

The findings that girls outperform boys contrast the results from more than 30 previous studies summarised in (Cooper, 2006). One reason for this difference could be that the former IT assessments were more biased towards technology, while information and task support have been given a larger proportion in recent years. Another factor may be that young children now grow up with mobile phones and social media on the internet, and that communication is more aligned to girls’ interests, while boys are competing in computer games. The recent studies showing female superiority were carried out amongst children, while former studies have addressed all age groups.

Socio-economic factors are generally influencing competence levels, and this is also the case for IT related competencies (OECD, 2011). Having a computer at home has a positive effect on children’s IT literacy.
Exercises

1. Complete one module of the ECDL test.
   a. Which of the subject matter areas did it cover?
      i. Information
      ii. Information technology
      iii. Task support
      iv. Problem solving (Metacognition)
   b. Which level of competence did the questions aim at?
      i. Skill
      ii. Understanding
      iii. Problem solving
   c. Write a question for this module which addresses understanding or problem solving.

2. Complete one level in one of the areas in the iDCA test.
   a. Which of the subject matter areas did it cover?
      i. Information
      ii. Information technology
      iii. Task support
      iv. Problem solving (Metacognition)
   b. Which level of competence did the questions aim at?
      i. Skill
      ii. Understanding
      iii. Problem solving
   c. Write a question for this module which addresses understanding or problem solving.

3. Download the FITness report from
   http://www.nap.edu/openbook.php?isbn=030906399X
   Select one of the following three sets of competence areas:
   Set 1 from the Intellectual Capabilities
      Manage complexity (p.21)
      Organize and navigate information structures and evaluate information (p.23)
      Think about information technology abstractly (p.27)
   Set 2 from Information Technology Concepts
      Networks (p.29)
      Information organization (p.30)
      Universality (p.31)
   Set 3 from Information Technology Skills
      Using basic operating system features (p.38)
Using a word processor to create a text document (p.38)

Using instructional materials to learn how to use new applications or features (p.39)

a. Which of the subject matter areas did the set cover?
   i. Information
   ii. Information technology
   iii. Task support
   iv. Problem solving (Metacognition)

b. At which level of competence was the set?
   i. Skill
   ii. Understanding
   iii. Problem solving
Part III - Managing development of digital competence in organisations

The previous parts have considered the individual’s competence and learning. In order to consider organisational aspects of IT competence, we shift focus from individuals to groups. We will base the identification of groups on people who share a set of tasks, called a practice. Such groups constitute the units in a theory of learning at work within the class of situated learning theories. Situated learning refers to learning that takes place within the practice where the learning is applied.

Pedagogical theory – Situated learning – Communities of Practice
According to (Wenger, 1998) a community of practice (CoP) has three crucial elements; domain, community and practice. First, it has an identity defined by a shared domain of interest, whereby membership implies commitment to the domain. Therefore shared competence is an important factor that distinguishes the members from others working on the domain. Members in a CoP value their collective competence and learn from each other. Second, members in a CoP create a community through engagement in joint interactions and discussions, by helping each other, and also by sharing information. They also build relationships that enable them to learn from each other. However, members of a CoP do not necessarily work together on a daily basis. The third characteristic element of a CoP is the practice; the doing in a historical and social context which provides meaning and structure to the activities. The shared practice is created by practitioners who develop a shared collection of resources such as tools, experiences, and ways of addressing recurring problems. For example, a group of supermarked workers would constitute a CoP when they share the concern for the goods and customers, they interact, discuss and help each other, and they use common tools for sales and pricing of goods.

CoPs often differ from the formal organisational units, appearing neither on an organization chart nor on a balance sheet. In a small shop with a handfull of staff, managers may be part of the cashiers’ CoP, and in a large organisation, the accountants spread around in different departments may interact sufficiently to constitute a CoP.

Newcomers get socialised into a CoP by imitating its members, and also by getting punished or neglected if behaving in ways which are not acceptable in the community. The members may also tell newcomers explicitly how to behave, and the novices may have attended formal education which has prepared them for the introduction. When a community of practice receives a new member, it is mainly the newcomer who will have to adapt, while the community is less receptive for changing their practice.
### Pedagogical theory – Situated learning – Interaction between CoPs

In line with (Wenger, 2000) and (Cobb et al., 2003), we consider three aspects of interaction between CoPs: boundary interactions, brokers, and boundary objects. In *boundary interactions*, members from different communities take part in common activities. This might be short encounters, like when a manager calls the computer support for getting help in connecting to the network, or longer practices, for example when health managers participate in a course conducted by health information specialists.

