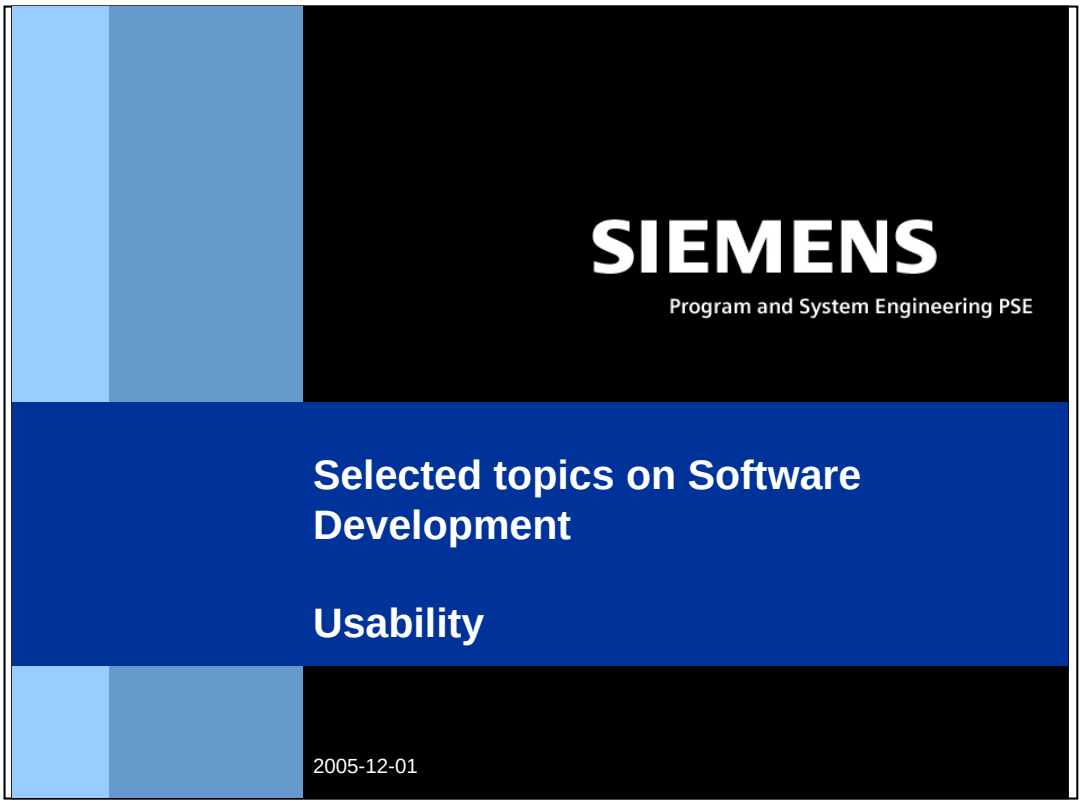
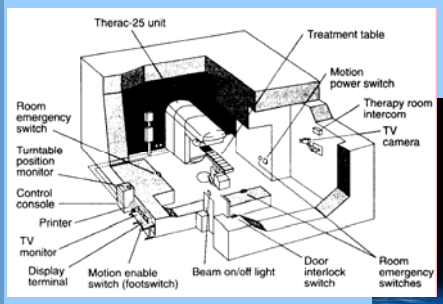

SIEMENS
Program and System Engineering PSE
Selected topics on Software Development
Usability
2005-12-01

Computer errors that made the headlines

Therac-25 – at least 5 dead



Discovery – satellite dish misadjusted

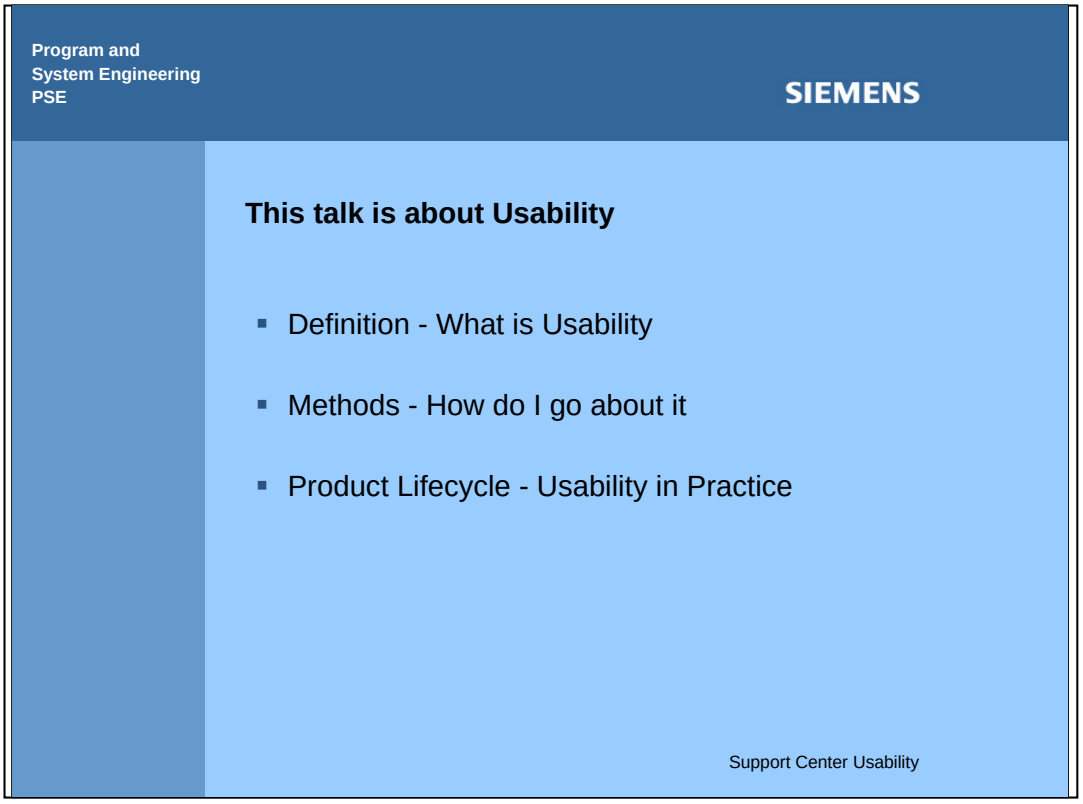
Support Center Usability

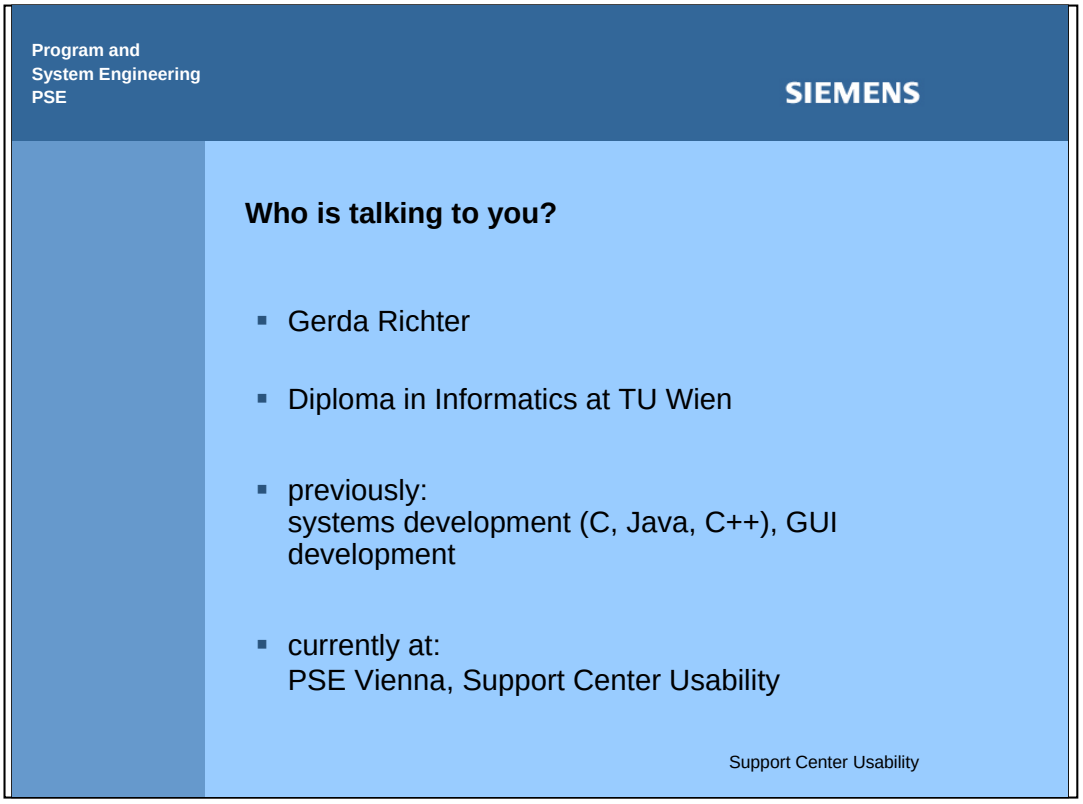
Therac-25:

a radiation therapy unit. A lot of things went wrong during development, among others resulting in a bad interface that caused at least 5 people to be killed by radiation overdose

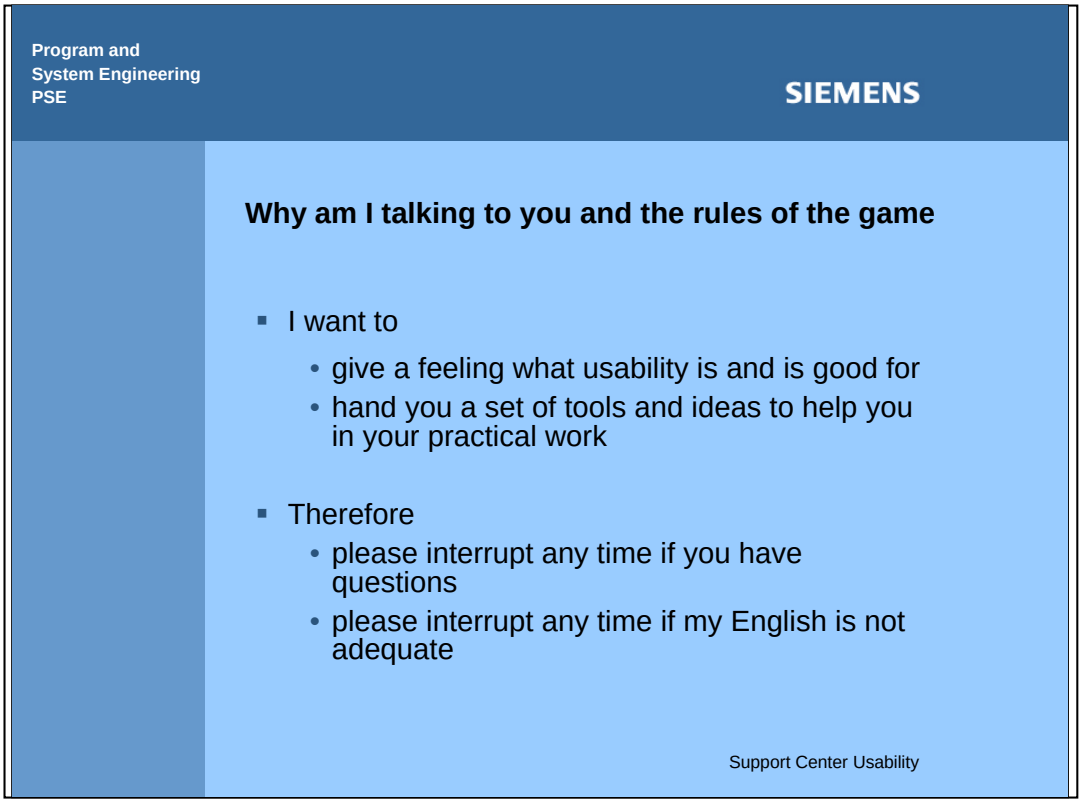
Discovery:

tried to place a satellite dish in space, the dish adjusted itself according to the data that had been entered. Unfortunately the person entering the data thought the number he typed in was measured in feet, when it was in fact measured in miles – resulting in the dish looking into outer space instead of onto earth.

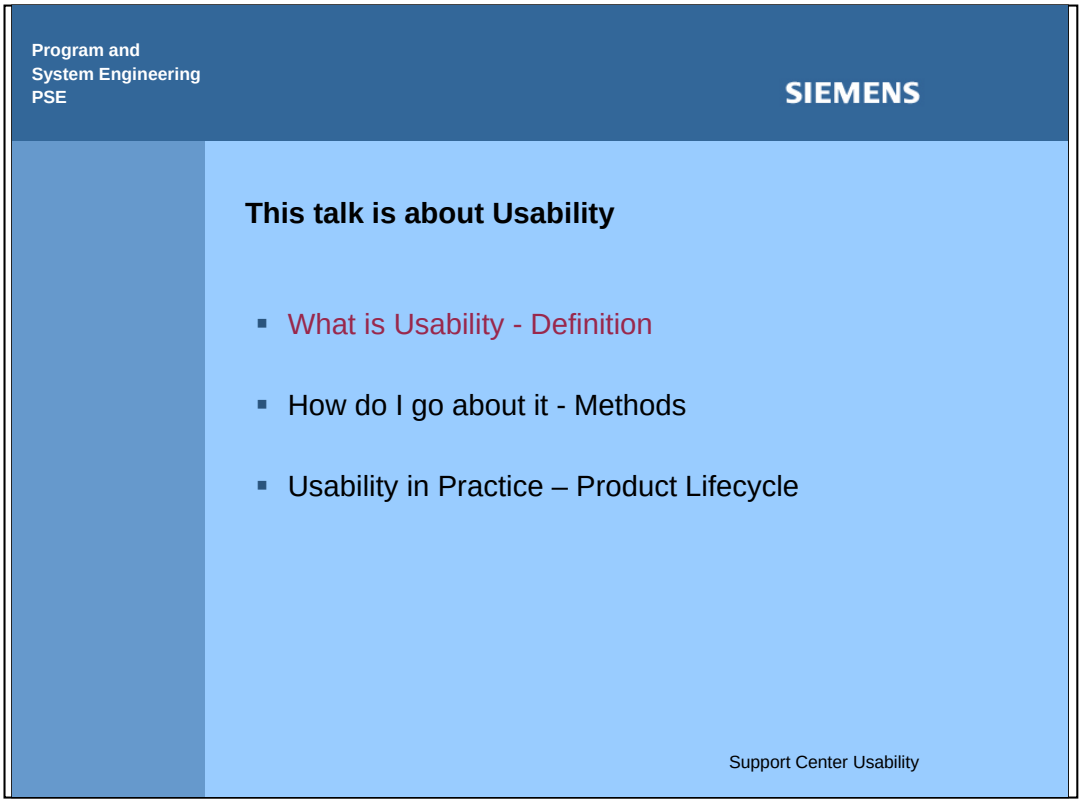

Program and
System Engineering
PSE
SIEMENS
This talk is about Usability
Definition - What is Usability
Methods - How do I go about it
Product Lifecycle - Usability in Practice
Support Center Usability

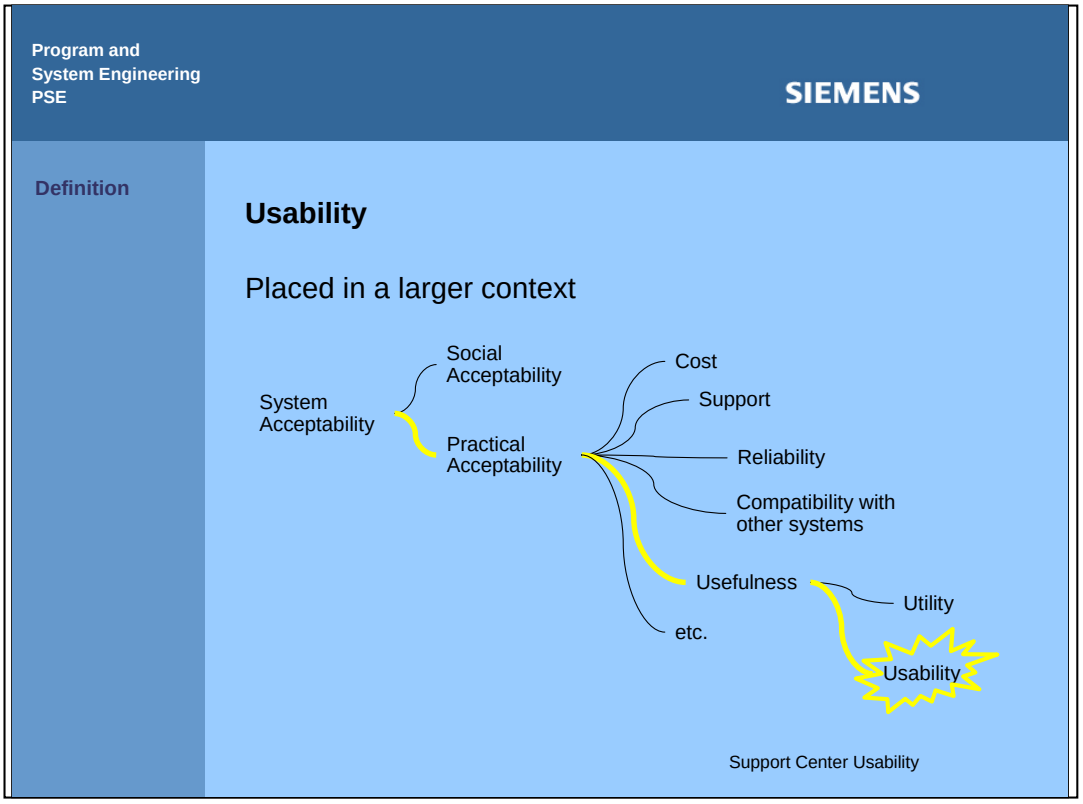


- Support Centers provide in-house consulting on their various topics.
- We consult on usability, others on testing or project management or …

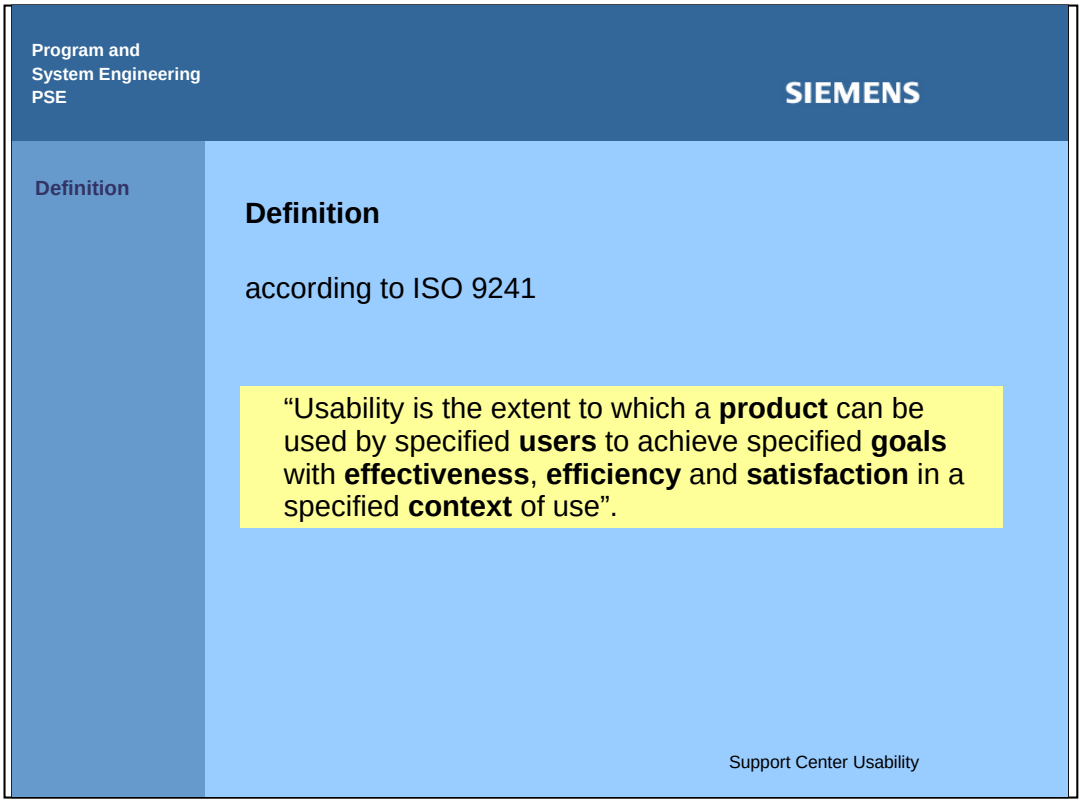


•The slides are available for download, they include notes containing some additional information

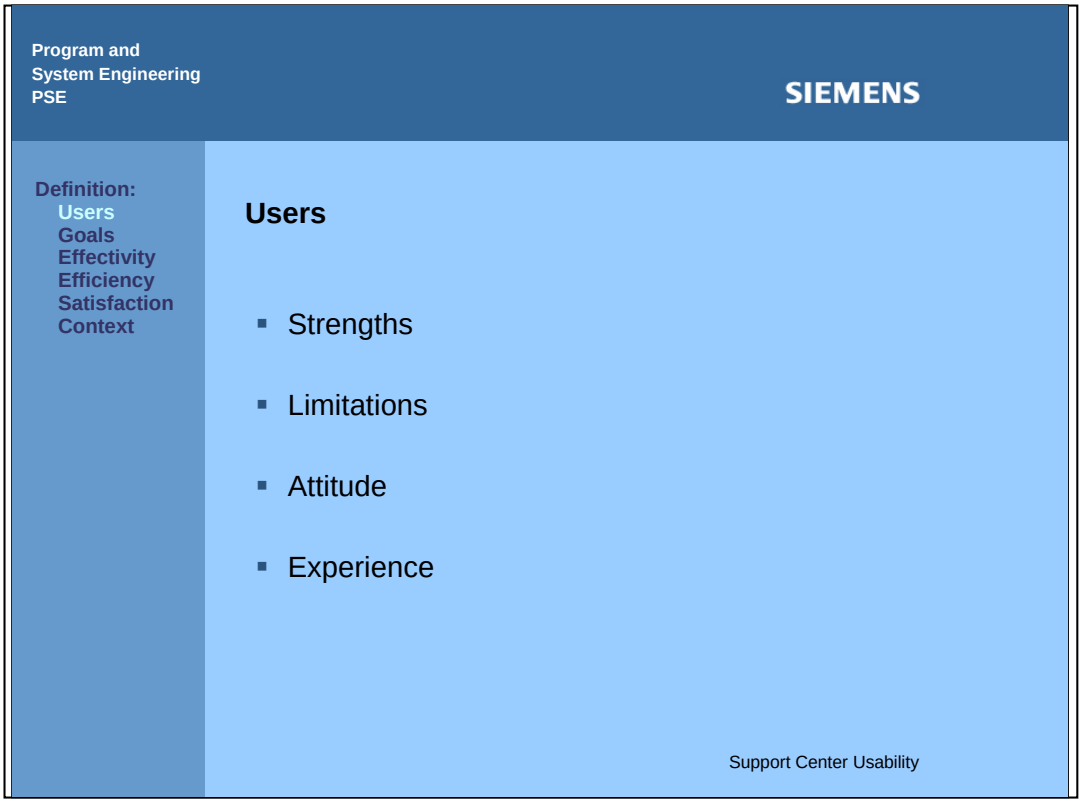

Program and
System Engineering
PSE
SIEMENS
This talk is about Usability
What is Usability - Definition
How do I go about it - Methods
Usability in Practice – Product Lifecycle
Support Center Usability



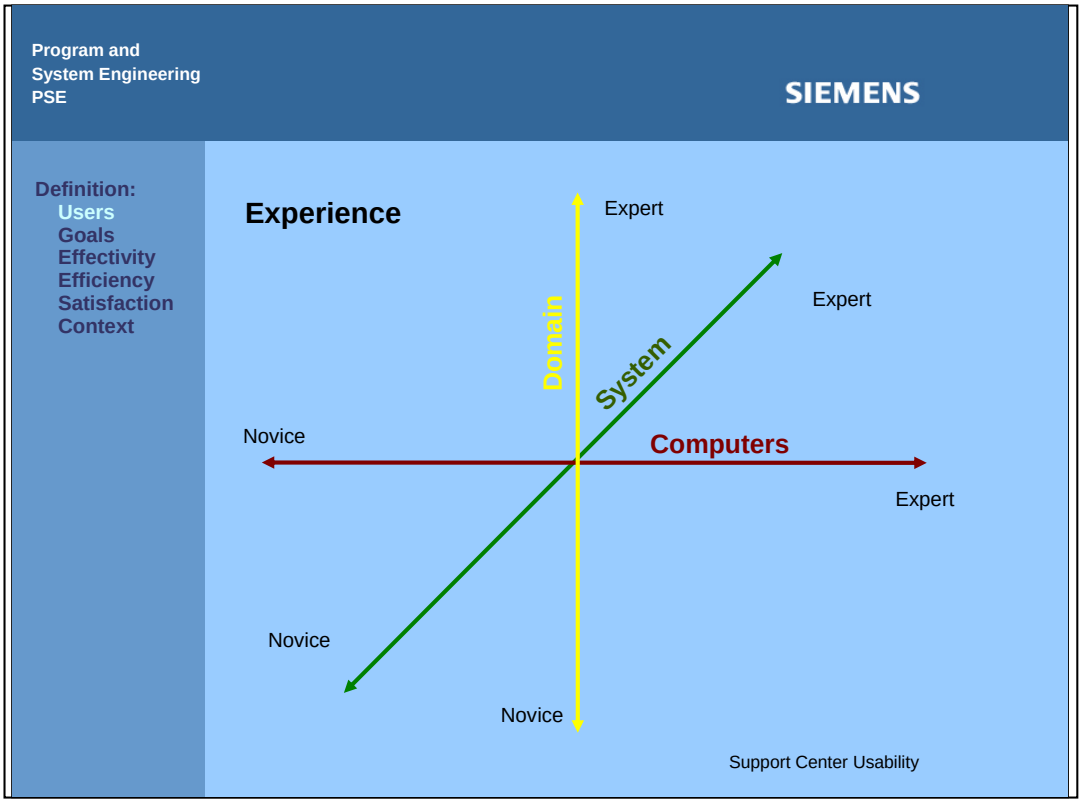
- When you buy a SW or any other product a lot of considerations influence your decision, whether consciously or not



- Key elements of the definition are highlighted – those are the points that can bear a closer look



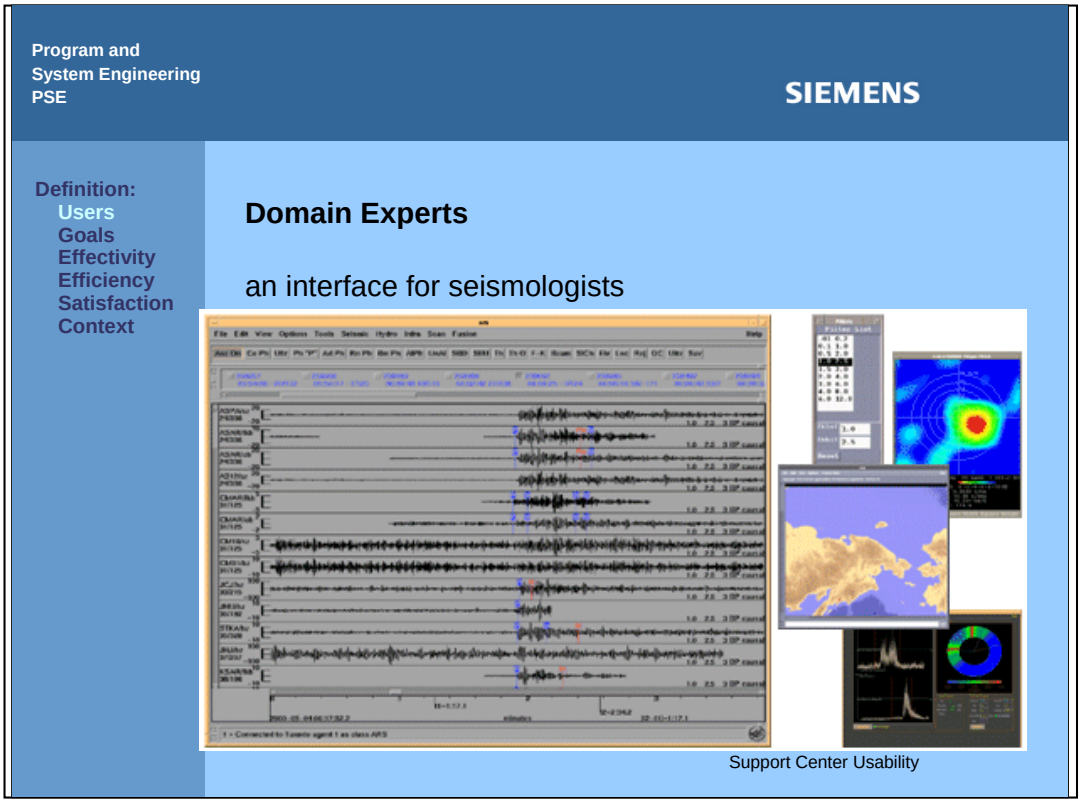
- have certain characteristics
- Strenghts: knowledge, abilities
- Limitations: What kinds of disabilities should be considered?
- Attitude: regarding system, computers in general, the task that is to be accomplished, etc.
- Experience: regarding system, what kind of learning,



- One of the categories to fit users into is their expertise
- domain:
  - Novice: much to explain, few domain terminology
  - Expert: information can be presented in a very dense way, not much explanation needed, full terminology

- System:
  - novice: finding needed features or function. How to go about reaching certain goals
  - expert: has a good idea how to go about things – on the other hand, the experienced user is quite convinced to know where to find what – expert is usually only expert of a part of the system

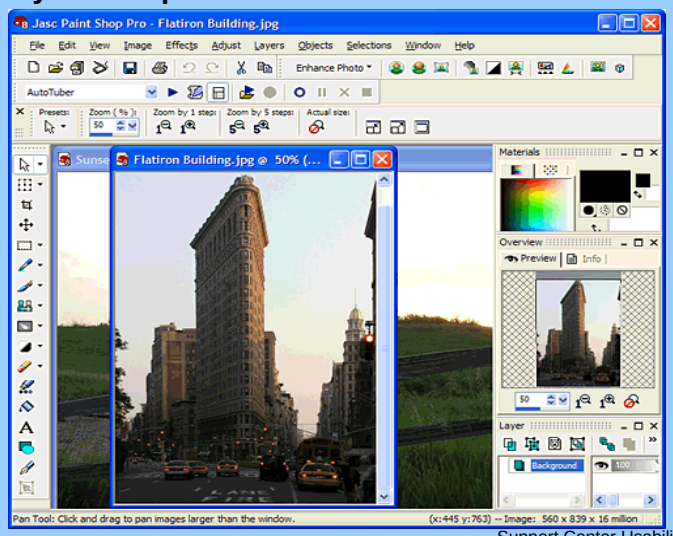
- Computers:
  - novice: is the mouse known? Can it be used? What about concepts like drag&drop? What are the limitations of a computer?
  - Expert: Has a good idea about underlying workings, what is possible, what isn't



- others may see that this is an interface for seismologists, but maybe not even that.
- For them the screen consists of a lot of black squiggles

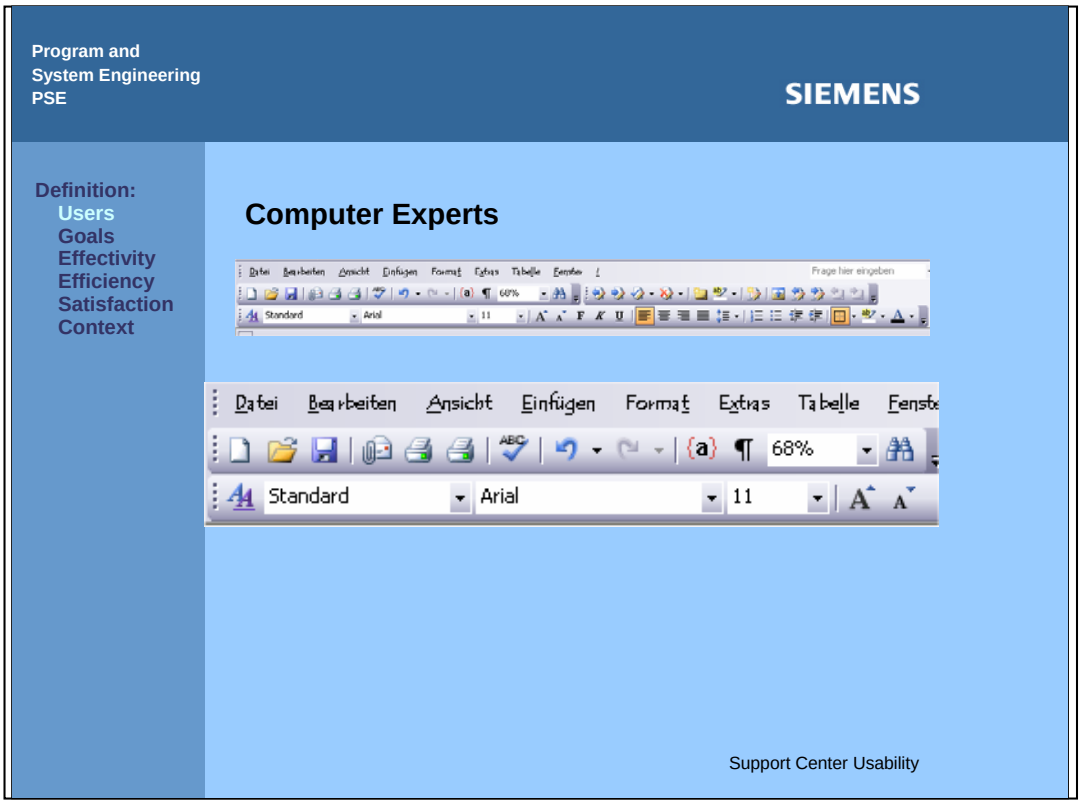
Definition:
- Users
- Goals
- Effectivity
- Efficiency
- Satisfaction
- Context

System Experts



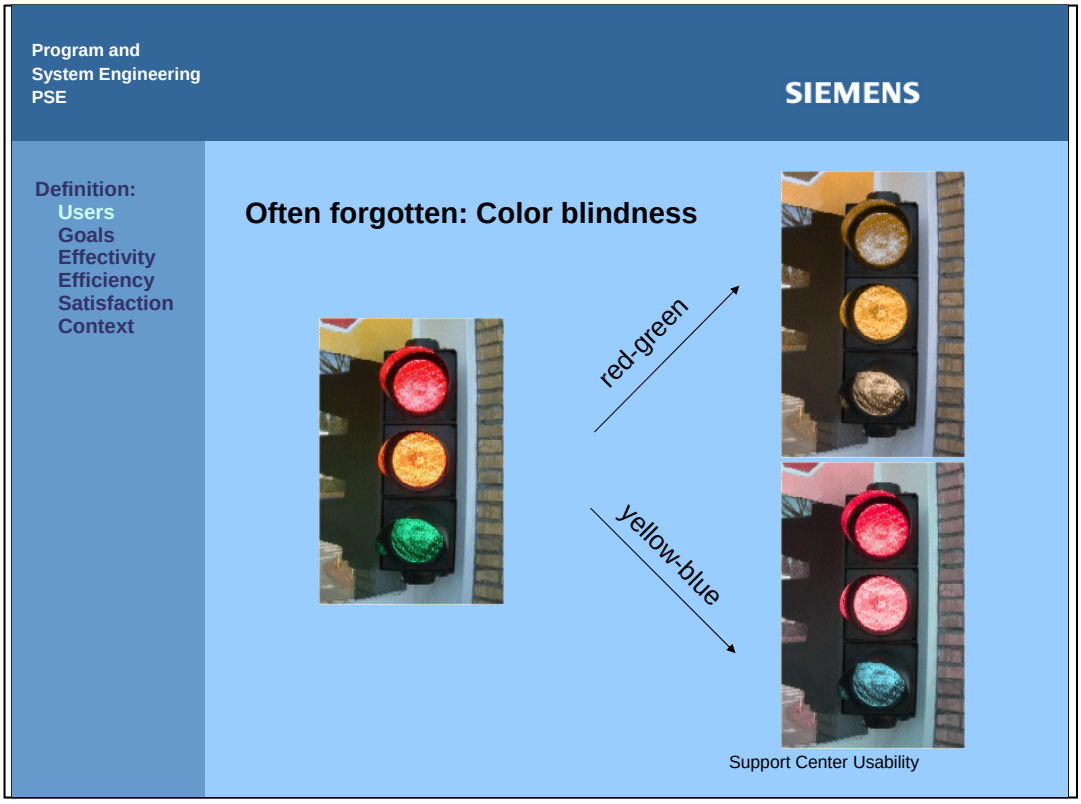
- You'd expect there to be a feature for cutting out the building and copying it somewhere else
- You'd expect there to be a feature for removing red eyes from a fotograph
- You'd expect there to be a feature for painting the sky a convincing green or purple

- but where are they?



- an interface where things like cut, copy, paste are not part of the toolbar, as the person uses the key-abbreviations anyway
- An interface using mouse gestures

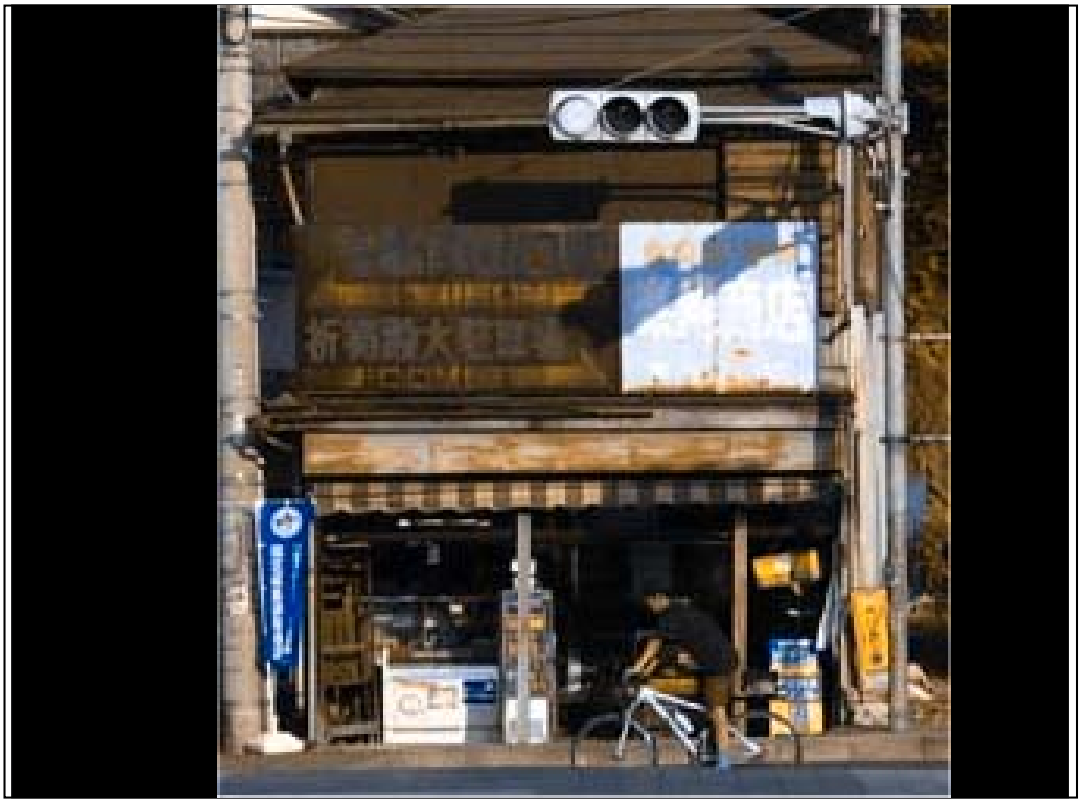
- Children versus grown-ups, elderly people, teenagers, twens
- Gender: There are some statistical differences. E.g. Men have better orientation, women are better at multitasking
- Ethnic groups: religious beliefs, unintentional puns, jokes that are taken badly
- Languages: Are the users all proficient in my language, or do they have different levels of expertise? Do they speak different languages and not have a common one at all?
- Learning styles: different speeds in learning, finding and application of patterns or not
- reasoning abilities: finding of patterns, application of analogies
- Disabilities: Do I need to accomodate hadicapped people? In SW for the control of a powerplant probably not. In a website for students probably yes. One group of people with a handicap is quite large and very often overlooked -> next slide



- A traffic light as it appears to most people.
- That we see it like this is due to how the eye is constructed: three kinds of receptors on the retina, each catching certain types of light – if one or more of the receptors are not working, we call this color blindness – depending on the kind of receptor either “red-green” or “yellow-blue”.
- Red-green, making the traffic light apppear like the picture in the upper right corner (red and green cannot easily be distinguished)
- yellow-blue, making it diffucult to distinguish between red and orange – see the picture in the lower right corner.

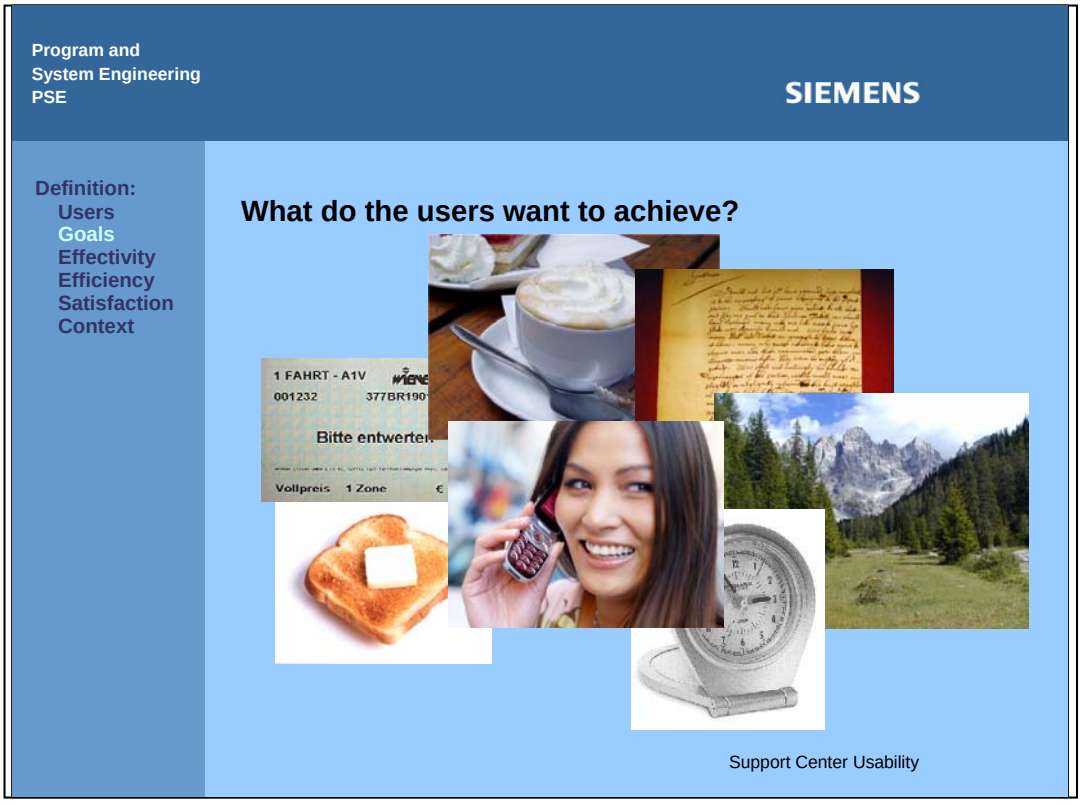
- With traffic lights in Europe this is usually no problem, as they are upright...

- Sideways installed traffic lights may problematic.
- This is a traffic light as it appears to most of you. Those among you suffering from colorblindness will see it differently and those suffering from red-green blindness will see it something like on the next slide



- Now – where is the green light?
- There's an international convention, placing the green light on the side, where the car is supposed to go (on the right-hand side in Czech Republic)
- In most of Europe the green light would be on the right-hand side, in UK it would be on the left-hand side.

- So if you are red-green color blind better keep in mind which side of the traffic light is the one for which you ought to stop if it is shining...



- the goals of the user may be quite simple or quite complex
- Something to eat,
- a ticket to somewhere
- a refreshing cup of coffee
- to write a letter, to layout a written text beautifully, to create a handout for role-playing
- to manage pictures, to print them, to create them, to take them
- to be reminded of something, to get up at the right time
- to call a friend, to have a nice new toy with lots of cool features

Users' goals

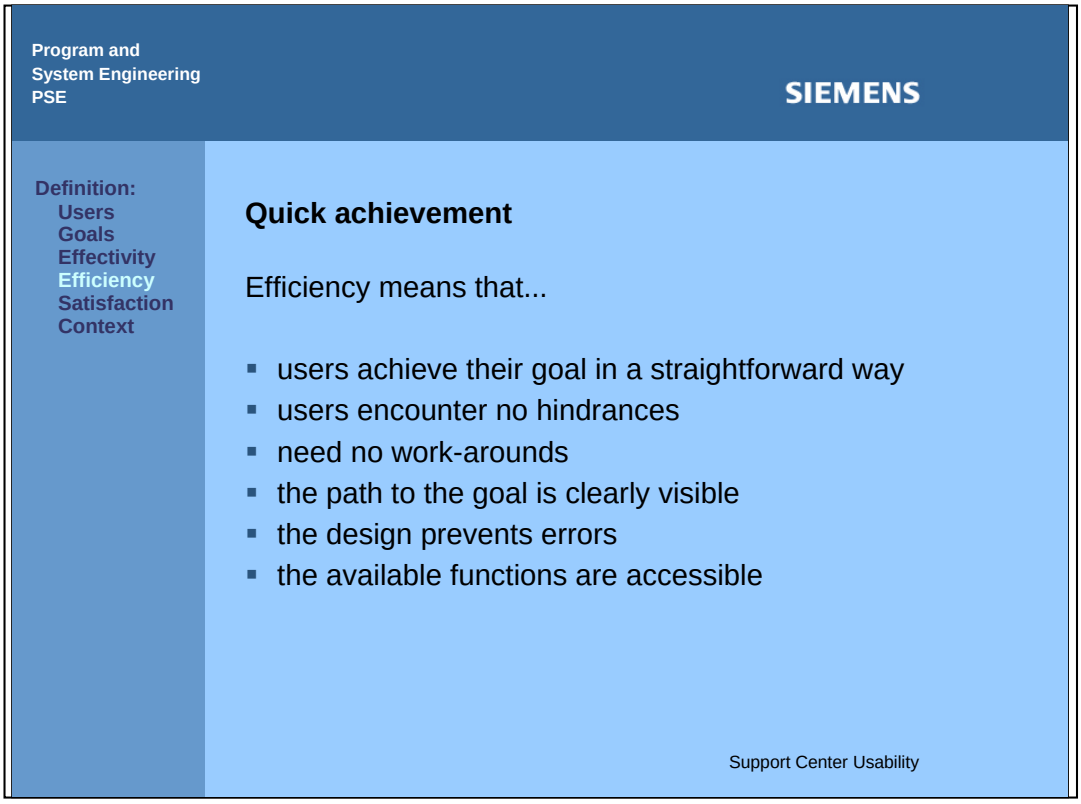
- Crucial point for any product to know this
- Output of requirements engineering
- Usability engineering assumes the developers know those goals
- Product design depends on the users' goals

Support Center Usability

Achievement

Effectivity means that

- Users achieve their goal
- Need not give up in frustration
- Are prevented from errors that make the system crash
- Can access the functions they need



Users achieve their goal in a straightforward way
Now how did this work? How did I get here?
“If want to do this (e.g. download a tool or a driver) you first have to fill out this form, it does not help you in any way, but we'd like the data, it just takes about an hour and we ask very detailed questions, but hey, we want to live!”
Ok, that's done so far, now where's that printer
they do not make errors
they do not make unintentional and frustrating detours

Program and
System Engineering
PSE

SIEMENS

Definition:
- Users
- Goals
- Effectivity
- Efficiency
- Satisfaction
- Context

Efficient achievement

Support Center Usability

- In whose family is there a VCR?
- Who programs it?
- A lot of relatives refuse to touch it?

Program and
System Engineering
PSE

SIEMENS

Definition:
- Users
- Goals
- Effectivity
- Efficiency
- Satisfaction
- Context

Emotion

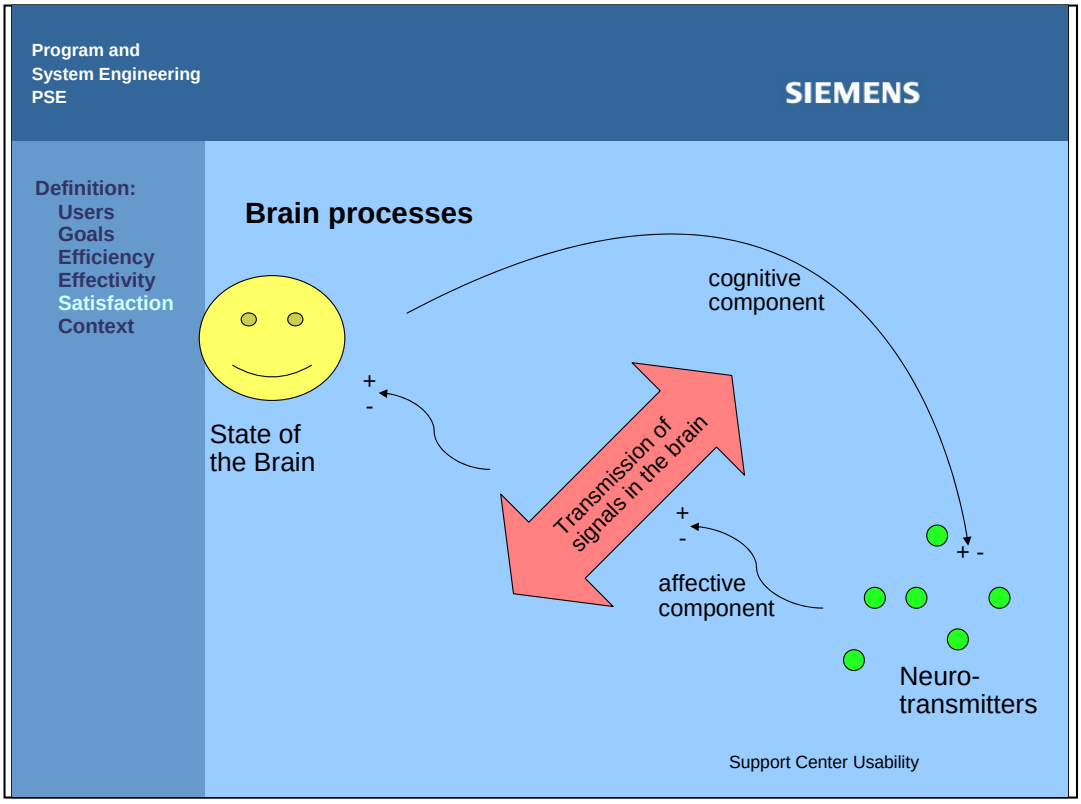
- A sign of high evolution
- Necessary for decisions
- Abbreviation of weighing process
- Influences the attractiveness of an object
- Influences perceived performance

Support Center Usability

- The higher up on the evolutionary tree an animal is, the broader its range of emotions
- certain brain injuries – people have no more emotions – can't even decide on what to wear or what to eat – there is no real value in those decisions, they are unable to do it, as ratio does not really help.