A *boundary object* is a material thing which makes sense in more than one CoP, and which also has a structure that is common enough to be recognized in both CoPs (Star and Griesemer, 1989). Boundary objects are used for communication between CoPs, and they may provide a common understanding of a phenomenon as well as give rise to misunderstandings. A database could be a boundary object for accountants and computer scientists, where both parties would recognize its ability to store and retrieve financial data. However, the accountants would emphasize its role of representing the financial affairs of their company, while the computer specialists could regard it as an instantiation of a relational database management system.

*Brokers* are at least peripheral members of two CoPs and can introduce parts of practice from one CoP to the other. A headmaster could be a broker between the community of teaching practice and the community of school management practice in the town. Construction engineers could be members of engineering, architectural and construction work practices, providing some joint understanding between the three partners.
Chapter 11. Super-users

The learning aim of this chapter is to be able to identify groups with different roles as learners and supporters and to specify conditions for these groups developing into communities of practice.

11.1. Roles

The areas of IT use competence were identified as information, IT, task support and metacognition. Each of these has their specialists, while users in general will be specialists in their tasks and the domain of the system, see Table 10.

Table 10. Areas of expertise and corresponding communities.

<table>
<thead>
<tr>
<th>Role</th>
<th>Area of main practice</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information officers</td>
<td>Information</td>
<td>Accountants. Archivists.</td>
</tr>
<tr>
<td>IT personnel</td>
<td>IT</td>
<td>IT support staff. Software developers.</td>
</tr>
<tr>
<td>Super-users</td>
<td>IT + at least one of the other</td>
<td></td>
</tr>
</tbody>
</table>

Users are characterised by having the domain of the information system as their primary domain of work. Second, people having information as the main domain of their practice, like accountants and archivists will be called information officers. Third, IT personnel have IT as their main domain of work, so these are in the IT practice. Fourth, teachers enable learning and have metacognition in their competency base. People working in each of these four roles can constitute communities of practice, since they share a main practice.

Finally, some users, information officers or trainers develop specific skills in using computers, so they provide support to their colleagues, and this group will be called super-users. A super-user can also come from the IT side and adopt competence from one of the other areas. Super-users will be members of at least two communities of practice, hence become brokers between these communities. They can also become a community of super-users.

Users

For the majority of IT users, the technology is a means to get work done, and not an aim in itself. They find IT problems annoying and prefer to spend their time on their primary tasks. Their shared domains of interests are therefore not IT or data, but any other work area; hence they may constitute communities of non-IT practice. Correspondingly, the eventual learning of IT use taking place in these communities will be of secondary importance to the learning of the main tasks.
Information officers

Information officers are normally people of other professions than IT, having data management as their main responsibility. This could be accountants keeping the books, clerks doing data entry, statisticians producing reports, epidemiologists analysing data, surveyors measuring locations, or archivists storing and retrieving files. Information officers are the experts on information in the IT competence model. Having IT as their main tool for work, they often develop into super-users, thereby achieving double expertise in relation to the information system.

In larger organisations, there may be an accounting department, a central archive, or a management information systems group, each having a number of staff working closely together. They could constitute communities of information practice within their application area. However, such specialists can also be scattered around in an organization, leaving them few opportunities for developing into a community. In these cases, user forum meetings, e-mail groups, professional societies or the odd phone call may provide sufficient contact for their expertise to be shared and thrive. If they work in a place where there is also a community of information practice in the same domain, like the accountancy department or the central archive, these communities could provide the support for the scattered individuals.

IT personnel

Larger companies or agencies would have IT personnel who carry out a mix of tasks. Network administration and user support would normally be the two most time consuming ones, while procurement and application tailoring could be other tasks.

The idea of a community of practice is that people share a domain of interest, and we could say that the IT systems and their users in the organization is the domain of the IT personnel. They would normally share information about the technology and its users through lunch conversations, meetings, e-mail, documentation and random encounters in the corridors. Larger organisations could also have a database for storing user requests and responses, where the IT personnel can search for topics with which they are unfamiliar. In these ways they may develop a shared repertoire of cases, problems, software and users, so that they constitute communities of IT practice. IT specialists meet users in boundary encounters on the phone and face to face, helping out those who need more IT competence, and they learn about users’ tasks and information through interacting with them. They also have boundary interaction with other communities of IT practice, e.g. at computer vendors, thus keeping updated in the IT field.