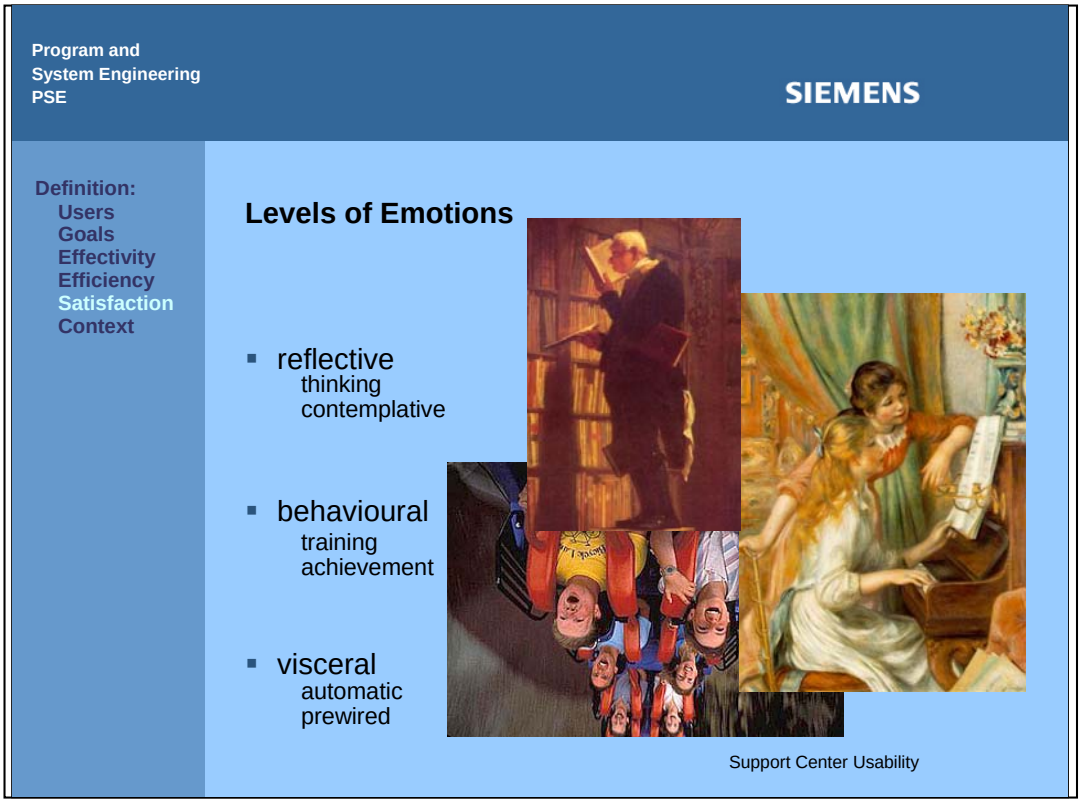
- Often decisions are done from a hunch, no careful thought and the reasons given afterward are just rationalizing that first hunch (they are absolutely valid and correct, but the decision went first).

- Influence on perceived performance is an important information regarding usability
- ATM test in Japan found out that persons likedr the performance of those ATMs better that had the more pleasing layout, a verifying study was performed in Israel (on the hypothesis that this finding stwould not be verified).



- A certain state of the brain affects the release of neurotransmitters
- Those enhance or inhibit the transmission of signals in the brain and thus influence the state of mind
- The circle can be entered either from the brain side – setting yourself consciously into a certain state of mind like listening to music you like in order to boost your spirits or reading a comic or watching a TV show
- It can be entered from the affective side – if you see something you don't like, like a spider or rat (or whatever your pet monster is) or getting a really bad fright because somebody jumped out at you from behind a corner.

- The affective and the cognitive component are always present, just in different ratios.
- The cognitive component assigns meaning to a thing
- The affective component assigns value to a thing

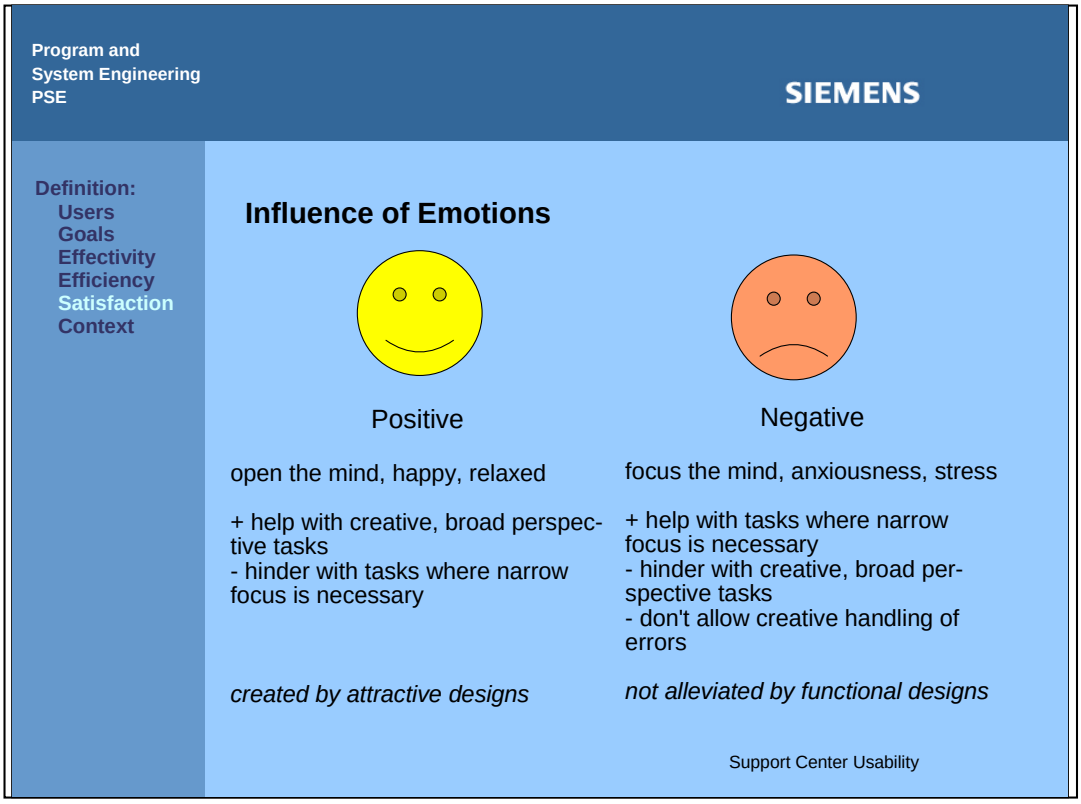


- **Visceral level** : automatic and prewired level, the level of gut reactions, the level of fixed routines, “realize-and-respond” behaviour – enjoyment may require other levels (roller coaster requires the knowledge that it’s really quite secure and that you are achieving something)

- **behavioural** level: the processes that control everyday behaviour
- level of learned and trained things. Behaviour can be analyzed and changed according to this analysis
- gives the pleasure of using a good tool effectively
- enjoyment requires accomplishment and skill

- **reflective level**:
- the level of thinking about things, of thinking about how to think
- enjoyment of music or art, initiates thought driven processes
- enjoyment requires study and interpretation

- usually all three are part of an emotion, though one may be the strongest component



- Positive emotions are necessary for creative tasks like brainstorming, finding alternatives, learning, research
- e.g brainstorming is best prepared by creating a happy and relaxed atmosphere, jokes, good feeling, that's why it's absolutely imperative that there is no criticism of ideas! Everything is welcome to keep the river of creativity flowing and not stopping it up.
- Use friendly coloring and attractive designs to make the users relaxed and get them thus to be more creative in finding solutions to problems the SW throws at them ;-)

- Negative emotions focus the mind, alternatives are much more difficult to find. In a stress situation the user is much more apt to repeat an action, even if it was not successful before. The user is not able to think of alternatives. Good for error handling, fault handling, catastrophe relieve if the procedures were fixed before.
- Extreme cases are panics induced by fires. If the doors ope the wrong way people will NOT be able to open them! They will perish!
- If they run down emergency staircases and the design is not guiding the flow naturally outside they may run down into the basement and become trapped there. Again, deaths were caused by such designs.
- Those are extreme cases, but keep in mind that in situations where the user has to stay focused cute colors or bunnies or cool anime icons are distracting and make the user lose focus. e.g. In a control room of a nuclear power plant...

- **CONCLUSION**

Definition:
- Users
- Goals
- Effectivity
- Efficiency
- Satisfaction
- Context

Influence of Emotions

Positive

Negative

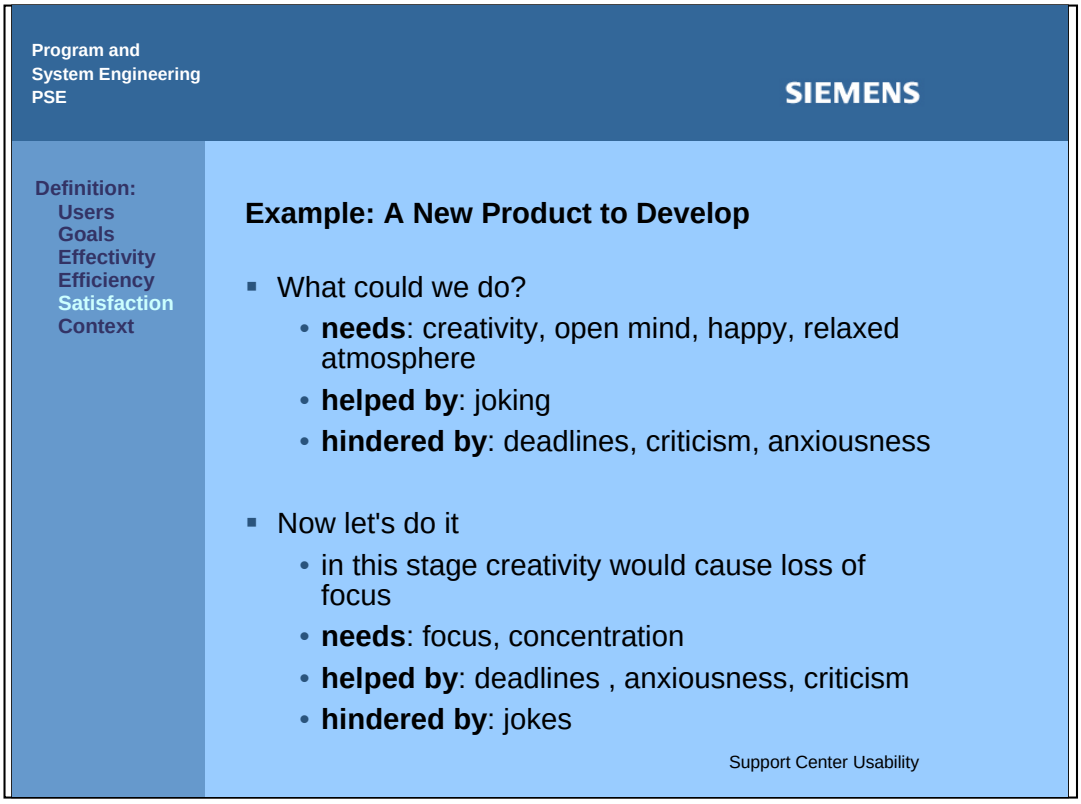
+ help with creative, broad perspective tasks
- hinder with tasks where narrow focus is necessary

created by attractive designs

- hinder with creative, broad perspective tasks
- don't allow creative handling of errors

not alleviated by functional designs

Don't make people under stress think !



- Of course, deadlines help only if they are realistic
- all negative emotions only help as long as they are not overdone and cause too narrow a focus or even panic
- This is a general statement only. Of course, in some debugging sessions a relaxed atmosphere may help finding the error, in others focus is the clue.

The context of use also is of great importance for the usability of a product.

Is the user going to be in a relaxed atmosphere, downloading music?

Is the user going to be in a manufacturing environment, a very high noise level, so that sound feedback cannot be heard, very hot, very cold (cannot move the fingers very well or cannot hold a pen due to wearing gloves), there's lot's of dust in the air – or strong magnetic fields that would influende a CRT monitor

Is the user going to use this application while moving around on foot or going by some vehicle or other. Using a pointing device on the PDA while walking is quite a challenge, Is the connection apt to be often interrupted (in a train)

Is the user going to use this application somewhere in the home, where it has to be unobtrusive, but always available? Regarding fridge surfaces as shown in this last picture. How many people, do you think are going to have space available to use the fridge door for a display?

Consider buying a ticket for the train. In a train station hall, many people around you, probably quite a noise level, probably under stress to reach the train you need to catch. You probably won't be very tolerant of the interface. You just want a ticket from Brno to Praha and back and of course a discount because of your student card. A default setting of travelling first class is probably inappropriate, an interface asking you how many kilometers you wish to travel, neither.

Conclusion

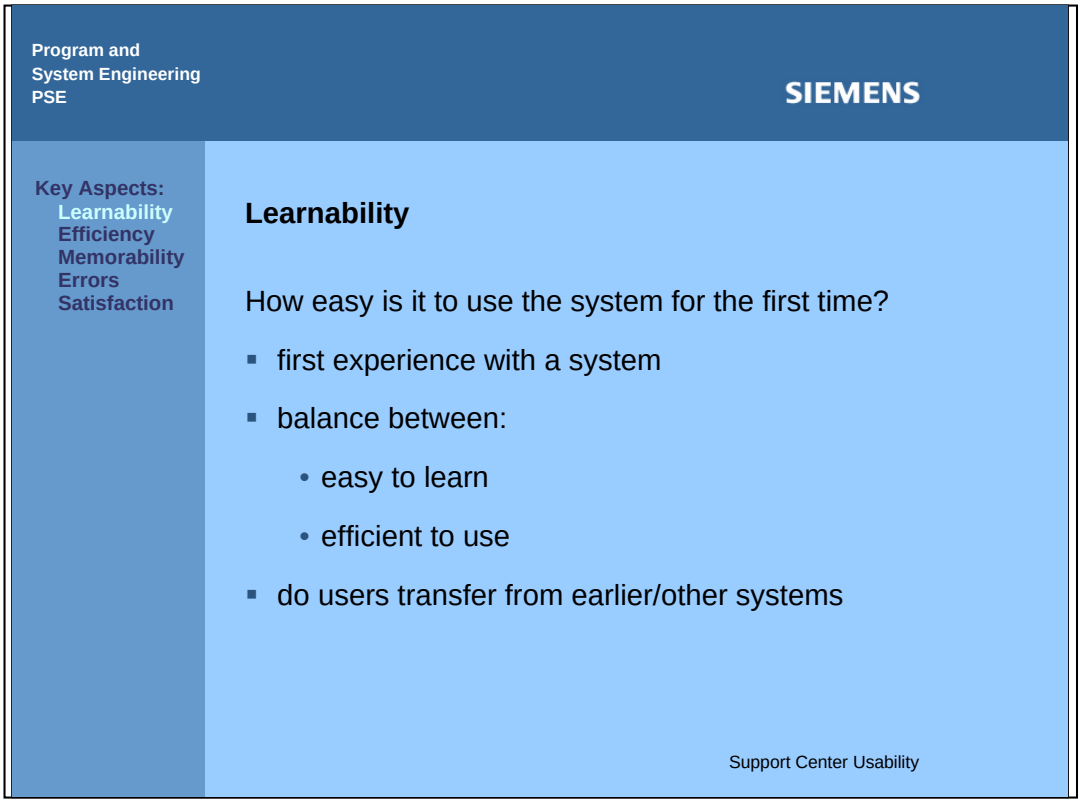
The important questions are:

- Who are the users?
- What do they want to do?
- What are their conditions?
- What are the circumstances?

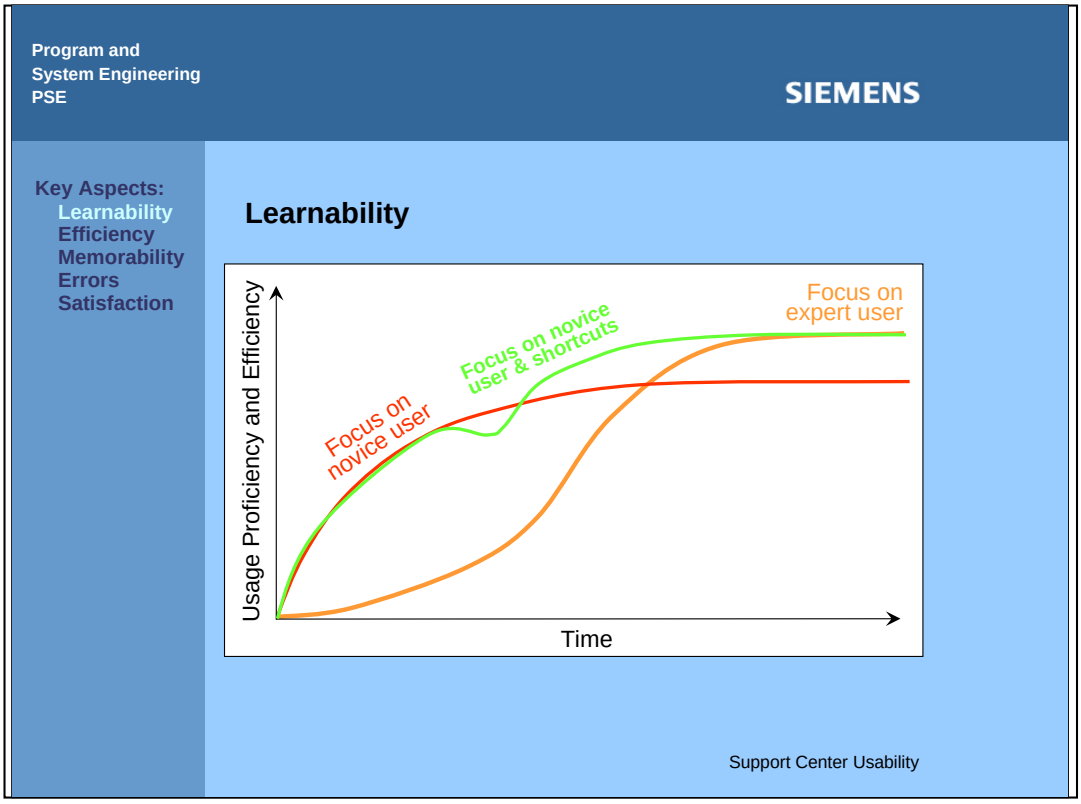
Key Aspects

Usability is not a one-dimensional property of a product, but multi-dimensional, e.g.:

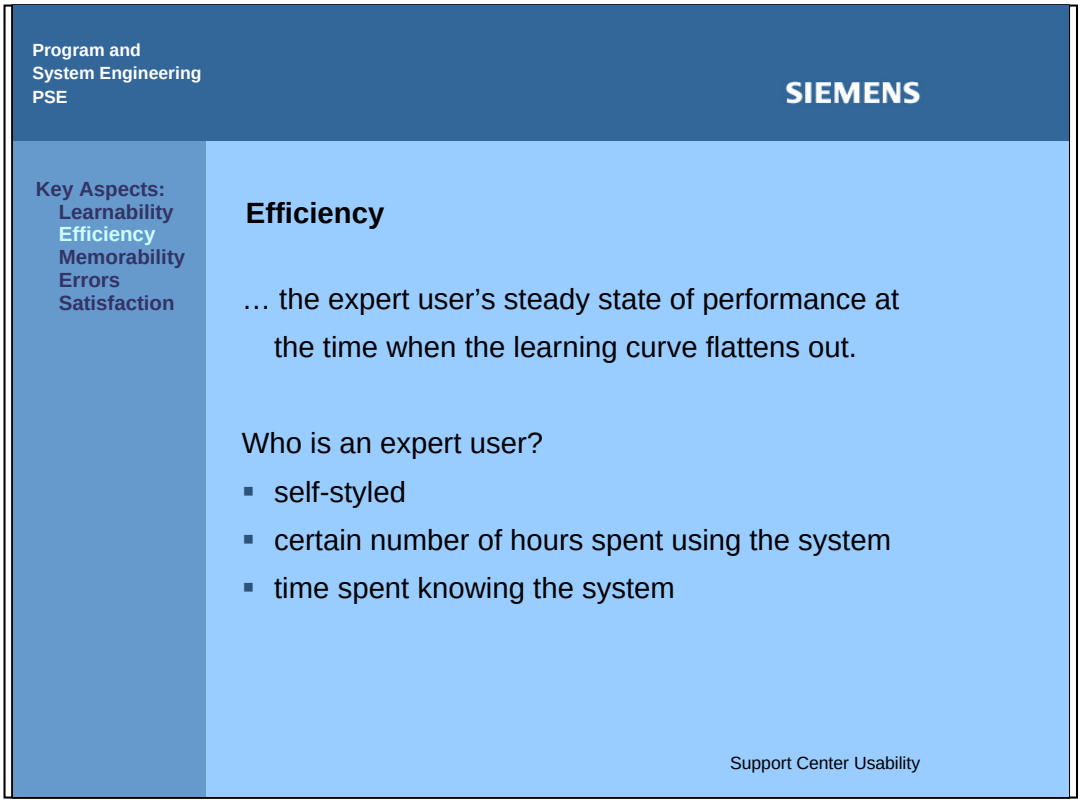
- Learnability – easy to learn
- Efficiency – efficient to use
- Memorability – easy to remember
- Errors – few
- Satisfaction – subjectively pleasing



most fundamental usability attribute, as it lays the basis for further interaction
e.g. Windows assitant
help for real beginners, but very annoying and
blocking for more expert users



- novice user: quick to learn, but not very high efficiency
- expert user: more difficult to learn, higher leovel of efficiency
- focus on novice users, offer shortcuts, etc. as abbreviations for experts -> maybe a small dip when learning the expert mode, but higher level of efficiency reachable



•a system that focuses on the novice user is going to be easy to learn, but less efficient to use

•a system focusing on the expert user is more difficult to learn but highly efficient for the expert user.

•know your user: Are the users going to walk up to the system, use it and go away (ticket machine – ease of use) or are they going to work with the tool day in and day out (efficiency)

•Now that's quite a dilemma, isn't it? Fortunately there are mechanisms allowing the combination of the two advantages. Design a system for novice users, but include shortcuts to certain commands.

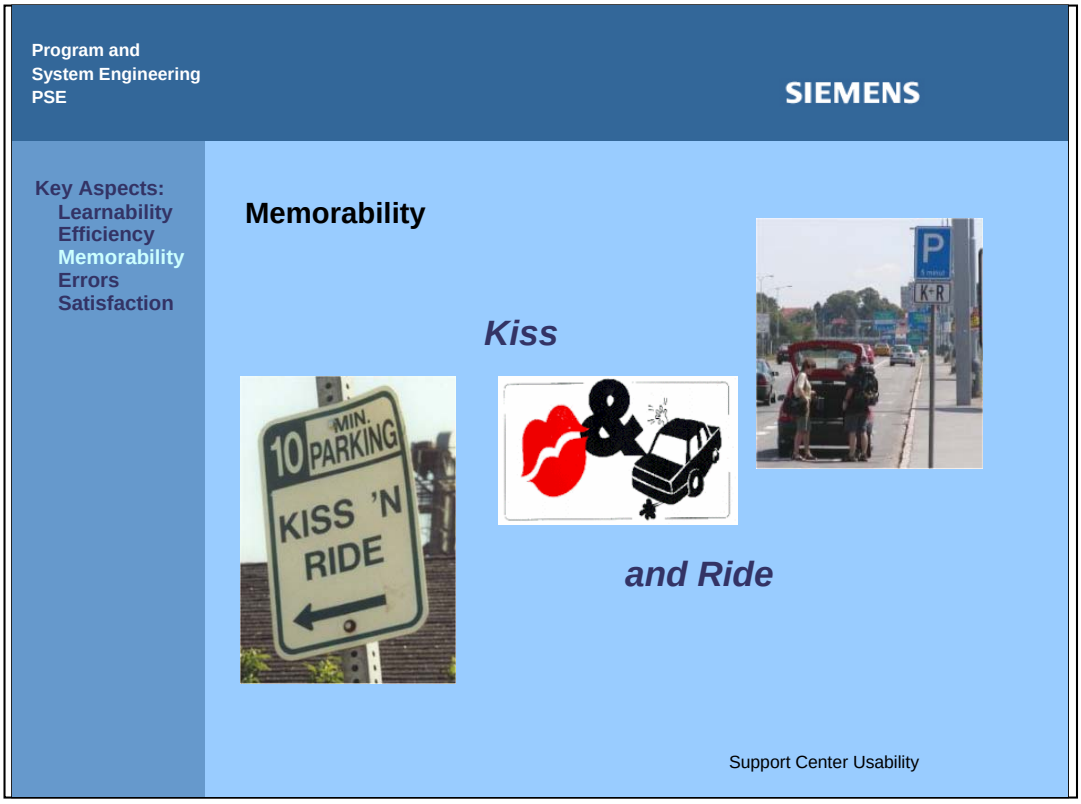
•the learning curve will be something like the green one in the previous slide: with a small dip where the near-expert is starting to use the shortcuts and other abbreviations

Key Aspects:
- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

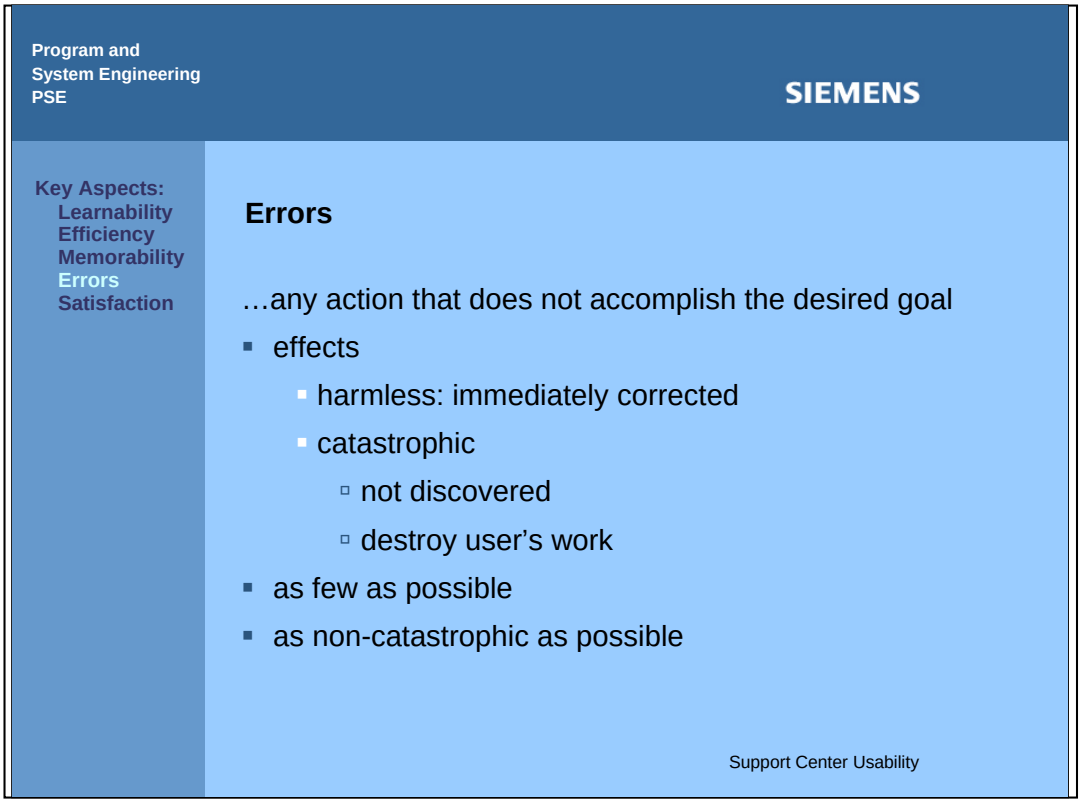
Memorability

How easy is it to use the system based on previous learning?

- important for the casual user
- uses the system once per quarter or per month
- does not need to learn from scratch again



- a concept that has to be explained, it is not self-explanatory, but afterwards it can be remembered very easily.
- Kiss & ride: a zone for persons dropping of others at schools or train stations or airports



•anything from hitting the wrong button to not hitting the desired menu item to crashing the system by pressing the button again too soon.

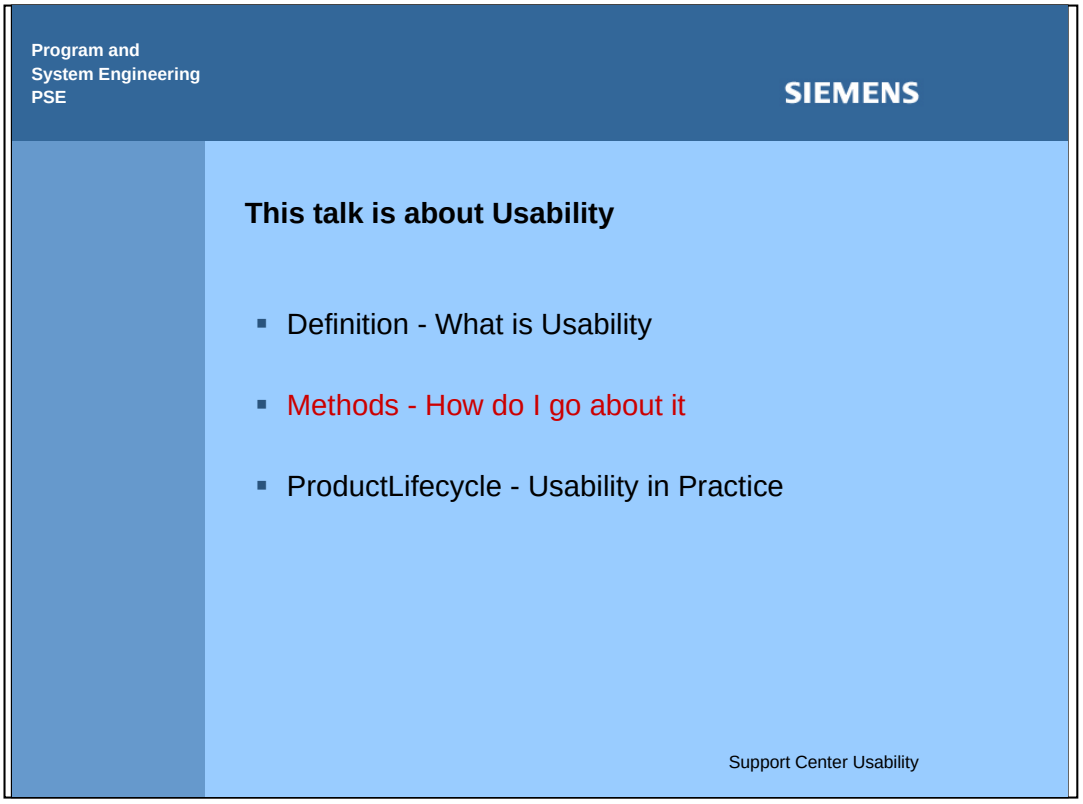
•Certain approaches to designing an interface for few errors will follow.

Key Aspects:
- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

Satisfaction

How pleasant/satisfying is it to use the system?

- different from the overall attitude towards computers
- perceived high degree of control increases satisfaction
- highly subjective
- if asked directly, people are usually very polite about an interface

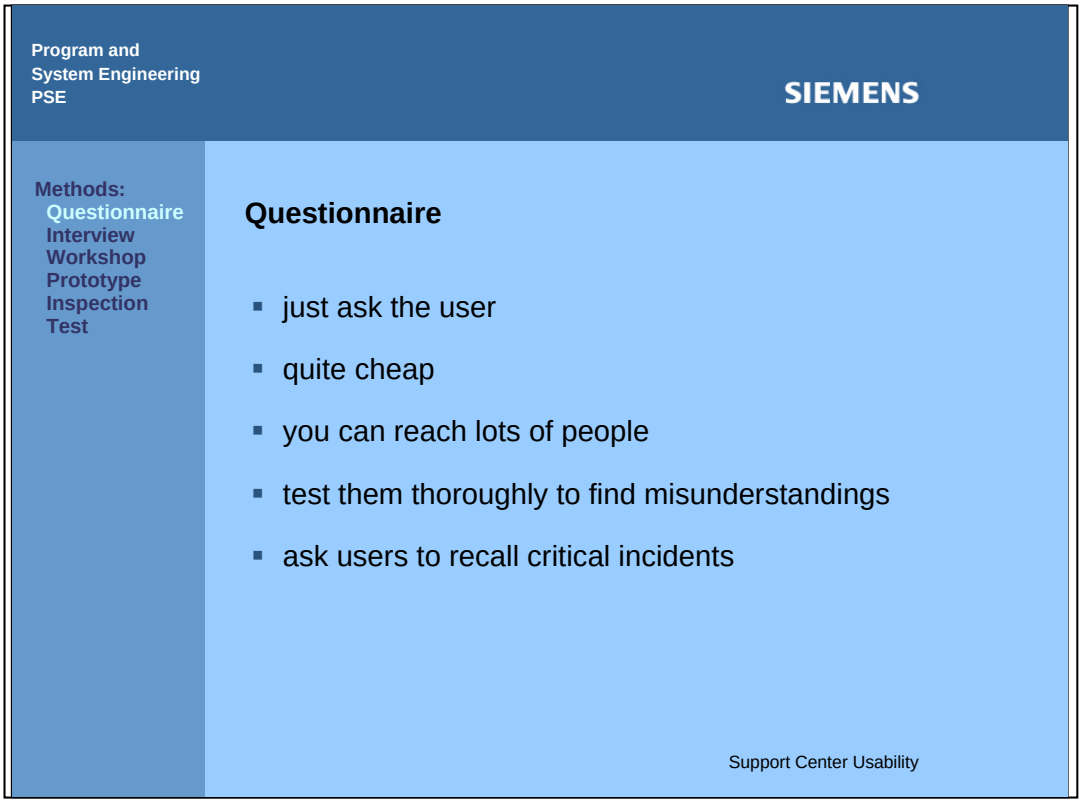


- Now let's talk about the methods employed to ensure usability

Methods

Available are

- questionnaires
- interviews
- workshops
- prototypes
- inspection
- test



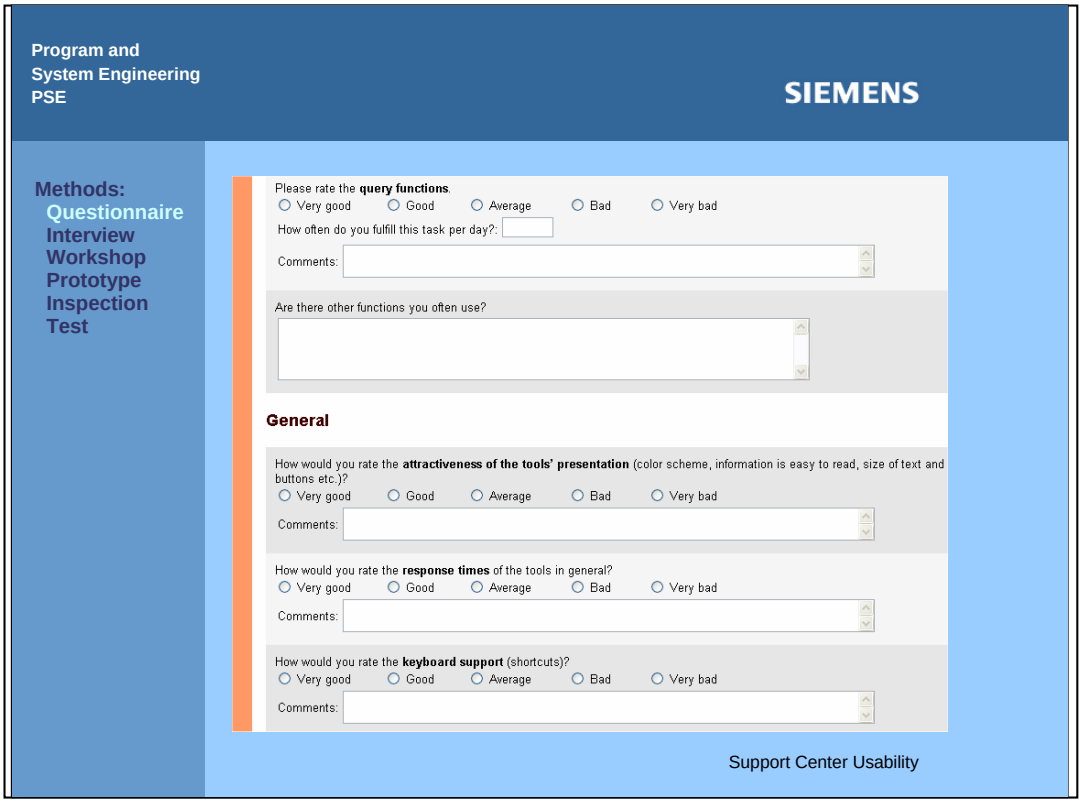
- Asking the user has to be done with a little bit of skepticism, as the user does not always know what is going to work or how he or she is going to do something.
- In the 50s a study was done regarding telephone handsets. They were quite heavy then and people were asked, if they would like them to be lighter. The result was no, people were happy with the handsets the way they were. Still, a test of handsets that looked identical but had different weights showed that people preferred handsets with about half the then-normal weight.
- This is just an example to illustrate that what the user says he does or does like is not always what he really does or does like.
- E.g. manual use. If you ask somebody what he'll do when encountering the problem, quite a lot will say: I look at the manual --- which, when you look at it has quite an awesome layer of dust on it.
- Just wait until the next problem arises and you will see how that person gets up and goes into the next room to ask whoever is the local guru.

- Users tend to be very polite when asked to rate an interface, so allow for that.

Questionnaire

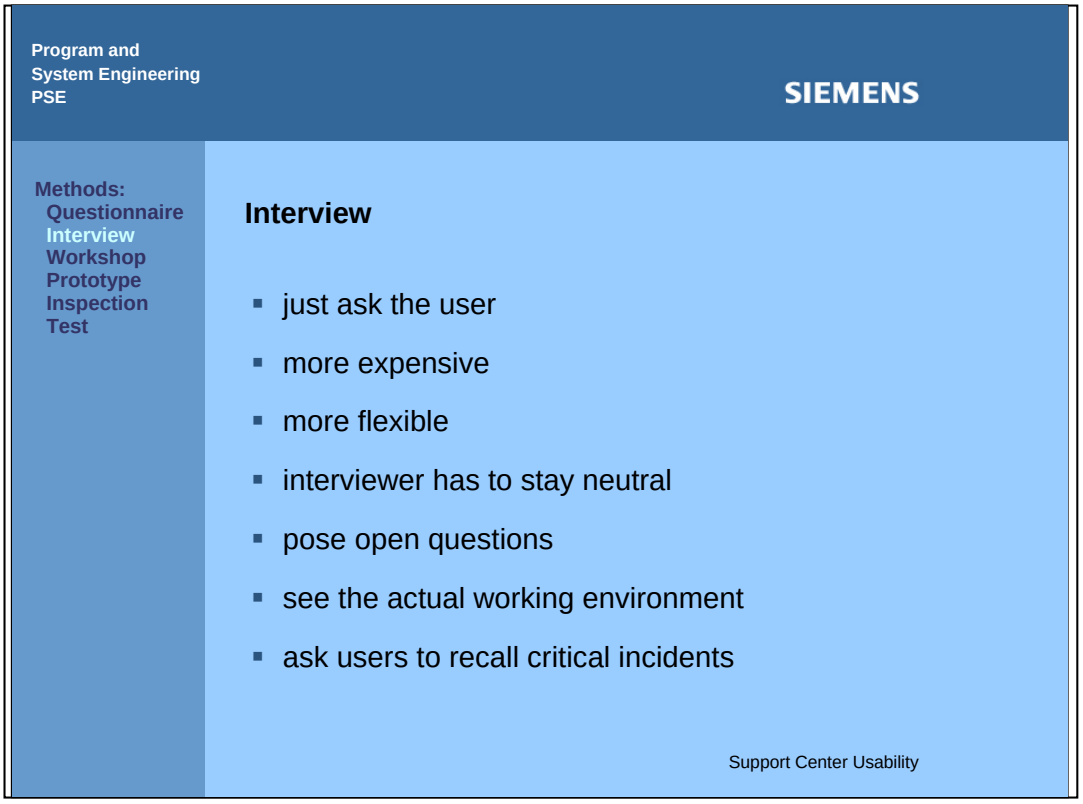
for:

- getting most annoying and most pleasing aspects
- reach lots of users
- get new ideas



- When did you last do this may be better than how often do you

- free text is sometimes tricky, people thend to leave it blank, you can't have it analysed by machines – whether this is a problem depends on the return rate of the questionnaire



- a lot of post-its around the screen may tell you that the memory load on this person is quite high. If everybody has a lot of post-its, there's probably something to support here.
- Emphasize that you have no stake in the product, that you are interested in the other's opinions and experience
- concrete incidents either positive or negative

Interview

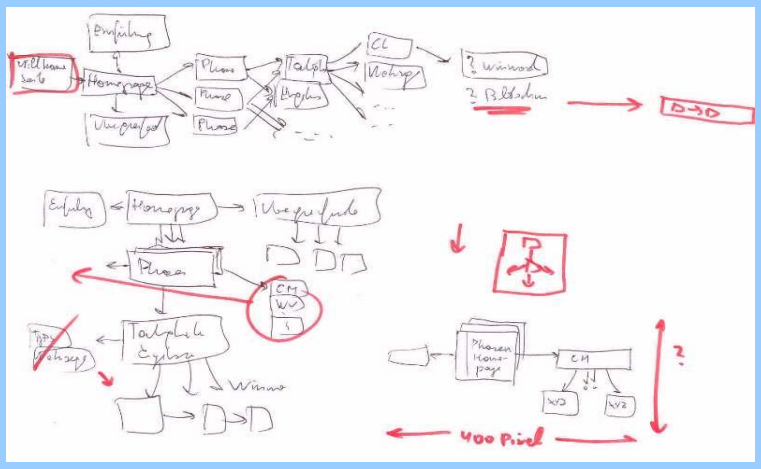
for:

- getting most annoying and most pleasing aspects
- asking a little deeper
- get new ideas

- apart from the experts, get those people that make the decisions!

Methods:
- Questionnaire
- Interview
- Workshop
- Prototype
- Inspection
- Test

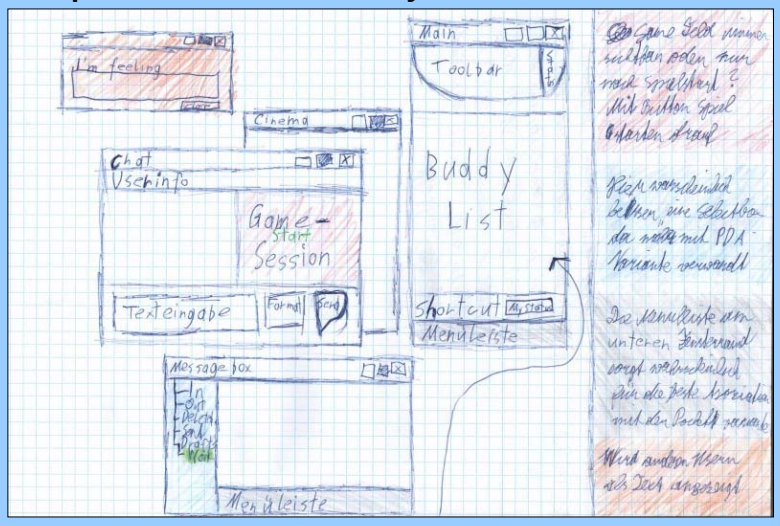
Output – ideas for menu structure



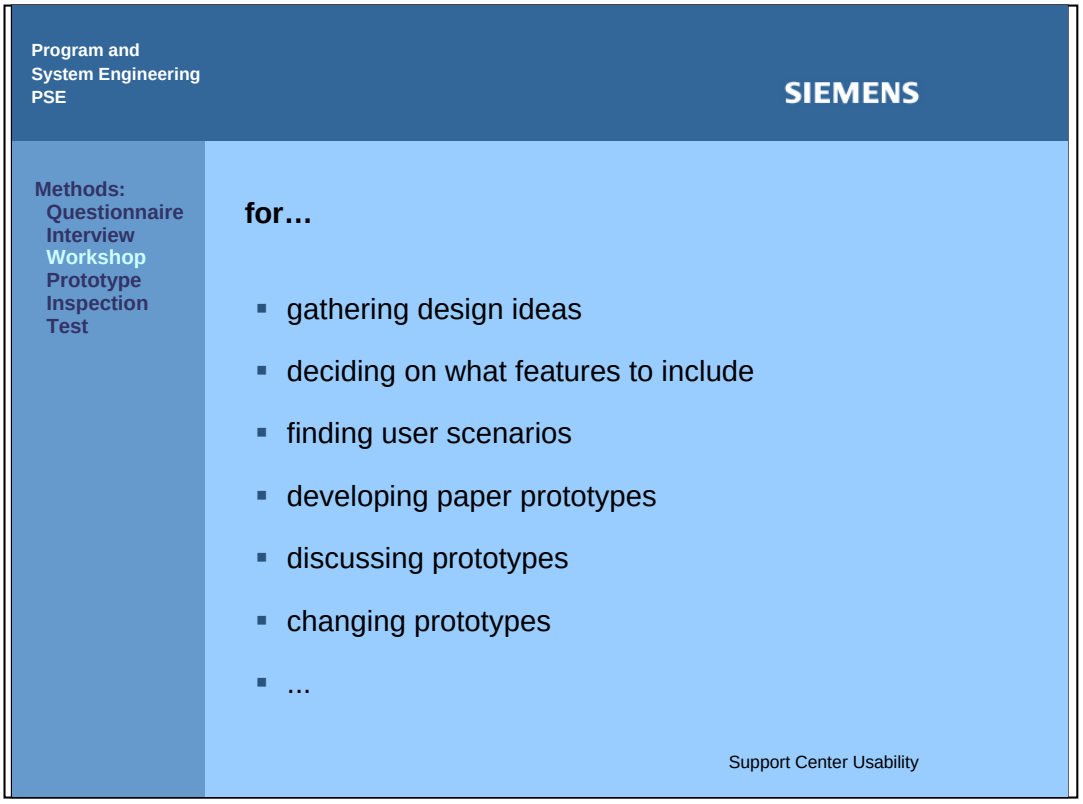
Support Center Usability

Methods:
- Questionnaire
- Interview
- Workshop
- Prototype
- Inspection
- Test

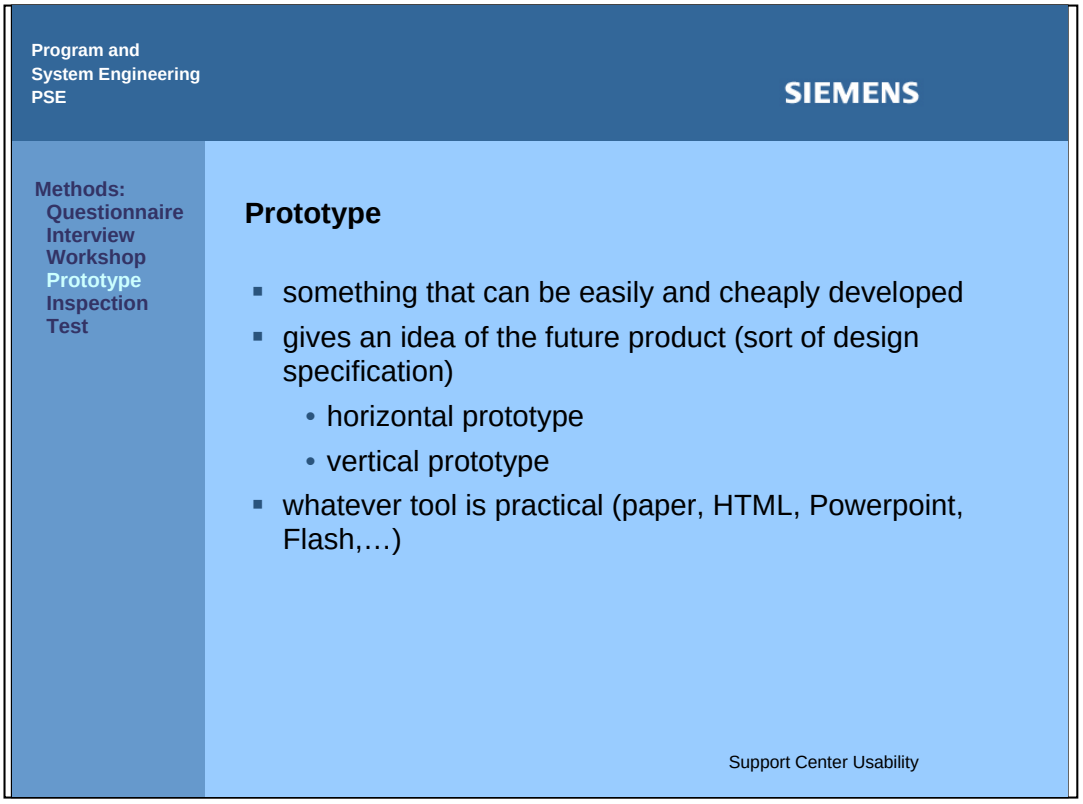
Output – ideas for screen layout



Support Center Usability



- at the end of the day: for whatever requires people to come together and work together to produce some output



- horizontal: shows all functionality, but (nearly) nothing will really work, a sort of Potemkin village: shows what the workflows are and what the look and feel is of the application
- vertical: take one function and implement it down to the database access etc.: gives an idea of how the different systems interact