Software companies and IT vendors can also have departments for support. For these organisations, their customers will constitute their users. A newly established, small company might just have a flat structure, where all members carry out development and support. These would constitute a community of practice, where the software product constitutes the shared domain of interest. A big vendor, on the other hand, might have a call centre in India with several hundred staff members who serve customers worldwide. If they have the opportunity
to communicate and exchange experience, they may also become a community of practice, where the users' requests and the corresponding responses constitute the shared domain. In between these extremes, there are many medium sized IT companies, where the user support is located in a department of a smaller size, such that the formation of the community is simpler than in the huge call centre case.

An IT department in a non-IT company would use the software and hardware vendors as their lifeline for support. They would engage in boundary interactions with the vendors, and the software and documentation would constitute the boundary objects of these practices.

**Teachers**

Larger organisations have human resource departments where educationalists are hired for organising and planning training, who may also act as instructors themselves. Schools are obviously special in this respect, as their main staffs have formal pedagogical qualifications. They would normally constitute one or more *community of teaching practice* in each school.

School teachers and business instructors sometimes also do IT training. In schools, IT competence could be an end in itself or a means for the students to learn other topics. In the latter case, the teacher may be fluent on tasks and information but short on the technological competence. Professional teachers bring training methodology and learning competence including metacognition into the realm of user support and training. This pedagogical competence is hardly found amongst IT personnel or information officers.

**Super-users**

Super-users are users, information officers or trainers who have specific IT competence and have taken on the role of supporting their peers in an organization (Boudreau and Robey, 2005, Olfman, et al., 2003). ‘Key users’ (Fitzgerald and Cater-Steel, 1995), ‘business coaches,’ ‘subject matter experts’ (Olfman, et al., 2003), ‘power users’ (McNeive, 2009), ‘computer gurus’ (McNeive, 2009) and ‘local heroes’ are other terms for this role. Super-users could have a primary domain of work completely remote from information or IT, for example nursing, sales or farming. They would therefore belong to two communities of practice; one on the IT side, and another on their primary domain, and they would also be brokers between these communities, see Figure 51. They could influence the communities of IT practice with their main competence, and introduce IT competence amongst others.
Figure 51. Superusers (yellow) being members of at least two communities of practice and brokers between these communities.

The text boxes present three super-users. Mozhdeh had some experience as an information officer when she did archiving. She became a super-user of an archiving system after her job was changed, but she still draws upon her knowledge of archives in her super-user role. Oksana is a super-user of a system she uses frequently in her accounting job. She knows the information in the system, how to operate the IT, and how it supports the tasks. Sigrun has a computing background and was selected super-user for a web publication system. She is familiar with how it can be used for creating structured web pages.

While these three super-users had been appointed, super-users emerge also when no formal appointments are made. In a purchase department of around 100 staff in a Finish company, all staff were provided training when a new information system Erchive was implemented. I was introduced to Erchive in a training course. The course covered the IT system but not archive codes and whether a document has archive value. Many user requests concern whether a letter should be archived, and then I have to find out whether it has archival value. There is a list where I can look it up, and I can also call the central archive if in doubt. Other user requests concern how to operate the system. Also, I solve logon problems and upgrading to the recent version of the browser. They need that when Erchive is updated.

Each department has a super-user, and we meet 4 times a year. We discuss changes we would like and communicate these to the IT department. We also get informed about changes and have to bring the news to our local colleagues.
A community of super-user practice could emerge if they engage in activities where they meet and exchange experience specifically on their super-user activities and role. Almnes (2001) conducted a study of superusers amongst nursing home personnel, and McNeive (2009) reports from nurse superusers in a hospital. Both emphasize that belonging to a group is important for superusers, since their role is the only ones of its kind amongst those whom they meet daily. In addition to group meetings, e-mail lists, newsgroups and lists of frequently asked questions may also be advantageous. The organised group should also provide the necessary opportunities for the super-users to update their skills, whether new software or other upgrades necessitates it. An accounting company formalised their super-users into a group with a coordinator in charge (Åsand and Mørch, 2006). Mozhdeh and Sigrun belong to such groups.