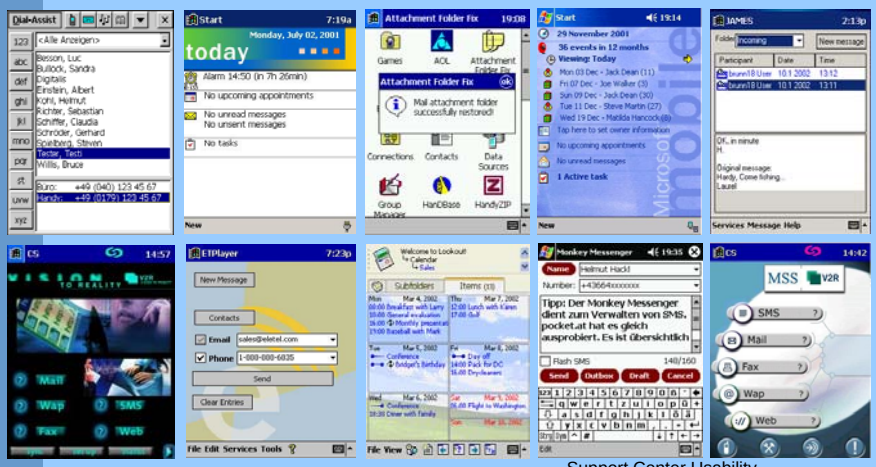
Program and
System Engineering
PSE

SIEMENS

Methods:
- Questionnaire
- Interview
- Workshop
- Prototype
- Inspection
- Test

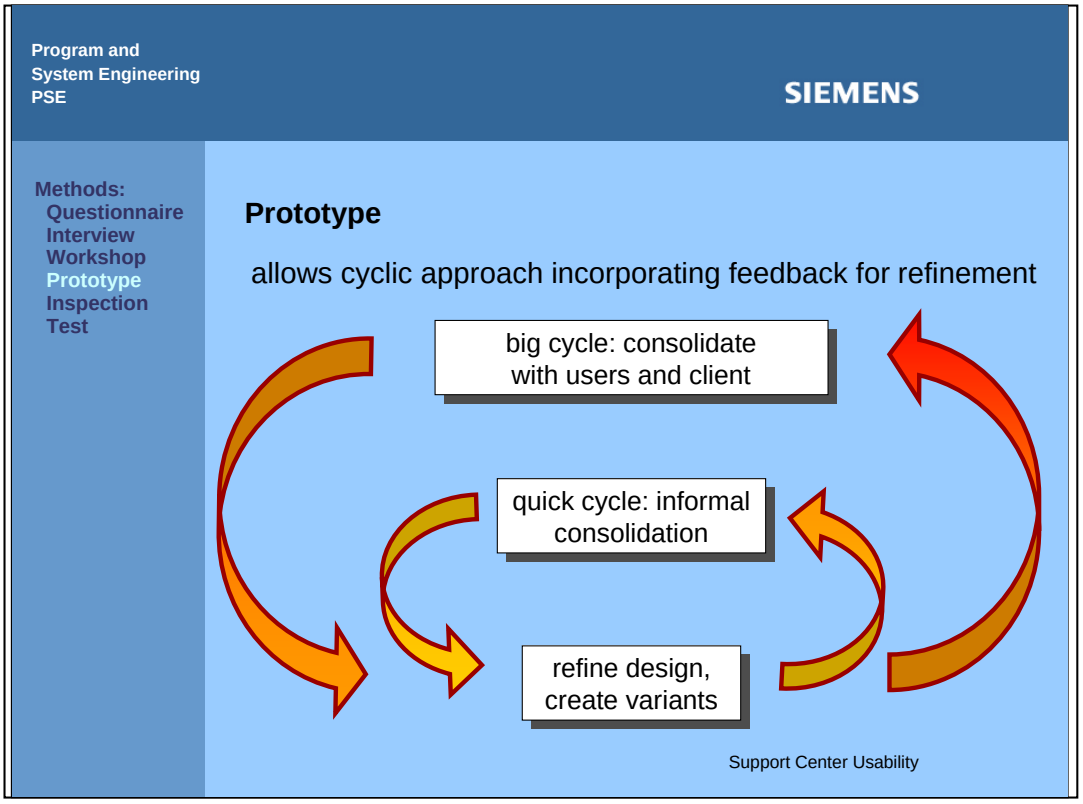
Prototype

different design prototypes for an application for PDA



Support Center Usability

- a picture says more than thousand words.
- instead of describing an interface, show a picture of it



- not only helpful, but necessary to gain good usability, as the design is being constantly refined with each new cycle

- The methods up to here, i.e. questionnaire, interview, workshop and prototype are the creative methods available to usability engineers.
- Now we will take a look at the last two methods, which are tools for evaluation.

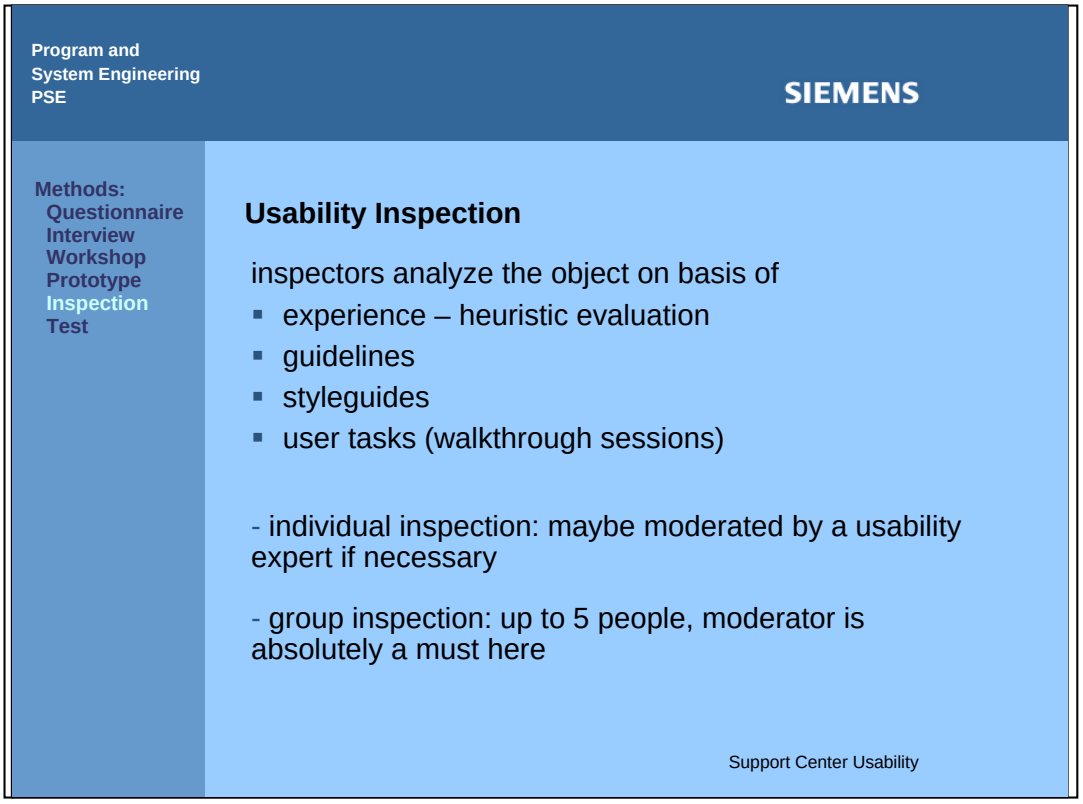
Methods:
Questionnaire
Interview
Workshop
Prototype
Inspection
Test

Usability Inspection

… is summary term for certain methods for evaluation and assessment of user interfaces

- object
  - paper prototype
  - prototype
  - interface
  - …
- inspector
  - usability expert
  - domain expert
  - end users
  - SW engineer
  - …

Support Center Usability



- individual inspection is recommended, as it assures, that each expert is heard, nobody is “drowned out” by somebody else

Program and
System Engineering
PSE

SIEMENS

Methods:
- Questionnaire
- Interview
- Workshop
- Prototype
- Inspection
- Test

Usability Inspection

"What happened here?"
"I can't find it"
"It's not comfortable like this"

Support Center Usability

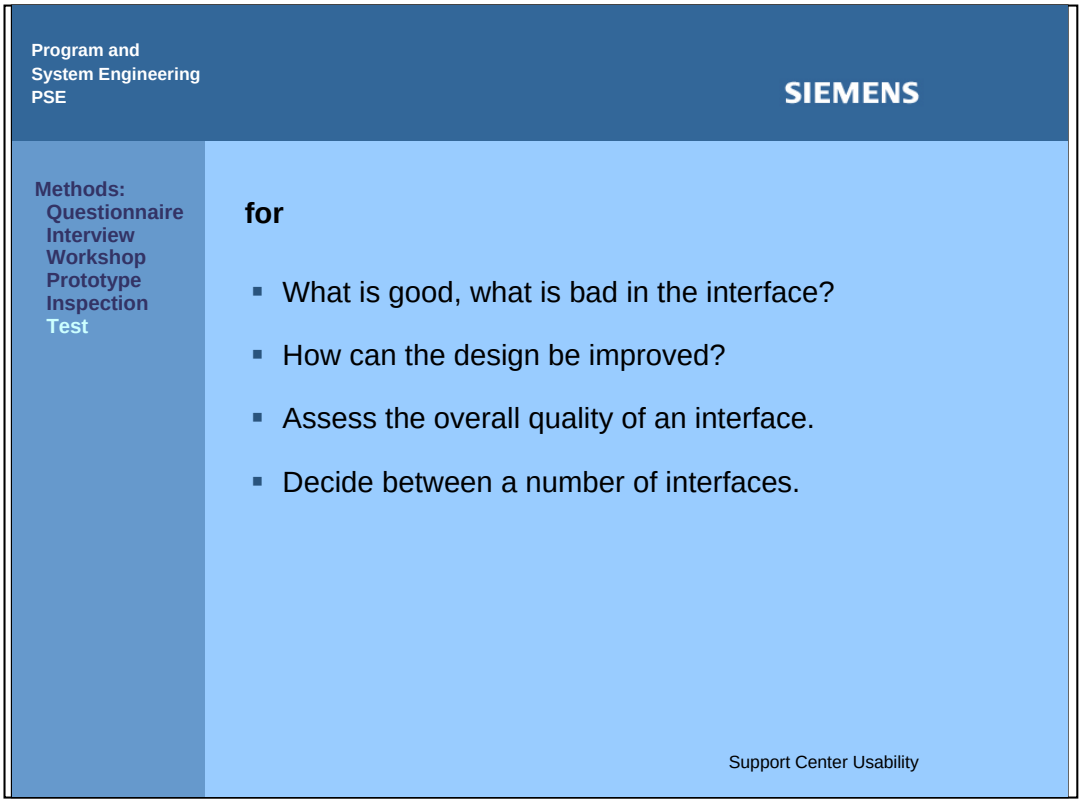
- comments are valuable hints at some underlying problem
- What happened here: surprise, the system acted other than the user expected
- I can‘t find it: What is needed is not accessible, it is not found
- It‘s not comfortabel like this: ask deeper: what is missing, or is there too much on the screen? What do you perceive as uncomfortable?

Usability Test

... is an empirical test procedure.

- you need real users
- you need real tasks
- formal method
- most objective method
- record sessions for later evaluation

Support Center Usability



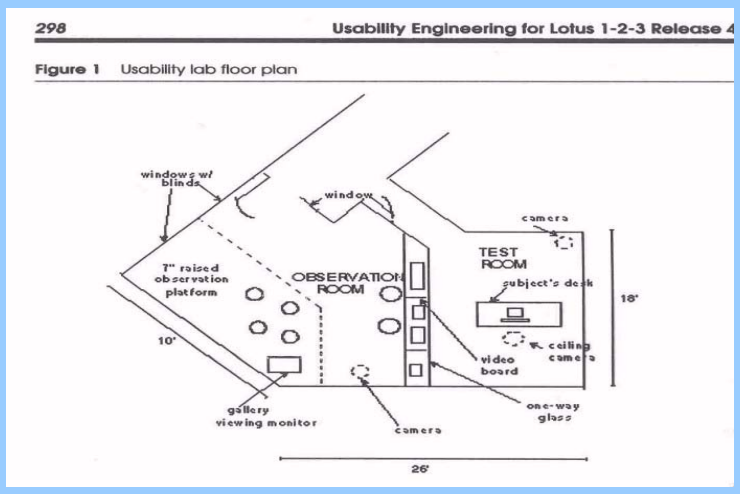
- A usability test is a very formal method. There are rules to follow in order to gain its one big advantage: you get the most objective data from usability tests. The user acts quite undisturbed (as far as that is possible in a laboratory situation) and you can really see where the hindrances and blocking points are located.
- The test also offers the possibility of measuring the performance of users in order to determine whether certain goals for the SW have been met, like
- 90% of the users should be able to complete this task within 5 minutes.
- 90% of the users should find the given information withinn 30 seconds“ and so on.

Program and
System Engineering
PSE

SIEMENS

Methods:
- Questionnaire
- Interview
- Workshop
- Prototype
- Inspection
- Test

Usability Lab



Support Center Usability

- show the user what is behind the mirror, so s/he need not wonder what is behind it and can concentrate on the tasks
- observation from behind the mirror in order not to disturb the user
- maybe an experimentor in the room with the user to help in emergencies

Program and
System Engineering
PSE

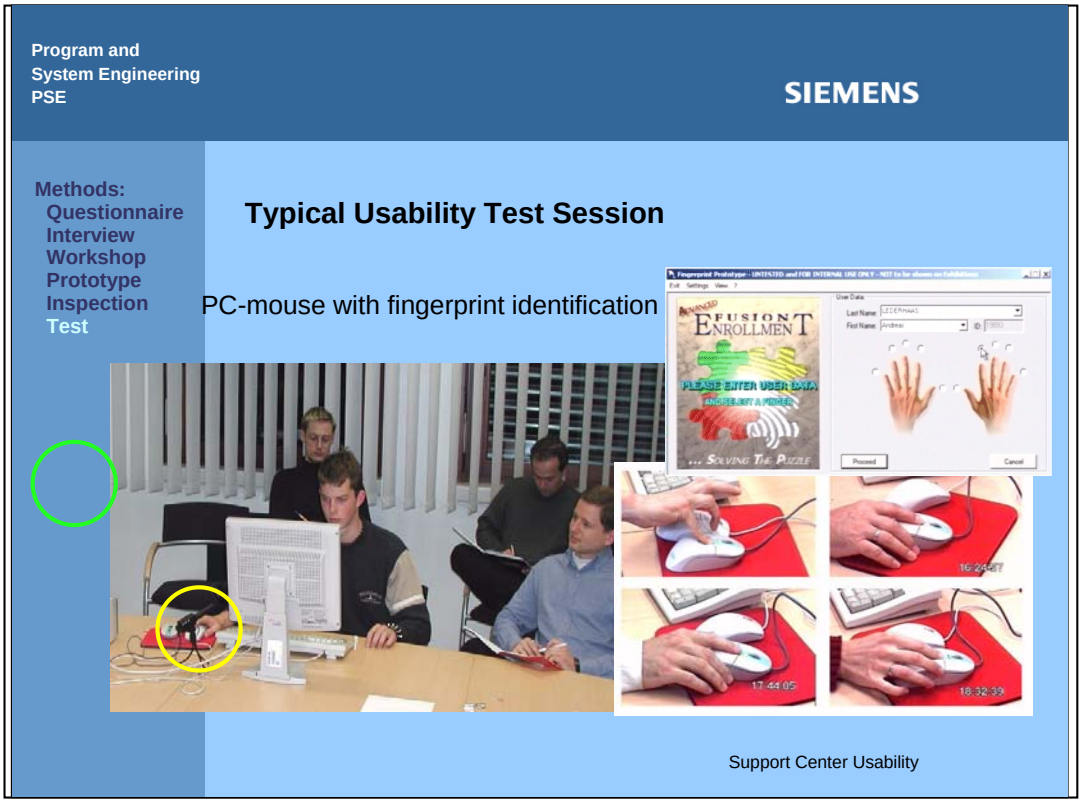
SIEMENS

Methods:
- Questionnaire
- Interview
- Workshop
- Prototype
- Inspection
- Test

Usability Lab (portable)

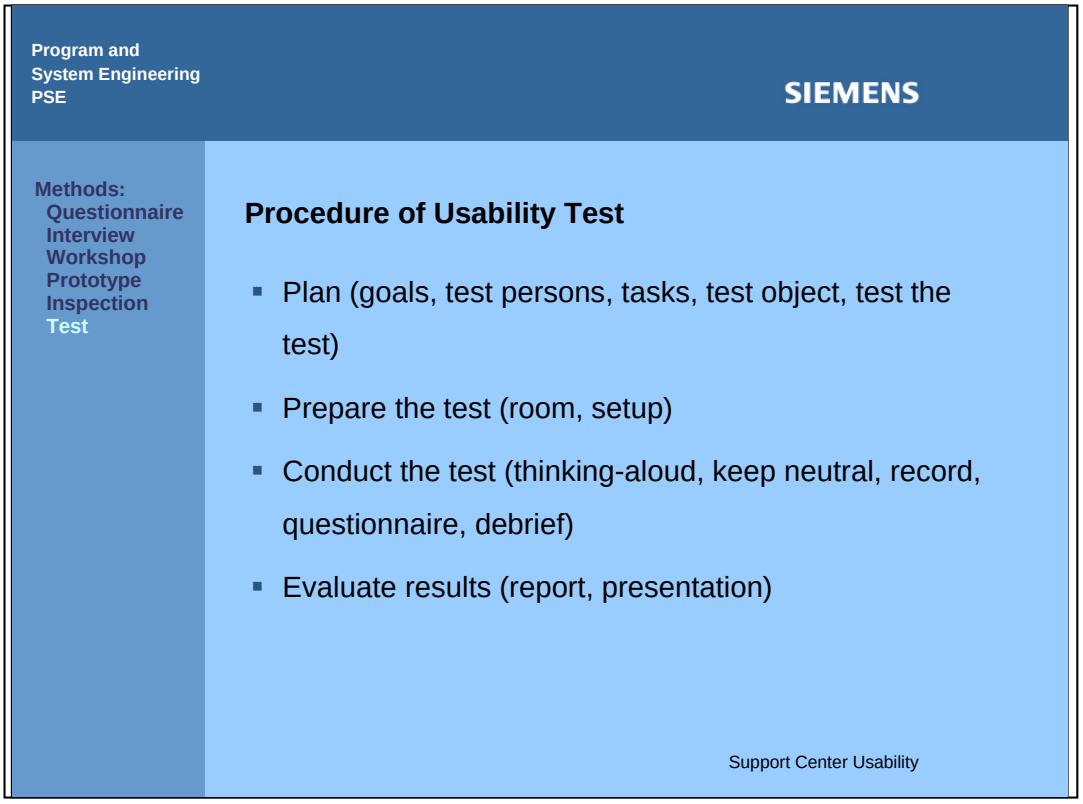
Support Center Usability

- As you can see, full usability labs are quite sophisticated things.
- the good news: most times you don’t need such a lab

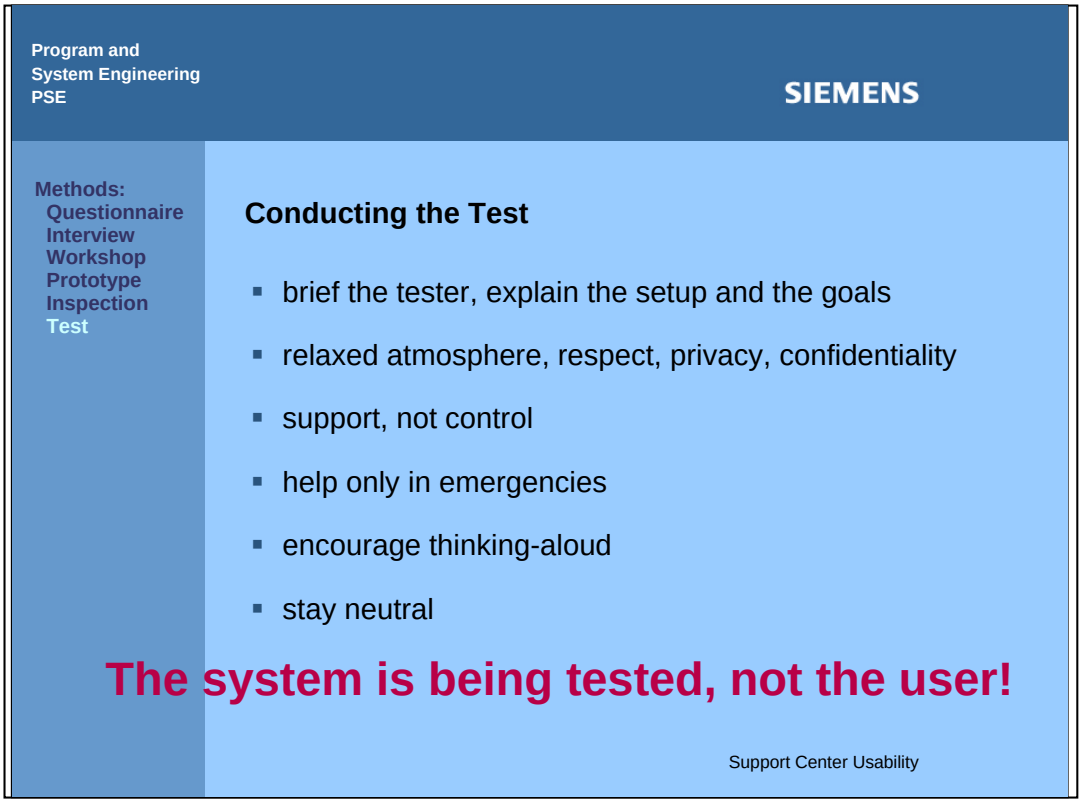


- This is a fairly typical usability test session. In this case an ID-mouse was tested.
- The lab is very portable, consisting of a microphone (in the yellow circle) and a camera (imagine one about where the green circle is) focused on the mouse. In this session also a screencam was used, as the interesting events happened on the screen and the mouse.

- The user gets a set of tasks. The usability expert is there to take notes, to help in situations that get out of hand and to ask questions if the users stops the running commentary.
- The two people in the background were developers, also taking notes of the problems that were found.
- The developers may watch, but **NOT** interfere (which can be quite hard).
- It is best if the person accompanying the test is neutral regarding the product, so the examinator should not be part of the development team.



- in the planning phase, define the goals, find out how many test persons (usually 5-6 of each user group) you need, what will the test object be (papaer prototype or full blown application), set the tasks to perform, test the test in order to remove misunderstandings
- set up the room and the system for the test (maybe prepare soft drinks, etc)
- during conducting of the test encourage the tester to do thinking-aloud, record the session, keep neutral regarding the prodcut. After the test session have the tester fill out a questionnaire regariding the product and only then debrief the tester and discuss things with him/her. The discussionmay influence the opinion of the tester, that‘s why it should be done after the questionnaire.
- do a report for each test (makes it easier to find the information lateron) and an overall report and maybe a presentation regarding the test for the customer/project.

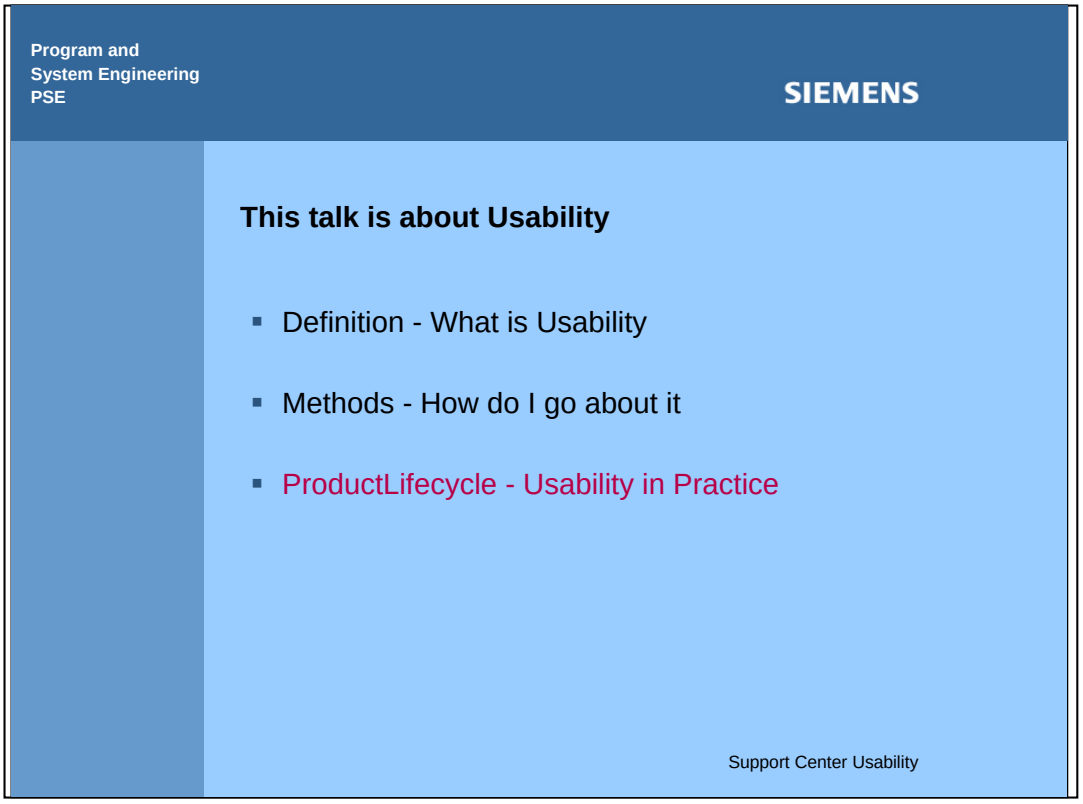


- depending on the product being tested, the tester may put a certain amount in trust into you. e.g. if it is some product that is being used at work. Breaking the trust the testers put into you by giving their managers detailed data about their performance during the tests has bad consequences for future tests, not talking about the bad ethics involved here. The data gathered here is also not representative of the usual performance of the user, as the goals the tester is given are different from the ones enforced at work (the system is being tested, not the user)

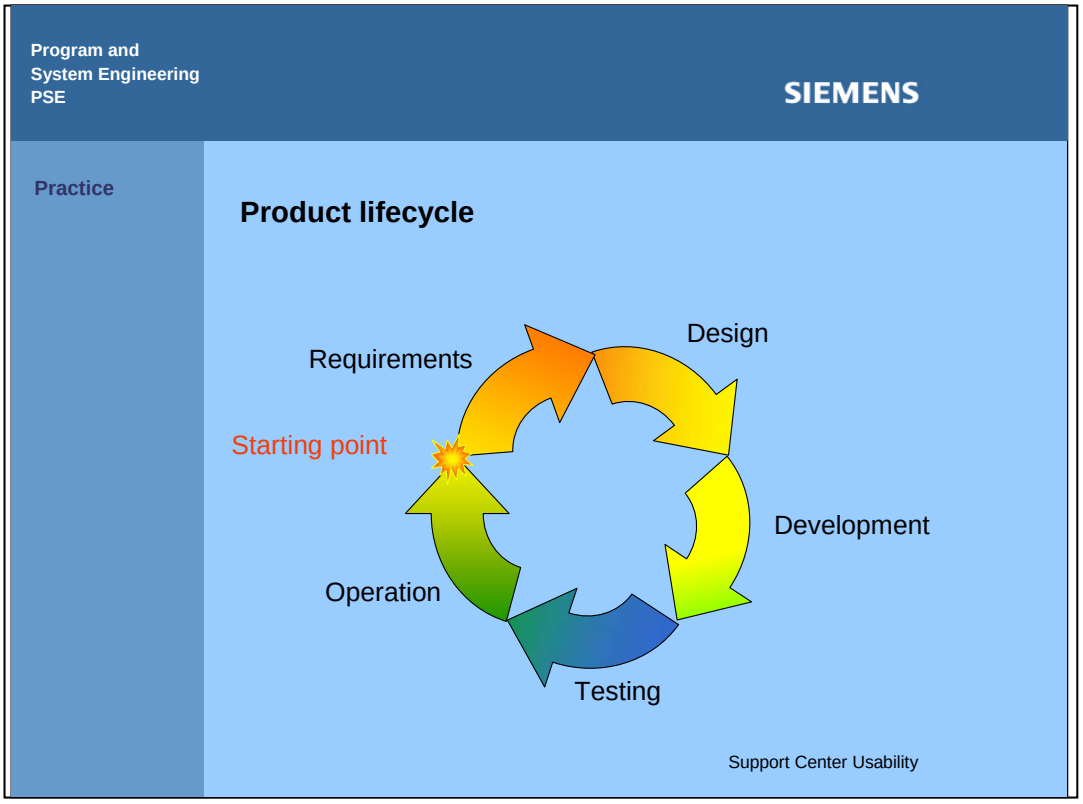
Methods: Overview

Method	Type	pro	con	use for
question-naire	creative	lots of people	not much depth	overall picture
interview	creative	very individual	takes time	workflow, details
workshop	creative	all experts together	maybe too many opinions	composing, defining
prototype	creative	try something out cheaply and quickly		refining, testing
usability inspection	validation	opinion of an expert, quick	Do not forget users!	quick testing
usability test	validation	valid data	formal, takes effort	formal testing

Support Center Usability

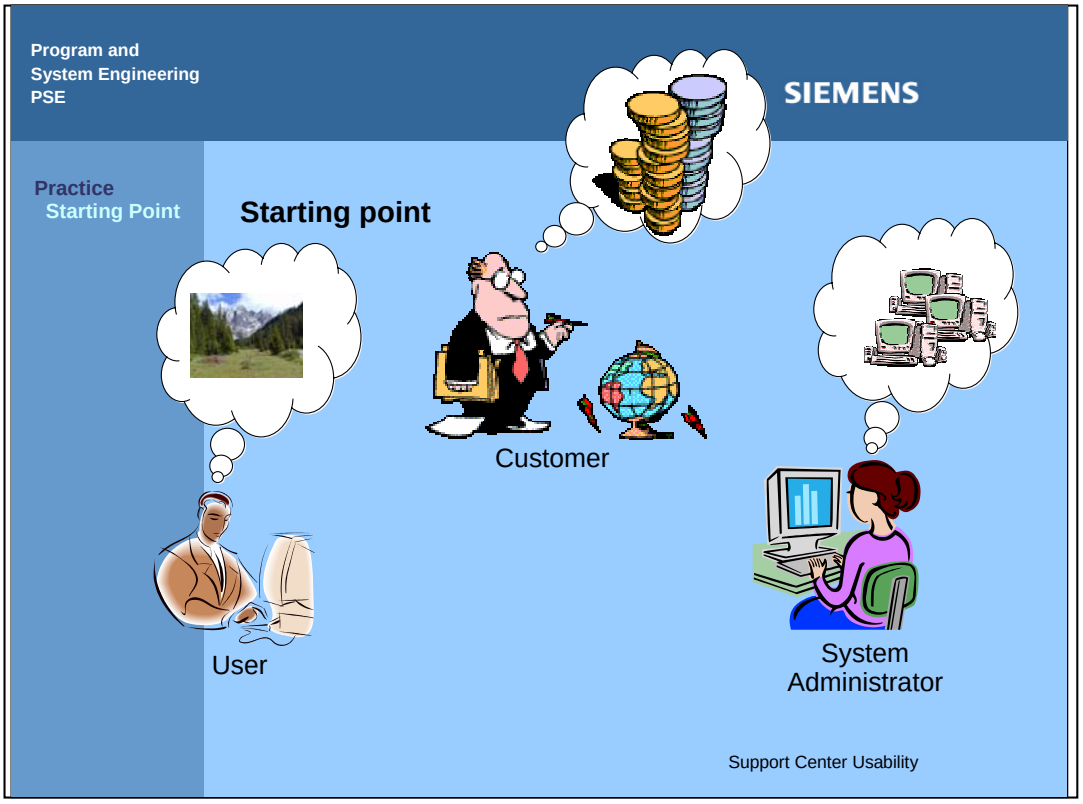


- We now know what usability is about and which methods we have at our disposal to evaluate and thus ensure it.
- Now let's have look at the practical side of it.
- Where can we use which methods and how do we go about it in the real project-life jungle out there

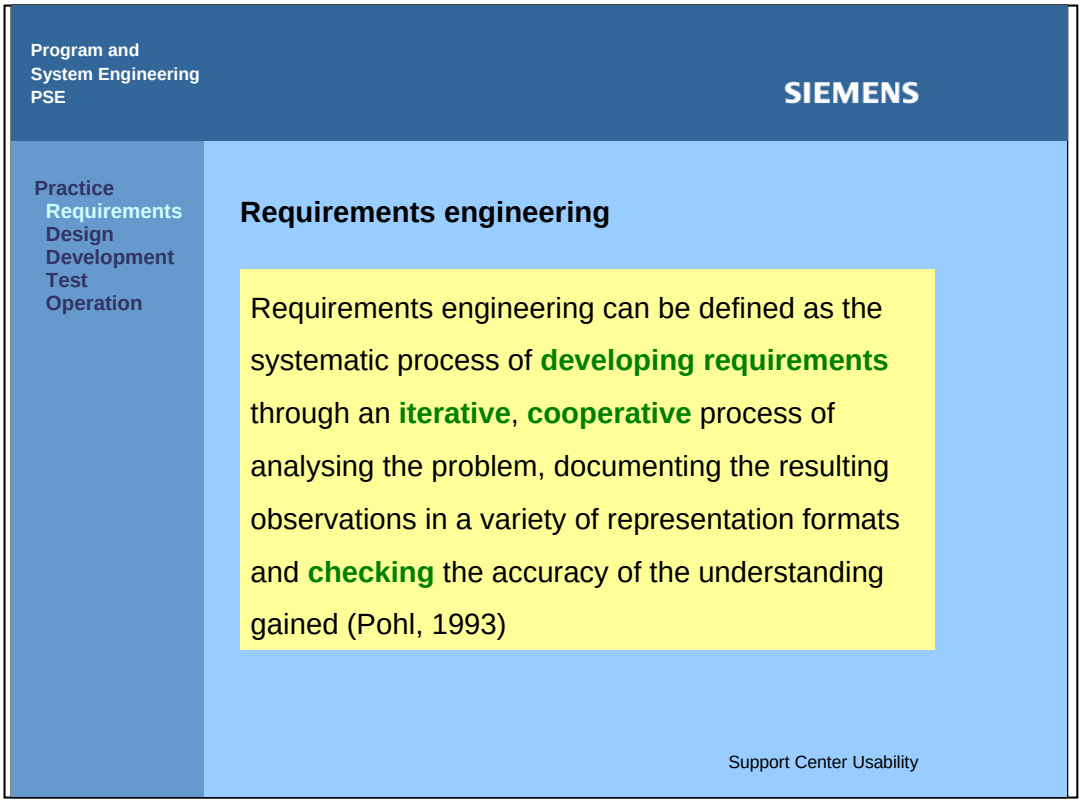


- This product lifecycle is not very detailed, but it shows the steps on the way of an idea becoming a product available on the market.
- In the requirements phase the idea is refined in such a way a viable product can be designed and built. It is a very important phase, as what comes out at the end is defined here. Errors made, but not discovered here usually continue on through all the rest of the development of the product until:
  - it is not bought because it does the wrong things,
  - does not do the right things or
  - does the right things in a wrong way.

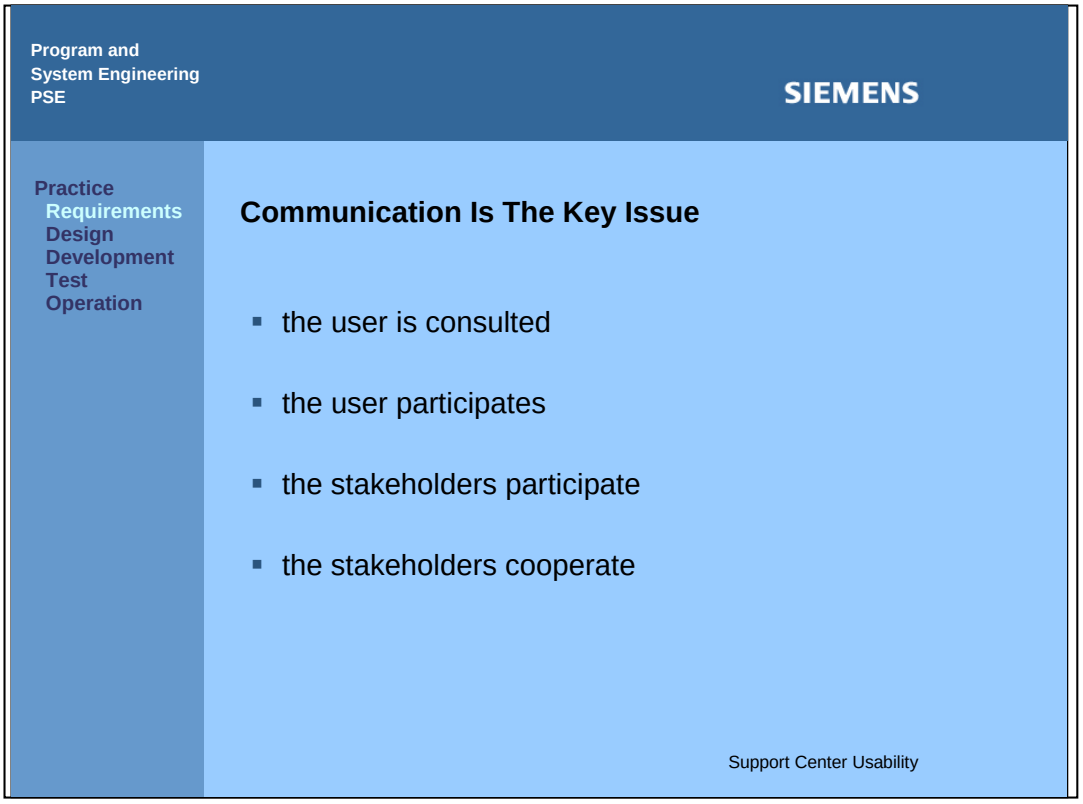
  In the requirements or analysis phase the basis for the product is laid. (What do we do)
- In the design phase the product is designed in such a way, that it fits to the requirements, fulfilling them (how do we do it)
- In the development phase the design is put into reality.
- In the testing phase one takes a good look at the product to get it to perform well and to conform to the design
- In the operations phase the users are using the product. That may be a mobile bought in a shop or some word processor, of course also bought, or a ticket vendor machine at the train station or even a coffee machine. The users get to work with what was analysed as being the right thing, then designed for them, developed and tested.
- Now how can Usability Engineering support that process?



- The starting point is where somebody decides to start a project.
- Usually that somebody is a company, having an idea how to gain money.
- Other stakeholders in a product are the users, who usually want a tool to make their life easier or entertainment, to enrich their life…
- Still other stakeholders may be the system administrators on whose system the product will run or the trainers who will teach other people how to use the product.



- As I said before, Requirements Engineering lays the basis for the product that is being developed. A lot of the success of a product depends on whether it meets the requirements of the potential customers and/or users.
- This slide shows you a definition of Requirements engineering. The important issues have been highlighted.
- Requirements are developed, you gather some ideas and refine them
- this is an iterative process, you won't catch them all at the first go, as your understanding deepens you will see further questions and possibilities. More than half of the RE projects do >3 iterations.
- One of the tricks of Requirements Engineering is to know when to stop and not to research for years on end, as it is always possible to do.
- The elicitation is also a cooperative process. You need the cooperation of the users, of the user's system administrators, of the user's administrators, of marketing people, of usability experts, etc.
- It is also necessary to check if what you understood is what the other wanted to tell you. It is also necessary to check the found requirements. They have to be consistent, complete and unambiguous in order to provide a good basis.



The user is consulted:
questionnaires, interviews, observation
user is quite passive, source of information
engineer is responsible for elicitation
- alternatively define viewpoints
- name a viewpoint authority

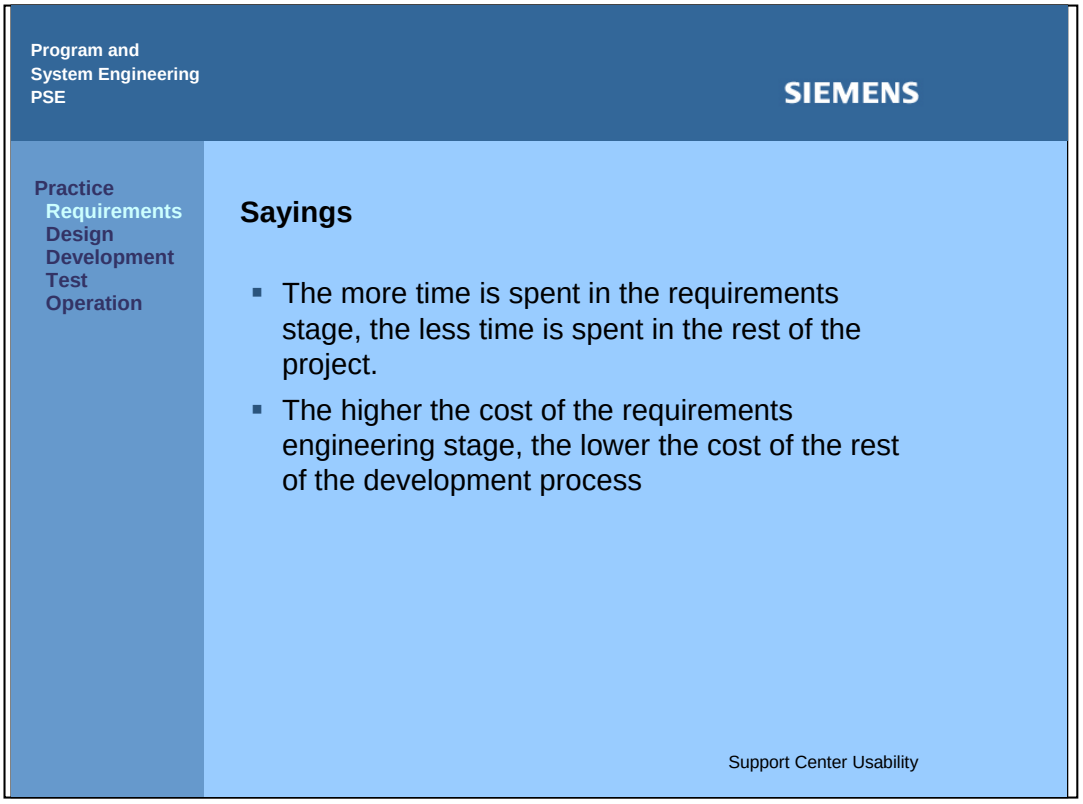
engineer has to understand the problem domain

The user participates
workshops
user is member of the team
assists in analysis
types of users
- Primary: frequent, hands-on
- Secondary: occasional or through intermediary
- Tertiary: affected by introduction or influence on purchase, not hands-on

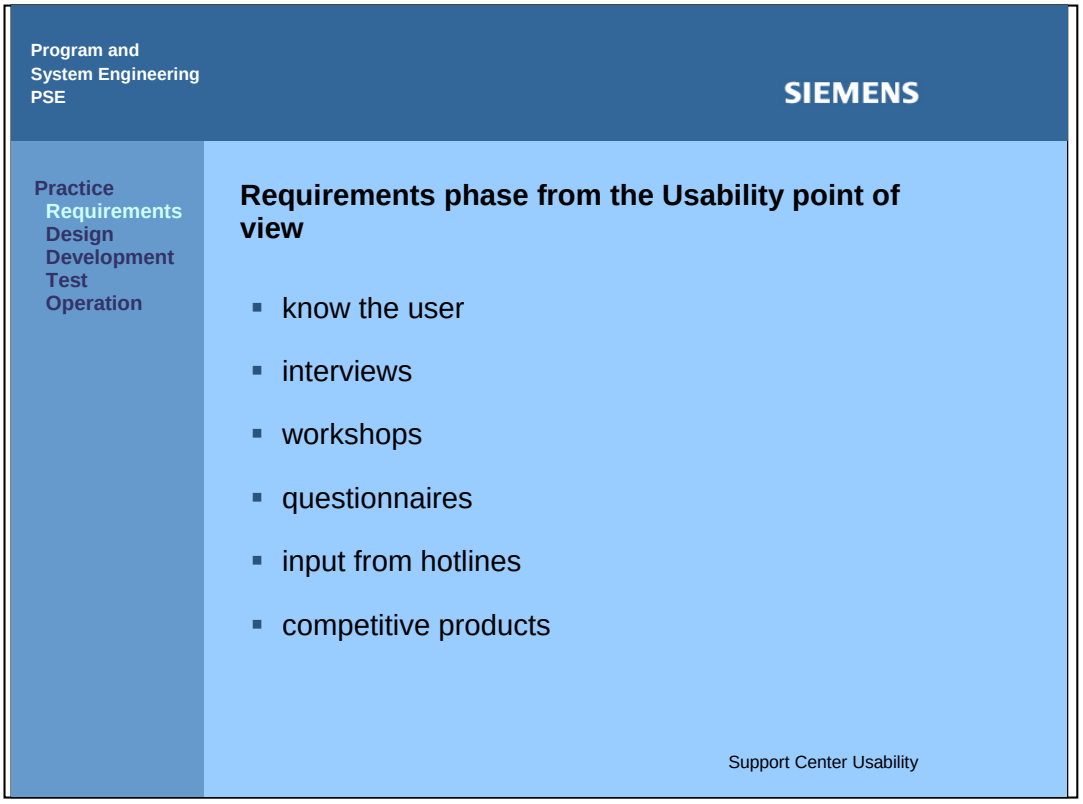
The stakeholders participate
all who have stake in the system under consideration
- may gain
- may lose

potentially incompatible perspectives
potentially conflicting objectives

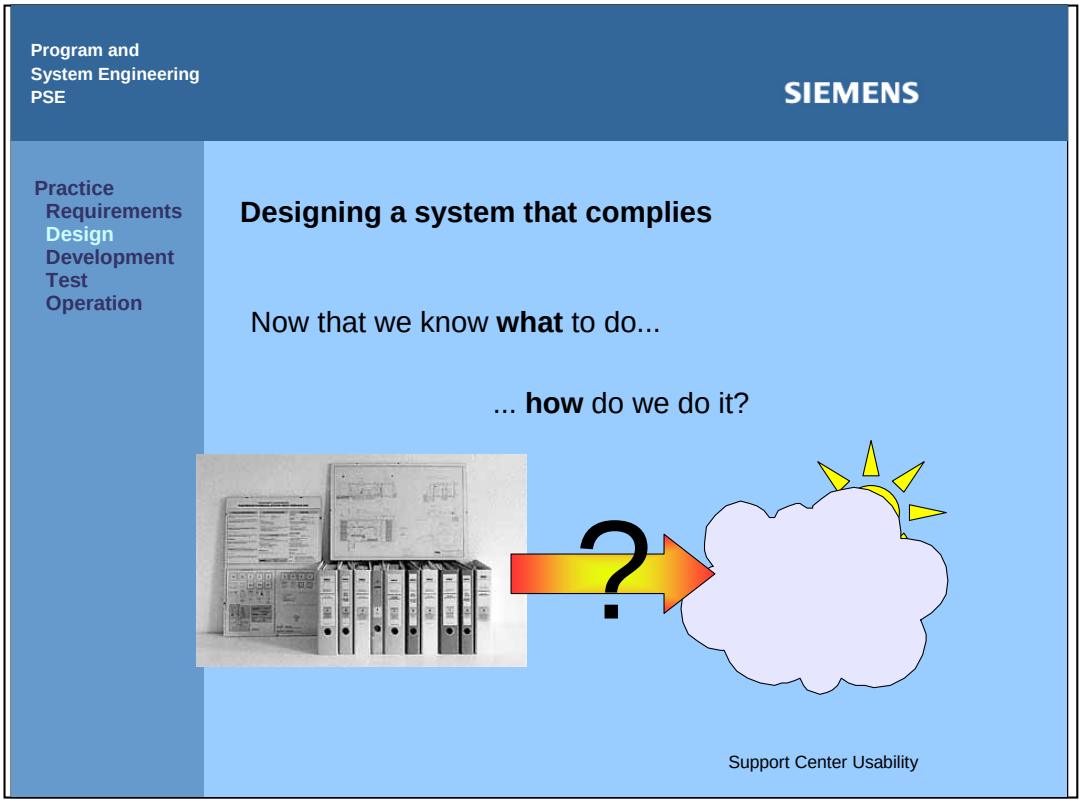
The stakeholders cooperate (the top discipline)
all are actively involved in making decisions as to the scope of the new system
danger of too large group meetings
group dynamics
moderation might be a good idea



- no unnecessary developments
- io you found the underlying rules and functions – they won‘t change as often as superficial process steps do



- Classify the users
- go to them
- try to find out the underlying functionality under everything the users do!
- Find out communication and information needs
- get input from various sources



We've got a lot of ideas on paper, let's find a way how to realize them.
Throughout computer history the interfaces have become ever more refined…