An organised group has to cross the organisational units. In the Finish company, the department was divided into three product lines, which again was split into a total of 11 groups (Sykes et al., 2009). A lot of the help was given across groups and also across the product lines.

The super-user is the first person in the support chain. She should handle most of the normal requests dealing with use of the computer system, for which she has received special training. In addition, the super-user should be able to take care of user requests concerning the operating system and standard tools. Both Mozhdeh and Oksana help out on information issues like getting the information into the right field and on IT issues like updating software.
Communicating frequently with users, the super-users receive requests for changes of computer systems. They are in a good position to communicate these requests to the computer department or those in charge of the software and hardware. This aspect of their role should be used so that the requests from the users are taken into account. The meeting of super-users could also be an agenda for discussing and distilling such requests. This is a regular item on the agenda in Mozhdeh’s group.

The super-users could also act as communication links in the opposite direction. When system updates occur or tools are replaced, the users need to be informed and trained in the altered functionality. The super-users could naturally take on this obligation, and provide small training sessions locally if needed.

The superuser should be given responsibility of the resources necessary for carrying out the role (Almnes, 2001, McNeive, 2009). A dedicated amount of time for the super-user activities is recommended (Almnes, 2001, Åsand and Mørch, 2006). The resources for sending users for training, is an obvious responsibility that should be attributed to the superuser.

The selection of people for the superuser role is an important issue for creating a well functioning support system. They could preferably be amongst those whom people often calls for assistance, which would guarantee a caring person. McNeive (2009) emphasize that they should be champions for the changes that the computer system should support. Almnes (2001) warns against local managers, since they are often too busy and not always available. In addition, many people do not like to expose their misunderstandings to their superior. Sigrun is such a boss-and-super-user which is not recommended, but she mainly helps out people outside her subordinates.

People who are unwilling to take on the role should also be avoided. They may behave hostile or less caring towards their peers, and if so, the users will soon stop consulting them.

Since super-users help others solving IT problems and also guide them in problem solving, they need problem solving competence. In addition to computer skills, the super-user also ought to have skills in guiding others, something which the trainer need in particular.
Some IT personnel like Sigrun change their career into other occupations, and they will naturally be more skilled in IT than their peers. If they have the necessary inter-personal skills, they would become very good at supporting colleagues as well as communicating with the IT specialists.

The significance of super-users is described in a study of implementation of a companywide information system where adoption was slow (Boudreau and Robey, 2005). It was found that most user groups did not attend the initial training programme, and when the software was implemented, the users found ways of avoiding using it. Rather than entering data, they got some groups of information officers to carry out the data entry. Later, some self-initiated super-users found out how to operate the new software, and this competence was spread in the organisation as folk wisdom. In the end, most people used the system, after the user communities had found workarounds and tweaks in order to get the system performing as needed. This competence was also spread throughout the relevant user communities. The account of this story in Boudreau and Robey's paper is told through a theoretical lens of human agency. If we regard it with the ideas from communities of practice, the organisational change eventually took place due to the broker role of the super-users.

11.2. Trainers

In a community of practice, the practice would constitute the tasks of the majority, while the minority would be peripheral people who could learn the tasks through interacting with the majority. A training session is of an opposite kind, where the majority of learners is supposed to adapt to the minority of trainers. Unless all trainers are super-users, it is highly unlikely that the trainers and the trainees develop a common knowledge base during a two weeks in-service training session, so such activities constitute boundary interactions rather than CoPs.

In-service training is acknowledged by Wenger (1998) as useful when providing a place for reflection on the practice, and as an opportunity for getting to know people whom one would otherwise not meet. However, Wenger remarks that often in-service training or education are too detached from practice to foster learning which strengthens the individuals’ participation in the communities. This could easily happen when the tasks and information are not included in IT user training, or when tasks and information included do not match the learners’ experience. Including super-users amongst the trainers could bring training closer to practice. Super-users who area also ordinary users could bring task support into training. Information officer could include relevant data which users recognise. Super-users who also belong to the community of teaching practice could teach ways for further learning, like problem solving competence.

Leaving the training to IT personnel only creates the risk of restricted interaction between users and trainers in the classroom. Including also a super-user improves the relevance of the training and creates variety amongst the trainers. The latter is in general an advantage for learning. Both Almnes (2001) and McNeive (2009) recommend that super-users should be involved in planning and conducting IT user courses, in order to include user tasks in the
training contents. Also, users feel more comfortable by being taught by one from their own profession than by a computer specialist.