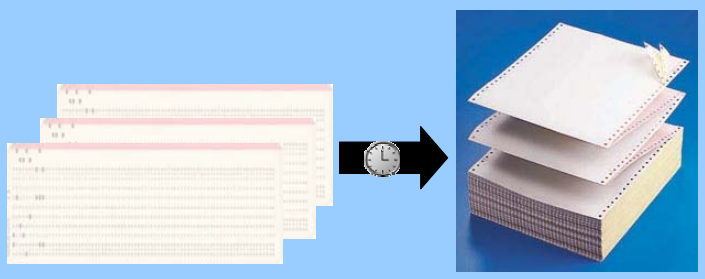
Program and
System Engineering
PSE

SIEMENS

Practice
Requirements
Design
Development
Test
Operation

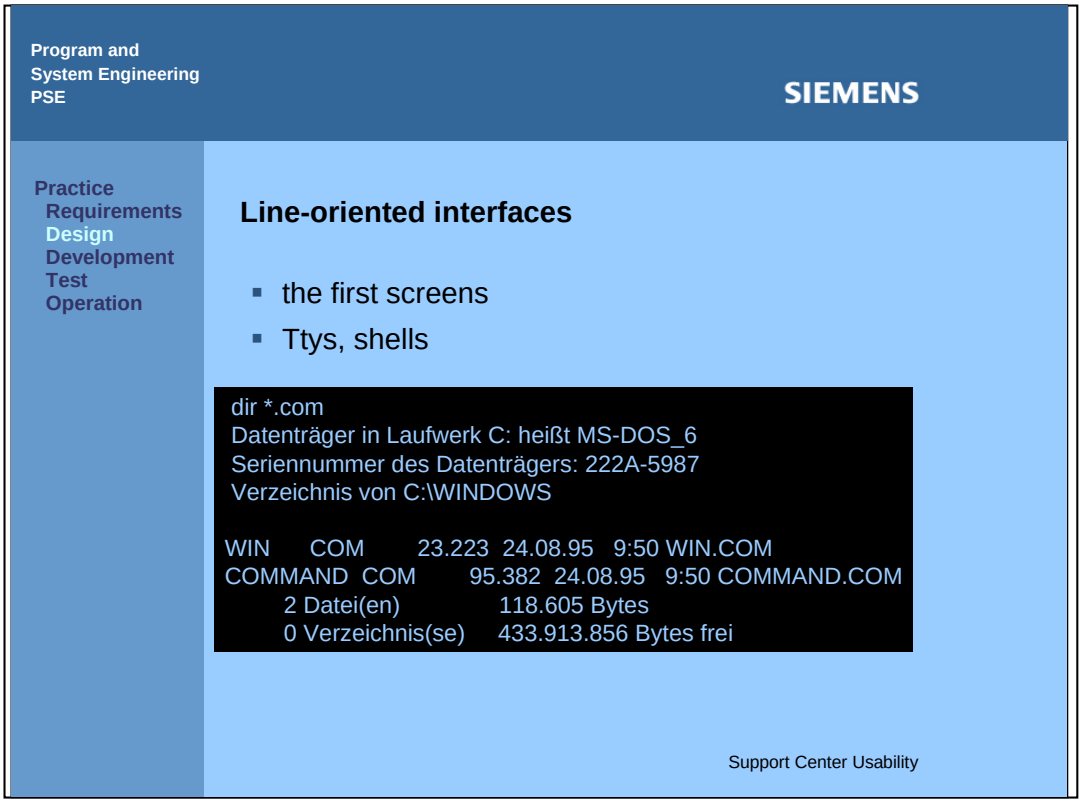
batch-oriented interfaces

- no screens
- type-writer -> loads of paper-printout

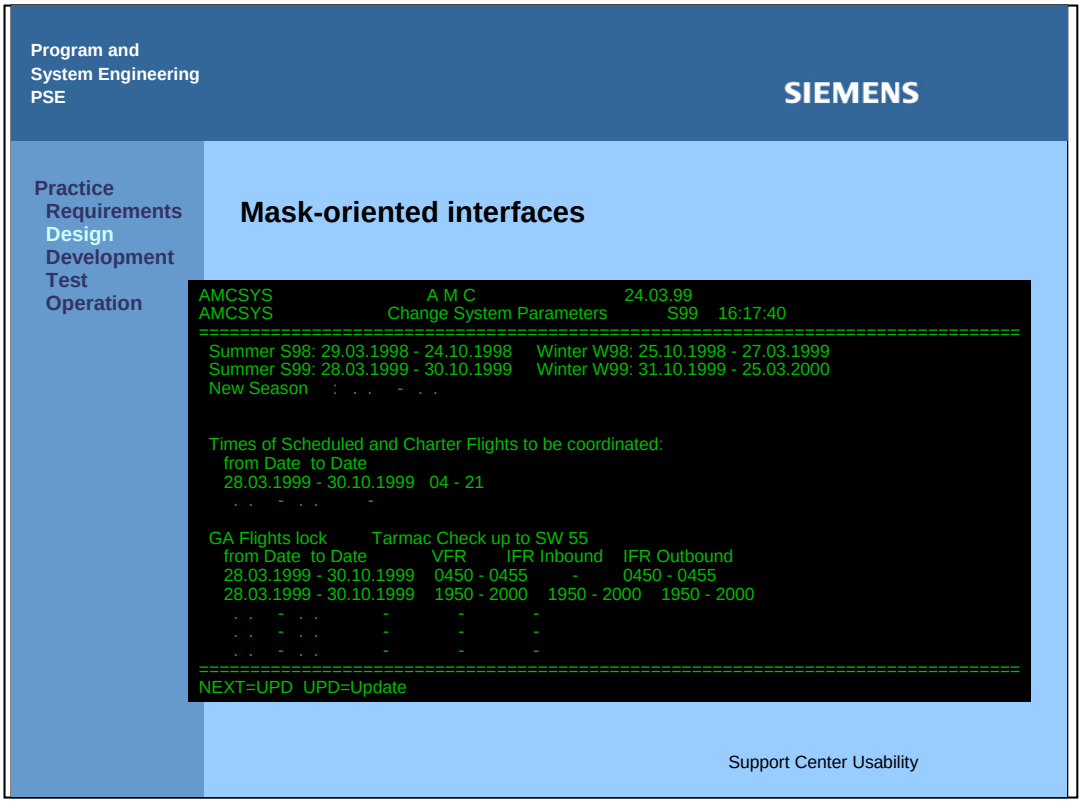


Support Center Usability

- no possibility of correction or interference



- a bit more immediate, but still no correction possible, from hitting the return key to gettting the screen printout.
- Data scrolls off the screen (even if there‘s a line memory, as in shells, at some time the lines will fall out)



- data does not scroll away
- certain applications (e.g. in the airline reservation system) still work with such interfaces
- supports function keys

Program and
System Engineering
PSE

SIEMENS

Practice
Requirements
Design
Development
Test
Operation

Graphical interfaces

WIMP technology Windows, Icons, Menus and a pointing device

Support Center Usability

Program and
System Engineering
PSE

SIEMENS

Practice
Requirements
Design
Development
Test
Operation

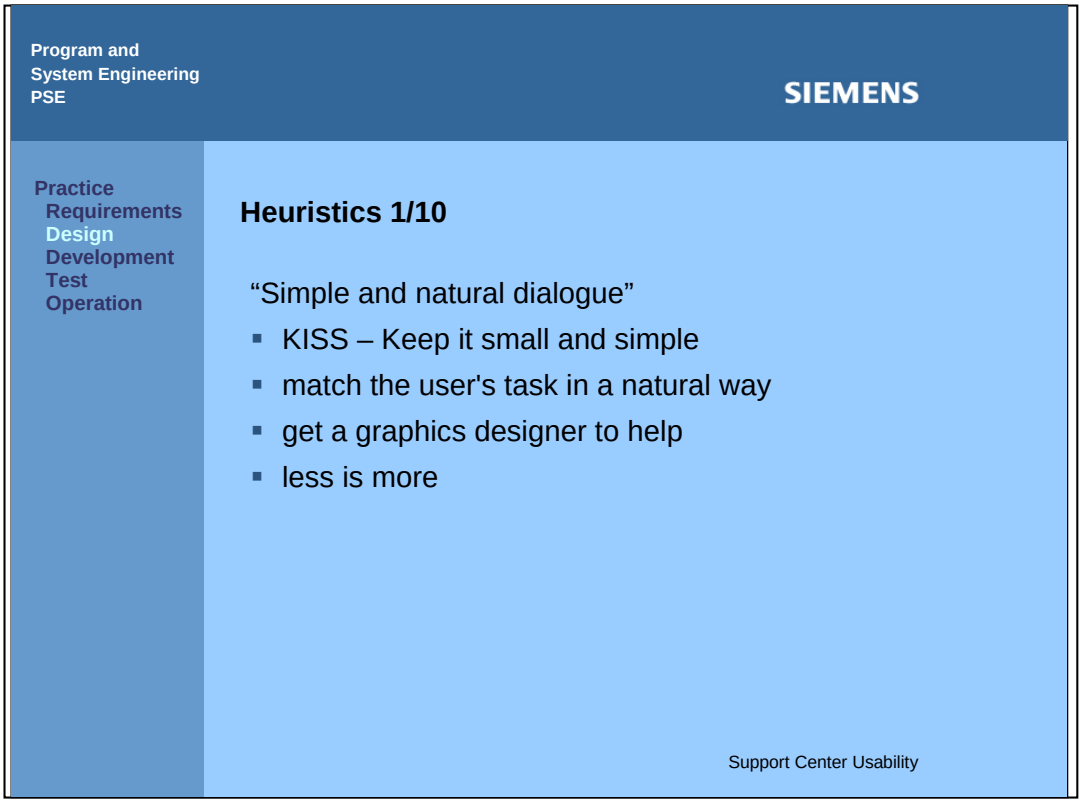
Graphical interfaces

... including Internet

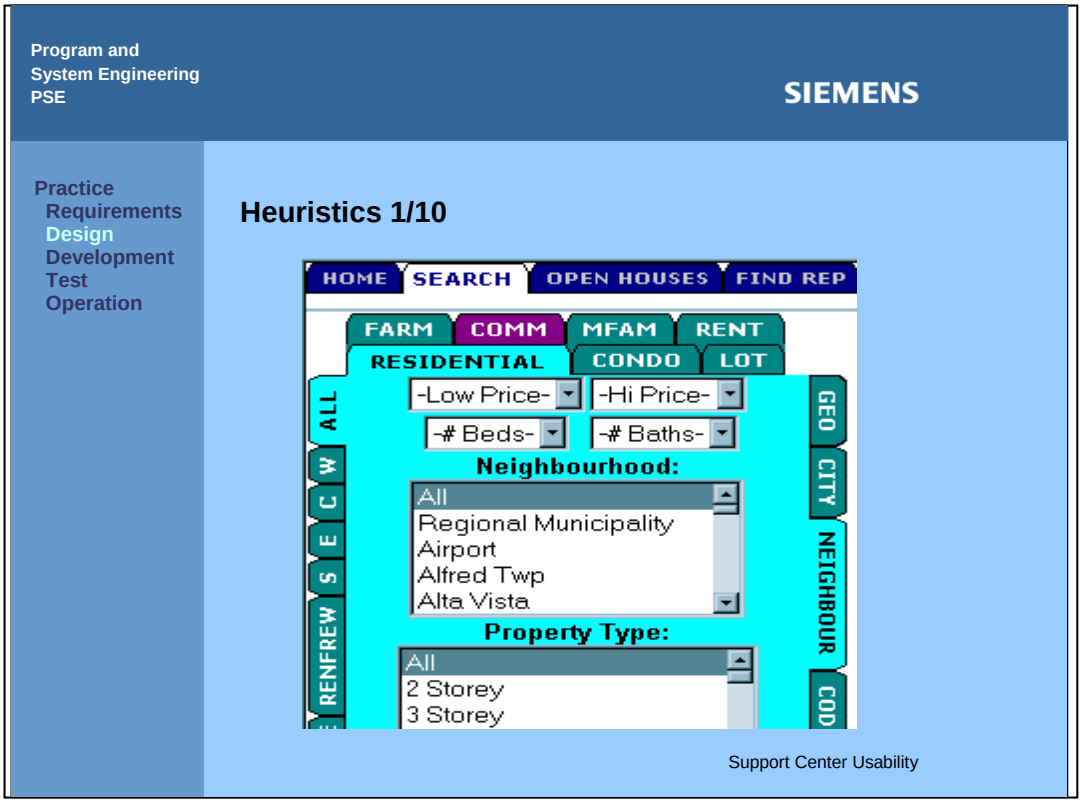
Support Center Usability

are graphical interfaces as well, with different limitations and advantages (links)

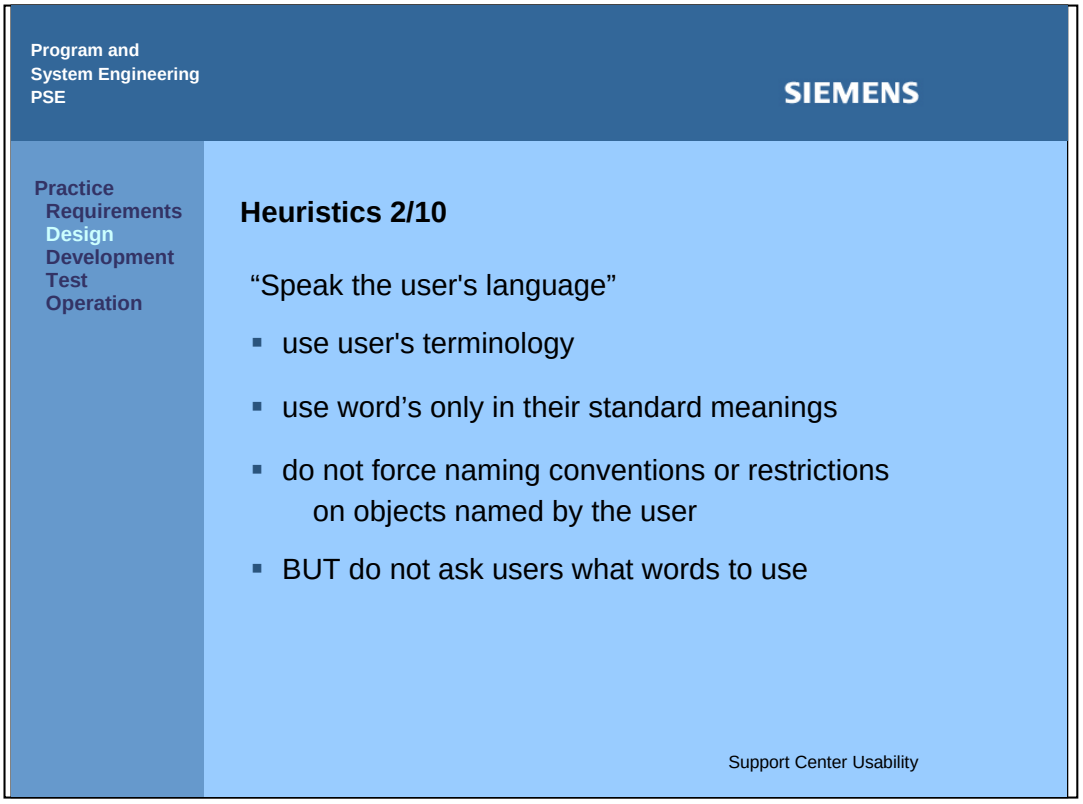
scene from “Minority Report”. The protagonist moves the data on the screen by moving his arms and hands.
Very intuitive, but probably quite exhausting.
There is much research going on regarding new interfaces: data gloves, 3D displays,…



KISS – also known as “Keep it simple, stupid!“
graphics designer knows about the psychological tricks when placing things, like how things can be perceived connected by placing them near each other, how many colors to use and how to use them, not to use screaming background colors, where and how to catch the attention of the user
BUT the designer is NO usability expert! What is designed absolutely stunningly beautiful still may be completely unusable.
Keep in mind, that there are a considerable number of colorblind people out there – do not put information into color alone!
abstract away from the content in order to focus on the layout
what is used together should be placed together
don't distract with unnecessary information, only place what is truly important. Additional information can be put into a „detail part“ or behind a button or into a tooltip text (though not too much here)


Program and
System Engineering
PSE
SIEMENS
Practice
Requirements
Design
Development
Test
Operation
Heuristics 1/10
HOME
SEARCH
OPEN HOUSES
FIND REP
FARM
COMM
MFAM
RENT
RESIDENTIAL
CONDO
LOT
-Low Price-
-Hi Price-
-# Beds-
-# Baths-
Neighbourhood:
All
Regional Municipality
Airport
Alfred Twp
Alta Vista
Property Type:
All
2 Storey
3 Storey
ALL
W
C
E
S
RENFREW
GEO
CITY
NEIGHBOUR
COD
Support Center Usability

bad example



Studies found out that asking many users results in words appropriate for 15-36% of the users – that's not many
let the users vote from a short list of alternatives
this needs a good understanding of users and their domain

Program and
System Engineering
PSE

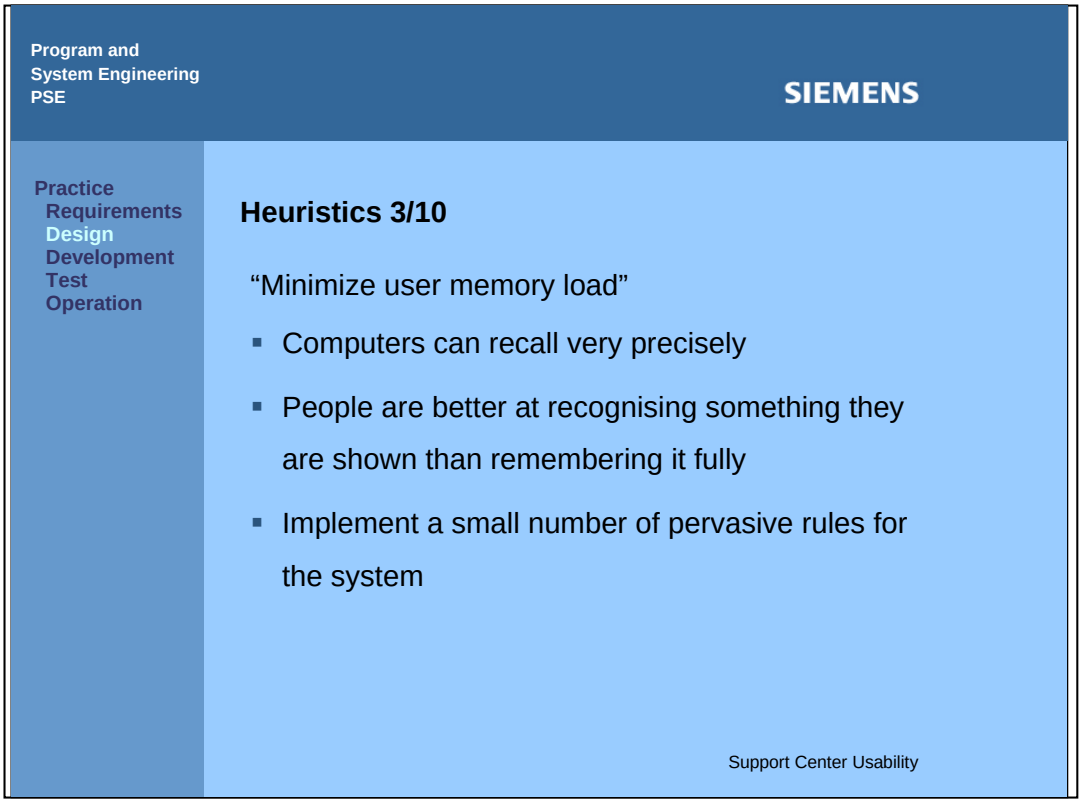
SIEMENS

Practice
Requirements
Design
Development
Test
Operation

Heuristics 2/10

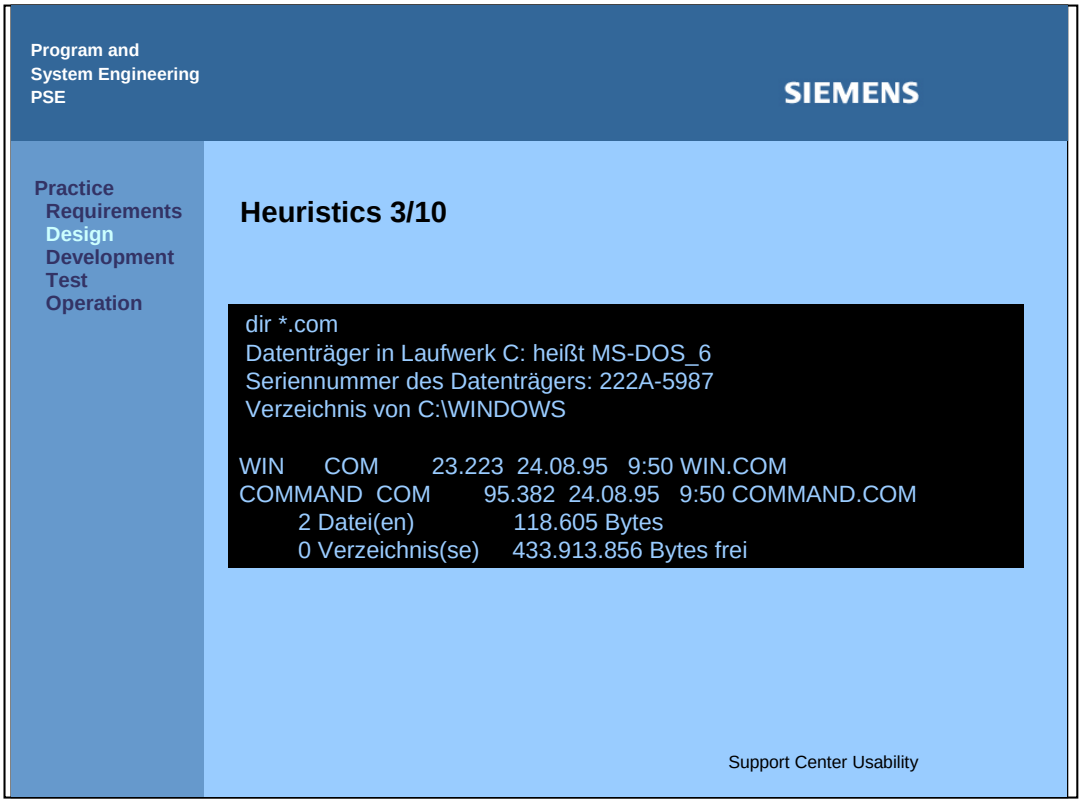
Support Center Usability

An interface for an elevator:
How do I get out of the house?
if there are more than 10 floors – how do I tell it, that my entry is done?
Where is the ground floor? 0 or 1?
How many levels are there? Where is the roof?

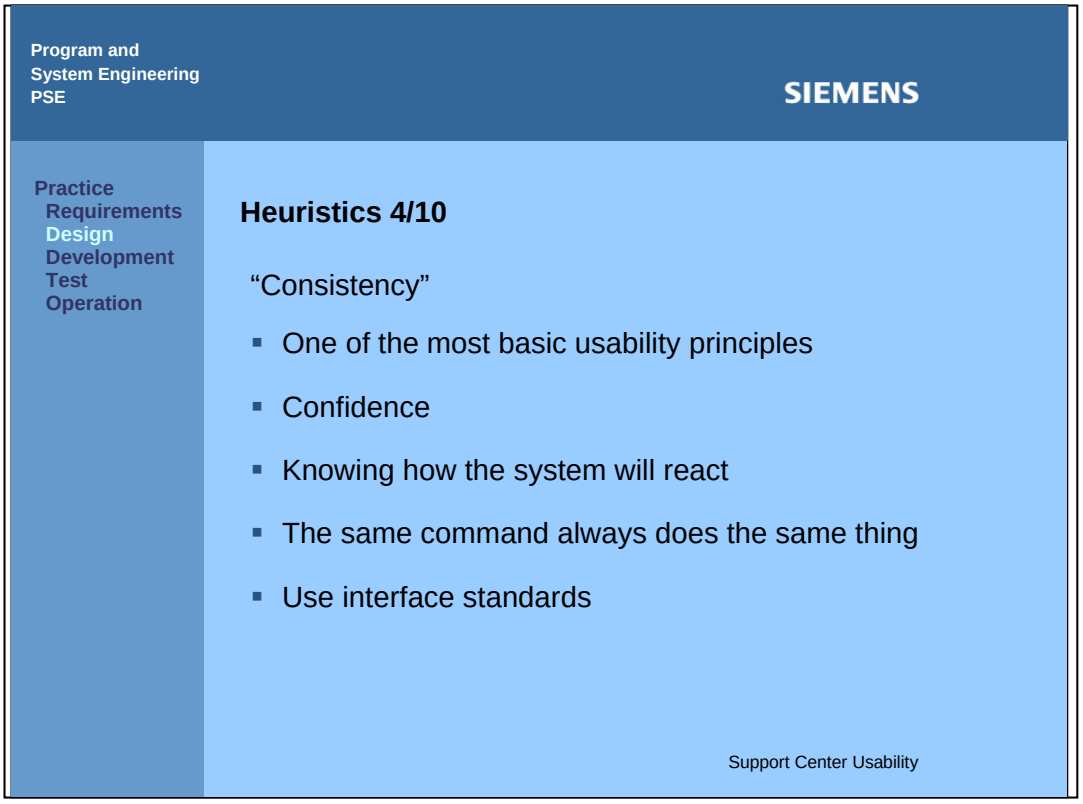


Whenever some input requires a special format give an example in order to show what is meant
icons and menus help you not to have to remember the exact syntaxes of commands,
code completers help you with IDEs (e.g. Eclipse) so you do not have to remember the exact names of the classes, interfaces, methods, etc.