Oksana lectured and guided other users on the human resource system. Her experience enabled her to convey how the system should be used to support the accounting tasks. She could also bring her inventions to the larger audience.

11.3. Organising training

The accounting company mentioned above appointed one super-user per 10 employees, and gave the super-users the obligation of training the others (Åsand and Mørch, 2006). Being organised in community of super-users helped them to become capable of carrying out this task. Oksana and Sigrun have similar roles, and they support 250 users. However, the information systems for which they are super-users are only used now and then by most of the staff.

The Finish company trained all 100 users simultaneously (Sykes et al., 2009). The study revealed that there was a positive correlation between how often gave or received help and how much she or he used the system. If the goal is high system use, helping each other after training is therefore efficient. Seen from the individual user point of view, it would constitute a facilitating condition in the technology acceptance model (Venkatesh et al., 2003), see Figure 35, p.75.

When a new system has a large number of users, training is costly and it leads to disruption of the organisational performance. The latter is unacceptable when clients have to be cared for, like in a hospital, or when processes cannot be halted, like the power plant. To reduce costs and avoid disruptions, training is often provided only to a group of super-users, who are selected from each organisational unit. All staff is given access to user documentation, and the super-users are thereafter supposed to help out the rest of the staff when needed.

11.4. Summary

In addition to IT skills and understanding, users need to learn also task and information competencies. Superusers and information officers have the right background for including these competencies in training and support. Enabling the developing of these groups into communities of superuser practices and information practices could boost their contributions to IT competence in the organisation.

7. Identify, organise, authorise and cultivate superusers.

8. Include IT, information and use competence in support and training.
11.5. Super-users as leaders

The extent of people’s use has been employed as a measure of success of introducing an IT system in an organisation (Davis, 1989, Venkatesh et al., 2003). Perceived usefulness, perceived ease of use, social pressure and facilitating conditions have been found to influence the amount of use (Venkatesh et al., 2003), see Section 6.2. Documentation (Chapter 2 and 4-6), training (Chapter 7) and IT support (Chapter 12) constitute aspects of the facilitating conditions, while super-users are members of the community of user practice, hence they can exert pressure on system use amongst its members. In order to know whether to put the effort into training, IT-support or super-users, knowing the relative influence of each of these factors would be needed. No comprehensive studies of all these factors have been carried out, but a comparison or co-workers’ influence versus training provides some insight.

In a non-profit organisation in the US, half of the 200 employees responded to a survey on IT use, perceived usefulness, perceived quality of user training, amount of user training, and coworkers’ IT use (Gallivan et al., 2005). 80% of the respondents were female, and the large majority had a university degree. The extent of coworkers’ IT use had the strongest impact on an individual’s use. Coworkers’ perception of the training quality and to a smaller degree also the individual’s perception of training quality influenced the extent of the individual’s use. The amount of user training and the perceived usefulness had no influence. The latter contradicts the technology acceptance model (Venkatesh et al., 2003), and there is no obvious explanation for this finding.

This study points to the importance of what happens in the local work group and the possible futility of putting many resources into training (Gallivan et al., 2005). Given that people follow their colleagues in their computer use, and that a new system is to be introduced, people will only use it if their coworkers do. For an innovation to be taken up, some have to lead, such that the rest of the community can follow suit. Super-users are in a favourable position to be the leaders, since they are well versed in IT in addition to being a coworker of the others. In order to become a strong leader, super-users would need to be well trained and preferably a member of a community of super-users, such that they also can follow each other. Consequentially, providing thorough training for super-users would be more efficient than training the whole group of users.

The study also points to that the quality of the training is more important than the quantity (Gallivan et al., 2005). Since being able to help others would probably ease the leadership role of the super-users, their training should particularly emphasize understanding and problem solving skills.
Chapter 12. IT support

The learning aim of this chapter is to be able to support users so that they increase their IT user competence.

Learning can take place anywhere and anytime, but some activities are carried out with learning as their main purpose. Having identified the CoPs relevant to building user competence, we can proceed to characterise the activities where the different communities meet and where learning is supposed to take place. These activities encompass training and support.

12.1. Support as boundary interaction

Support is normally a boundary interaction between an IT specialist or a super-user and a user, where the supporter is supposed to help out when the user is stuck. The IT is a boundary object in the interaction, and documentation and data may constitute other boundary objects.

The support interaction normally last for a short time, like a few minutes. Contrary to training sessions, the topic of the support sessions are initiated by the users and the support is targeting the user’s current problem. Support personnel would normally not prepare specifically for a certain encounter, but they may subsequently note down information about the support.

While super-users have the advantage of knowing the work task and the information, IT personnel would be the support expertise for IT problems. Also, staff in an IT department in a larger organisation would normally have user support as a main part of their job, while helping others constitutes an additional burden for super-users.

When users and IT personnel meet, they talk about the same phenomena in different ways. For example, when a user say

\[\text{we have a group of students who cannot synchronize}\]

the technician talks about

\[\text{IP-errors or server-errors (Kanstrup and Bertelsen, 2006)}\]

We see the terminology problem from search in documentation (Section 8.2) reappearing. When the user and supporter are co-located, they also have boundary objects like software and documentation which they can look at, point to and interact with, and they can observe each other’s actions. When helping out on the phone, the oral interaction is the only communication channel. The following conversation took place when a user of a printer/copier called the vendor’s support centre for help. The troubleshooter searches a knowledge base for finding possible solutions (Crabtree et al., 2006):

\[\text{Troubleshooter: OK, and what's the problem you're having with the machine?}\]
Customer: I’m getting poor quality prints – sort of smudges on them.

Troubleshooter accesses knowledge base and selects ‘image quality’.

Troubleshooter: When it’s printing?

Customer: Yes.

Troubleshooter: OK, do you get this when it’s copying?

Troubleshooter: So you get it printing and copying and they’re like smudges?

Troubleshooter selects ‘smears and smudges’ in knowledge base.

The troubleshooter has to translate the vernacular of the customer to the specialist terminology. Likewise, the customer has to grasp the technical terms ‘image counts,’ ‘xerographic’ and ‘fuser module’ (Crabtree et al., 2006):

Troubleshooter: You know your image counts, which is the amount in thousands of copies that the xerographic and fuser module have done, check them just to see if they’re running over their copy limit and causing that problem for you.

Troubleshooter: Of course, yeah, take your time, that’s fine.

48 second pause.

Customer: Where do I find them?

Troubleshooter describes how to use the menus to find the counts and customer goes to find them.

70 second pause.

Customer: 43

Troubleshooter: Hi, that’s from your fuser module

(writes down count).

Troubleshooter: OK could you - do you know where the xerographic module is in the machine?

Troubleshooter: OK, I’ll tell you exactly where it is as there’s something I want you to try, just to see if this will rectify the problem for you – if you open the front door of the machine ...

After having negotiated a common understanding of the terms, the troubleshooter instructs the customer to carry out an operation, which will lead them through the first cycle of troubleshooting. If the customer remembers the steps such that he can do them without support the next time the problem occurs, he has learnt this particular way of troubleshooting the machine. Since the troubleshooter has this as her main job, she is likely to pick up users’ terminology and the technical trouble. Efficient supporters interact with a variety of users and experience a multitude of user problems and terminology, and they are 2-10 times as efficient as supporters with more homogenous interactions (Chi and Deng, 2011).
The interaction between 11 IT supporters and 61 users was observed during implementation of a work flow system in a US bank (Santhanam et al., 2007). In addition to support, several meetings were organised for the users and IT personnel to discuss issues. It was found that users mainly learnt IT skills during interaction (Santhanam et al., 2007). The IT personnel gained know-why, i.e., they understood how the IT was embedded in users’ tasks during the same encounters. Also understanding of the technology was shared. User competence, particularly on how the IT is used in the task, is hence introduced into the community of IT practice and shared amongst the IT personnel (Santhanam et al., 2007).

12.2. **IT support versus super-users**

There have been several studies on the type of support which users prefer, and the results seem to depend on many factors.

Interviews of 40 users with education above high school in the US, showed that users preferred asking IT-personnel and colleagues at roughly the same rate (Novick et al., 2007).

In a survey amongst university staff in Norway, 49% preferred support from the IT services, while 31% chose colleagues (Nilsen and Sein, 2004). There were 222 responses to the questionnaire (37% response rate).

A survey of US middle level managers’ opinion on support gave the opposite result. 38% preferred super-user support, 26% other colleagues and 19% an IT centre (Govindarajulu et al., 2000). These results are based on 98 informants (response rate 11%).

These three studies agree that users prefer support from both IT personnel and super-users, but there is no consensus on which one is preferable. A survey in Norwegian organisations investigated some possible causes for choice of support (Munkvold, 2003). Responses came from 277 informants, yielding a response rate of 41%. Short distance to the IT support personnel made users go there, while when the distance was longer, users preferred colleagues. Higher skilled users consulted the IT support to a lesser degree, while they solved problem more often themselves than did less skilled users (Munkvold, 2003).

Convenience and skills may therefore be reasons for choosing one source of support to the other.

12.3. **Support quality**

In a qualitative study, 39 users in the Finish public and private sectors were interviewed about their learning preferences (Korpelainen and Kira, 2010). In general, they prefer learning IT use on the spot. Formal training courses take too long. Said one of the interviewees:

> There are so many [user training] courses and other rubbish that I can't be bothered to do an extra thing. I haven't left a single task uncompleted, so why would I bother... I don't need the extra information, and I am not interested. I am only interested in being able to do my tasks; I just want to find the information and complete my tasks. That is all I need the system for. (Korpelainen and Kira, 2010)
Also, users hardly read documentation (Novick et al., 2007), they rather ask others, unless they try and err or succeed. Getting support is therefore essential for most users both for learning and for solving IT problems without learning how to do it themselves the next time.

A survey of 484 users in a US university examined the correlation between support factors and user satisfaction (Shaw et al., 2002). The factors which influenced satisfaction the most are listed in Table 11. Factor 1, 5 and 6 are all qualities of IT support. Factor 3, user understanding, is partly influenced by training and support. Factor 4, software upgrades, is also a product of decisions in IT departments. Many users get annoyed when new upgrades appear, since they have to relearn the software, however, others push for new versions.

Table 11. Factors influencing user satisfaction (Shaw et al., 2002).

<table>
<thead>
<tr>
<th>Rank</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fast response time from system support staff to remedy problems</td>
</tr>
<tr>
<td>2</td>
<td>Data security and privacy</td>
</tr>
<tr>
<td>3</td>
<td>User’s understanding of the system</td>
</tr>
<tr>
<td>4</td>
<td>New software upgrades</td>
</tr>
<tr>
<td>5</td>
<td>Positive attitude of information systems staff to users</td>
</tr>
<tr>
<td>6</td>
<td>A high degree of technical competence of systems support staff</td>
</tr>
</tbody>
</table>

In general, the findings point to the central position of user support and learning issues for user satisfaction with IT. Software upgrades and response time were the only actors found to correspond with studies from the beginning of the 90’s.

When broken down into three distinct user groups, administration, academics and students, there was a great variation in the factors. This corresponds with a previous study, concluding that the specific activities of a user department influences its perception of IT support (Speier and Brown, 1997).

Users’ opinion of the performance of the IT support gave the lowest score to documentation to support training (Shaw et al., 2002). This issue is not amongst the general top factors influencing satisfaction, since users regarded it as less important. However, non-academic users had this item in their top factors of dissatisfaction (Shaw et al., 2002). Knowing that reviewing material from training is twice as successful as searching for help (Novick et al., 2009), putting more effort into training material could save the IT supporters from many encounters.

The physical place where users find IT support personnel has been called a helpdesk, while the phone call support is called a helpline. A survey of user satisfaction with helpdesks and helplines in the Netherlands compared user preferences (van Velsen et al., 2007). There were 64 responses concerning the helpdesk and 242 (11%) for the helpline. Concerning the helpline, user satisfaction depended to a high degree on the quality of the solution which the support personnel came up with. Surprisingly, the users who contacted the helpdesk were more satisfied when having a good time at the helpdesk, while the solution was of secondary
importance. Thus, the helpdesk should have friendly staff, while the knowledgeable ones should be working on the helpline.

12.4. IT departments

Information Technology Infrastructure Library (ITIL)

Communication tools and knowledge databases for IT supporters

See Bruton (2002) How To Manage The IT Helpdesk - A guide for user support and call centre managers, chapter 23

9. Provide a variety of support channels and frequency.

10. Train local groups of users, not only individuals.
References


© Jens Kaasbøll, 13th April, 2012

*Creating formulas using cell ranges in an Openoffice calc spreadsheet* 2010. Directed by COL CCNC. YouTube.


© Jens Kaasbøll, 13th April, 2012


REYNOLDS, G. 2010. Presentation zen design: simple design principles and techniques to enhance your presentations, Berkeley, New Riders.