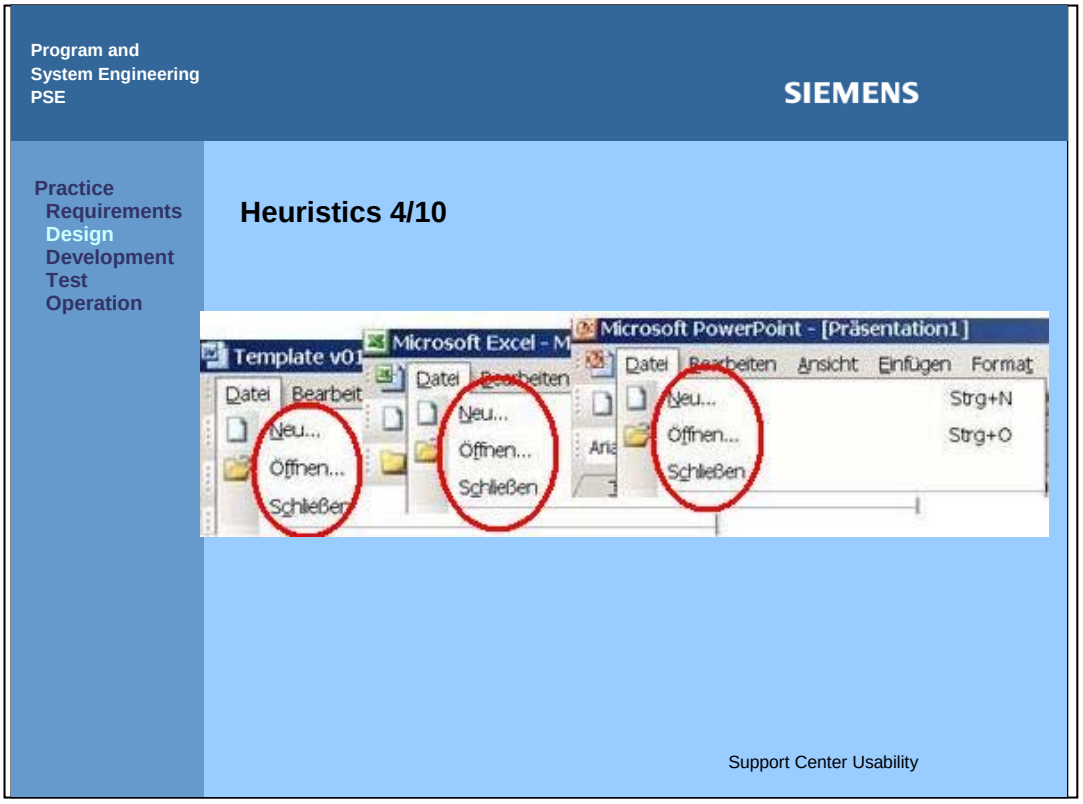
pervasive rules: paste works the same, whatever the object it is applied to click, copy, select goal and paste or you can use the CtrlC, CtrlX, CtrlV, ShiftInsert and CtrlInsert, CtrlDel



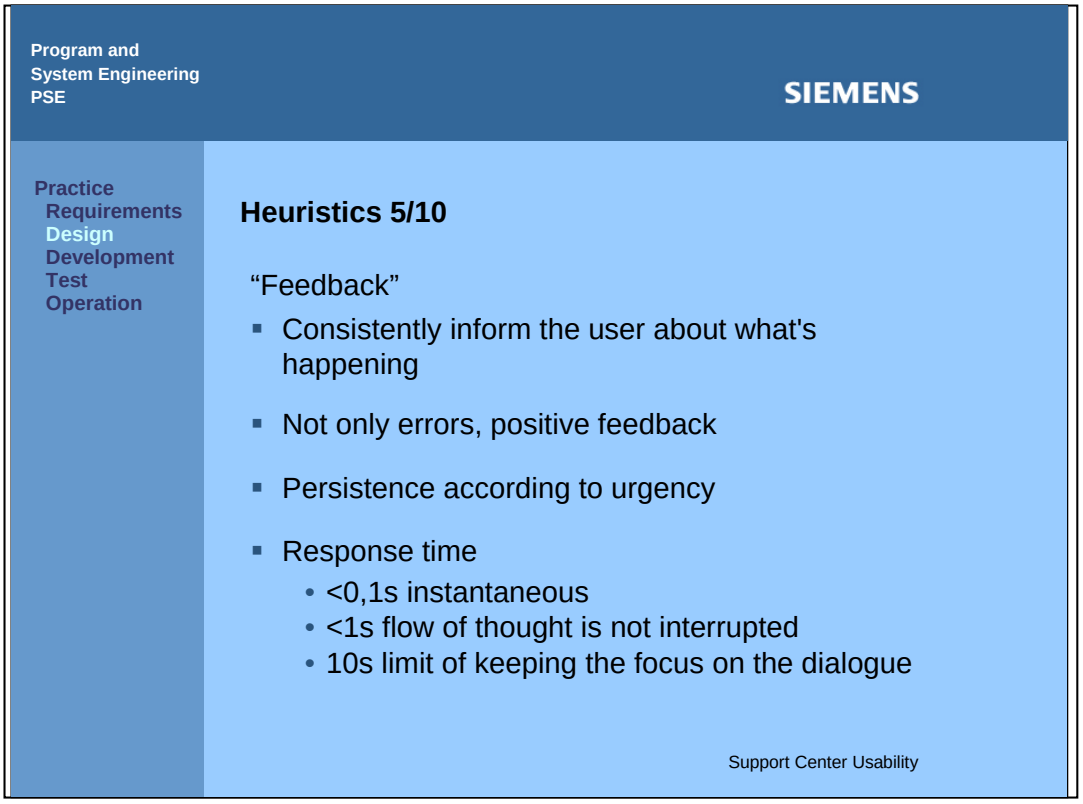
Command line interfaces take heavy tolls of user memory.
Who knows vi? Experts are very quick – until they go on holiday for a month



Do not be afraid that standards will cut down too much on creativity and options for good design. They leave lots of leeway.
be consistent, but do not force an awkward design for the sake of nothing else but consistency – the design should always conform naturally to the user's task.

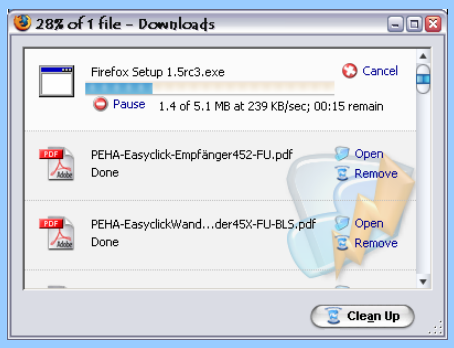


The same commands can be found in the same places in this set of appplications.



Between 1 and 10 seconds, just give feedback, that something is happening, a progress bar is an overkill and disrupting in this case! Something like a number ticking up and the mouse changing to the busy-icon should suffice
more than 10 seconds: provide a progress bar and an option for cancelling the action! Maybe even for stopping and continuing later.
The user will want to switch to other tasks while whatever is being done is being done.
If no progress bar is possible, at least show some animated gif or so in order to show work is being done.
Animations should be timed with the system clock, not with the CPU-clock – otherwise a more powerful computer will speed up the application considerably (as happened with quite a few games in the DOS era)
informative feedback includes error messages, of course

Heuristics 5/10



This tells you how much has been done, how much is still to be downloaded and how long you can expect this to take.
The title bar gives information about the current status, so this will be shown in the task bar as well, being available even while I open other appplications.

Heuristics 6/10

"Clearly marked exits"

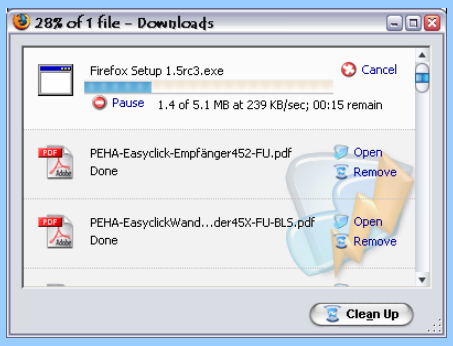
- user should feel in control
- „cancel"
- „undo"
- „escape"
- users WILL make errors, make it easy for them to recover from those errors

Program and
System Engineering
PSE

SIEMENS

Practice
Requirements
Design
Development
Test
Operation

Heuristics 6/10



Support Center Usability

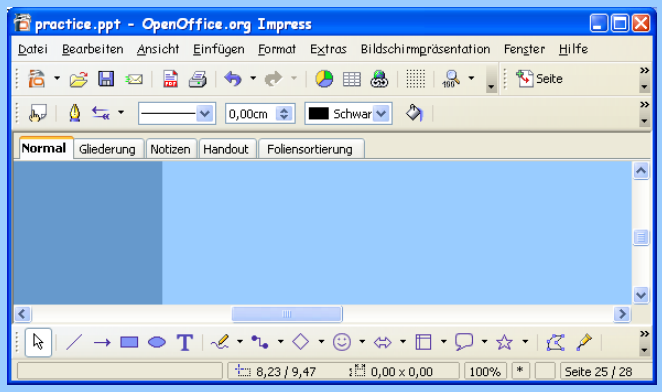
Pressing cancel will stop the downlad, pause will momentarily interrupt it, it can be continued.

Heuristics 7/10

"Shortcuts"

- for experienced user
- abbreviations
- double-clicking
- gestures
- templates
- macros
- scripting facilities

Heuristics 7/10



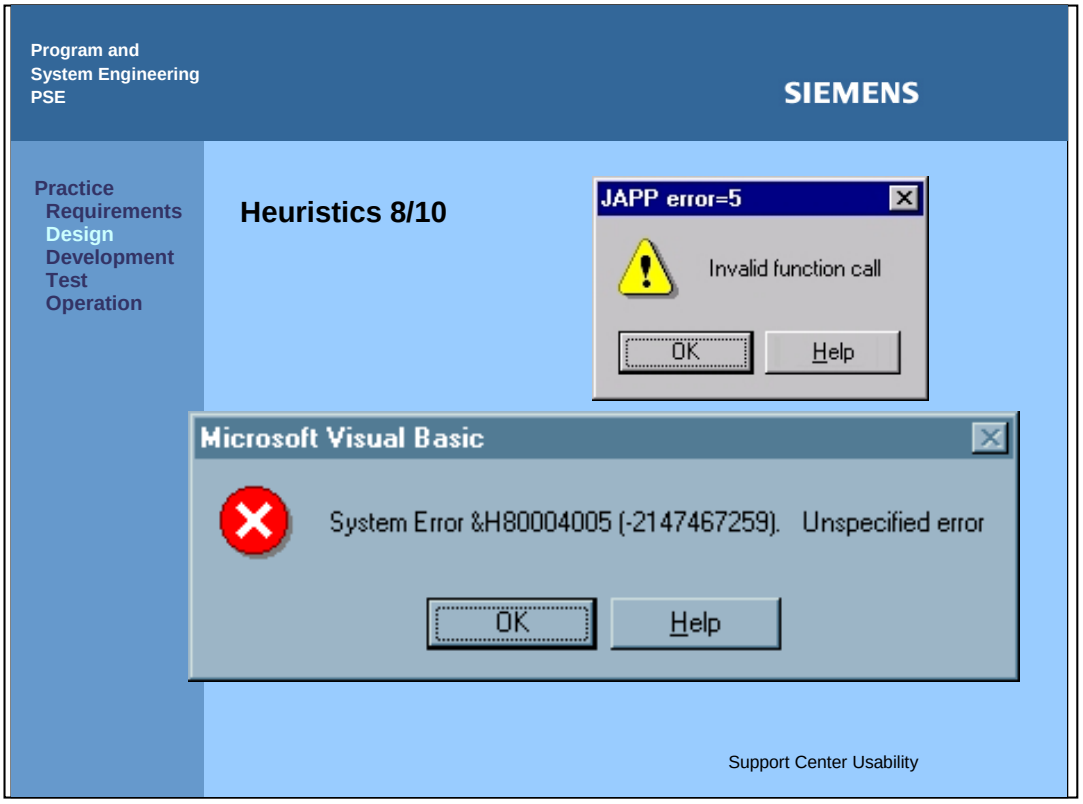
Support Center Usability

Toolbars, Mnemonics for menu commands

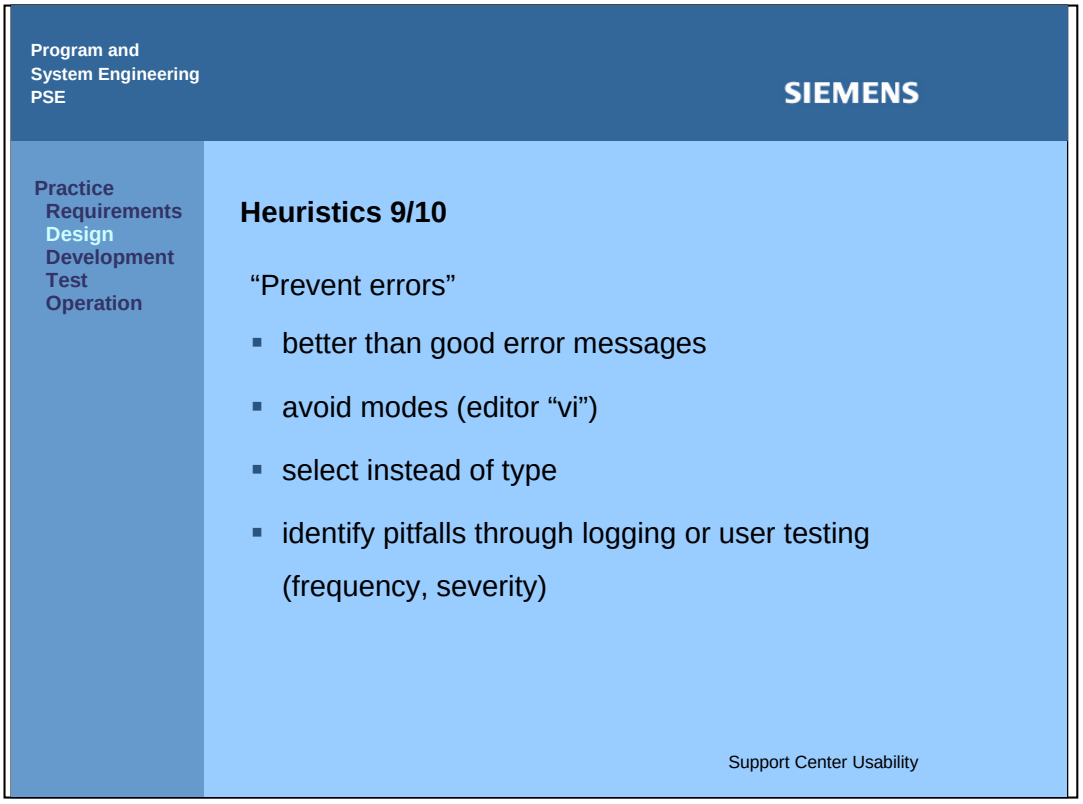
Heuristics 8/10

"Good error messages"

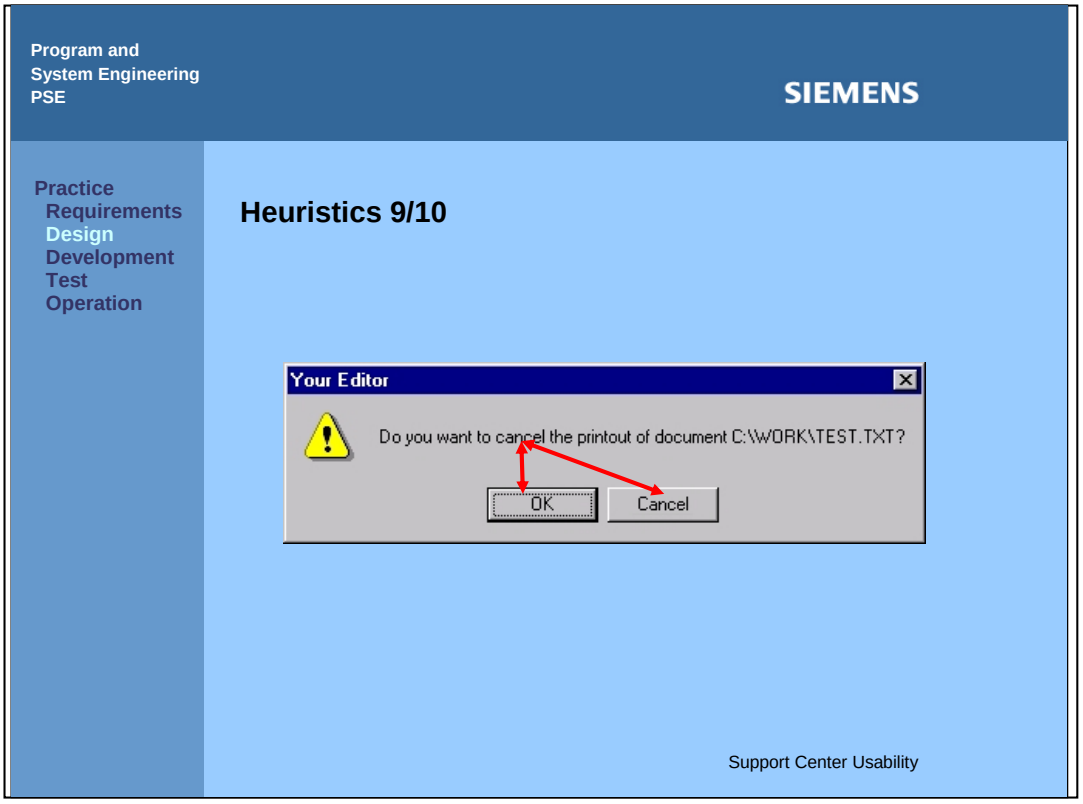
- phrased in clear language, avoid obscure codes
- precise, rather than vague or general
- constructevily help the user solve the problem
- be polite, don't intimidate, don't put blame on the user explicitly
- provide good error recovery



no comment necessary



if modes are necessary, mark them really clearly (different backgrounds, whatever)
if at all possible, let the users choose from a selection instead of having them type the entry



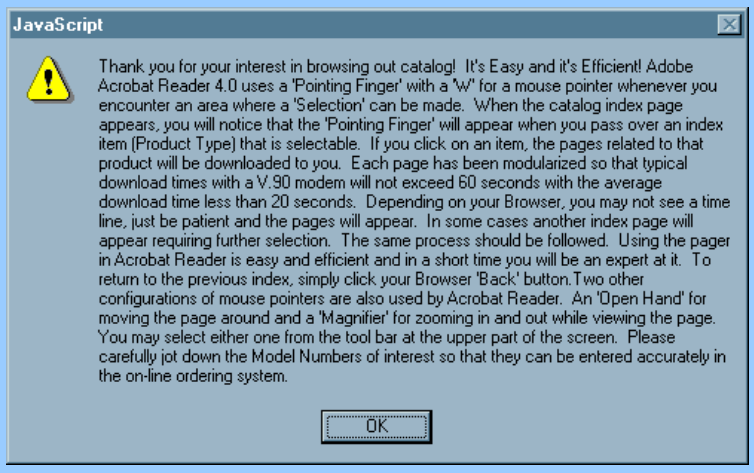
Cancel in the text and Cancel on the button are very similar, people in a hurry will probably correlate the two and press cancel....

Heuristics 10/10

"Help and Documentation"

- **Help doesn't!**
- users do not read manuals
- manuals/online documentation are fallback – design them for that
- test the documentation

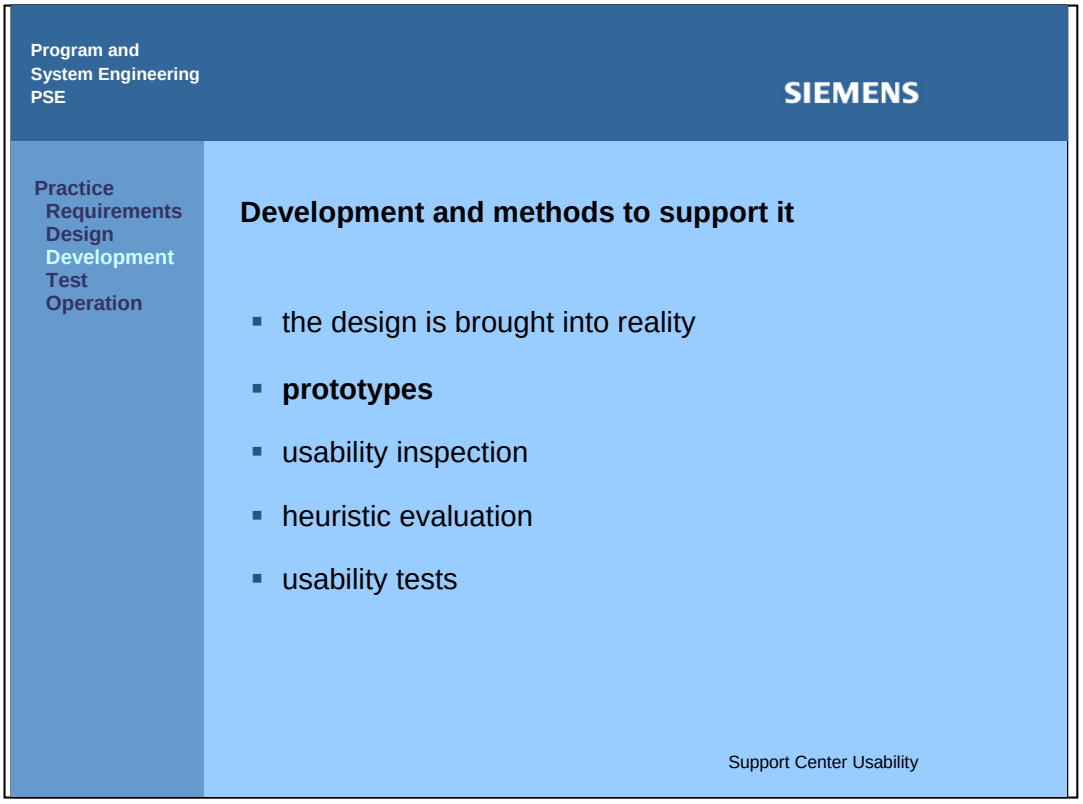
Heuristics 10/10



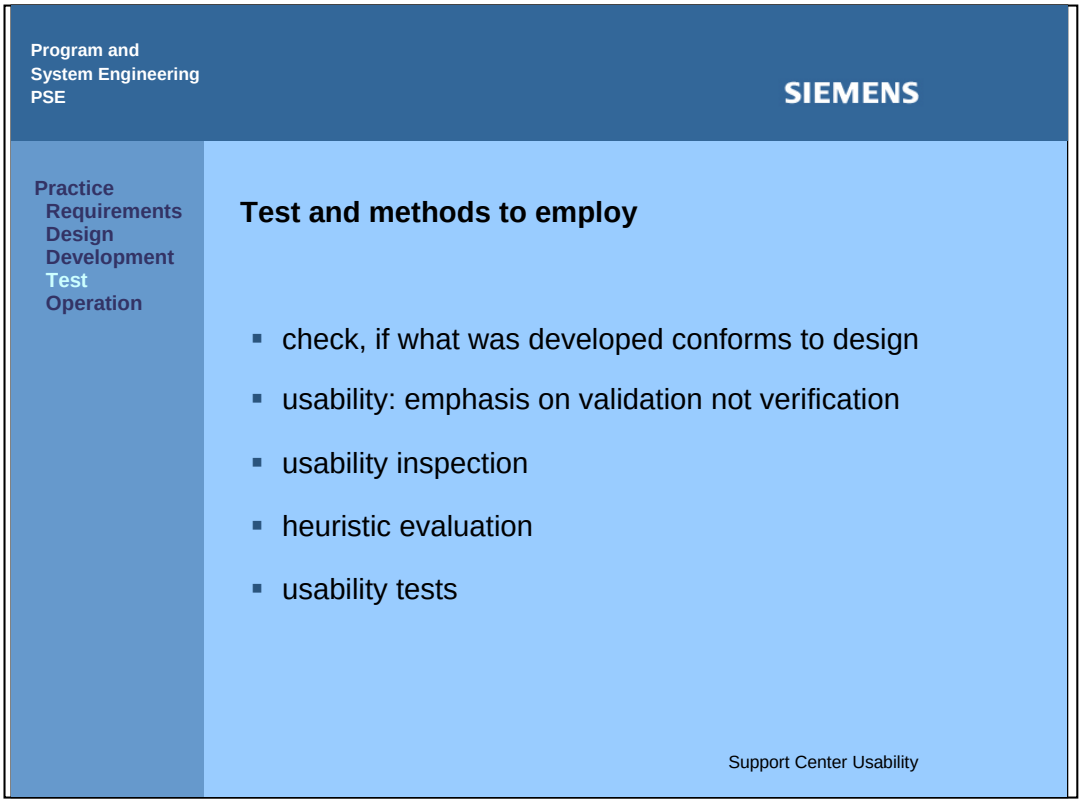
Support Center Usability

Usability Methods for supporting design

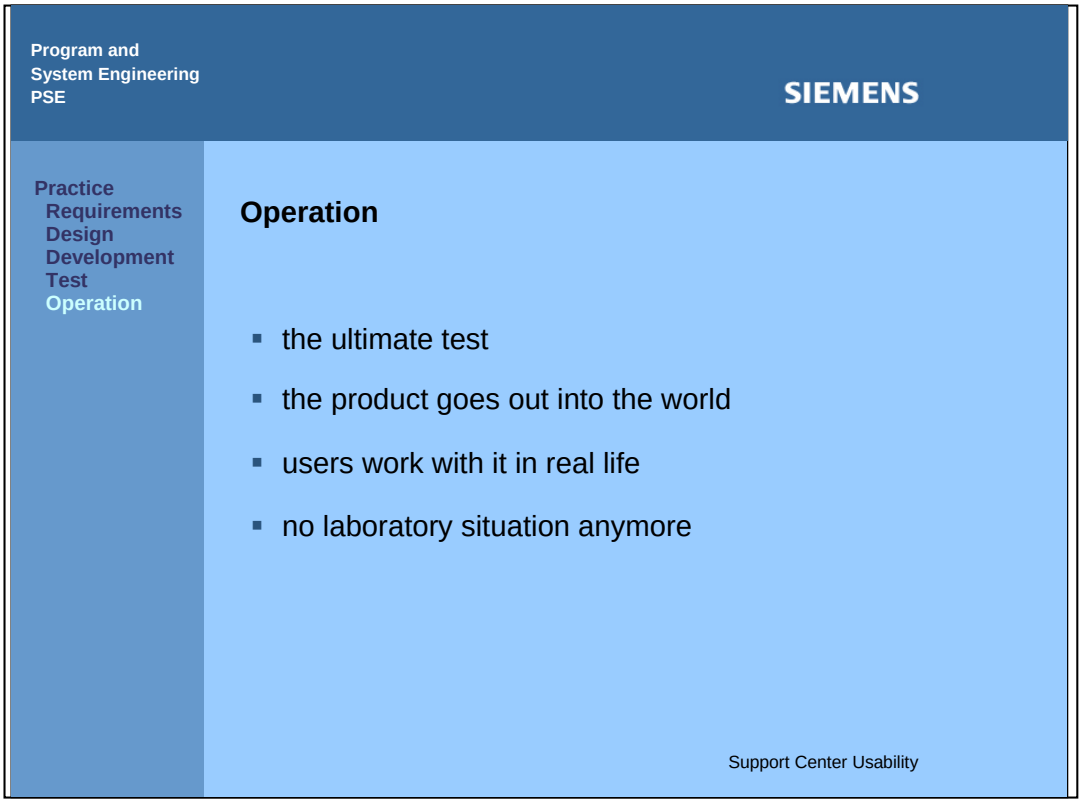
- workshops
- **(paper) prototypes**
- heuristic evaluation
- usability inspection
- cyclic approach



not much to do during development in general

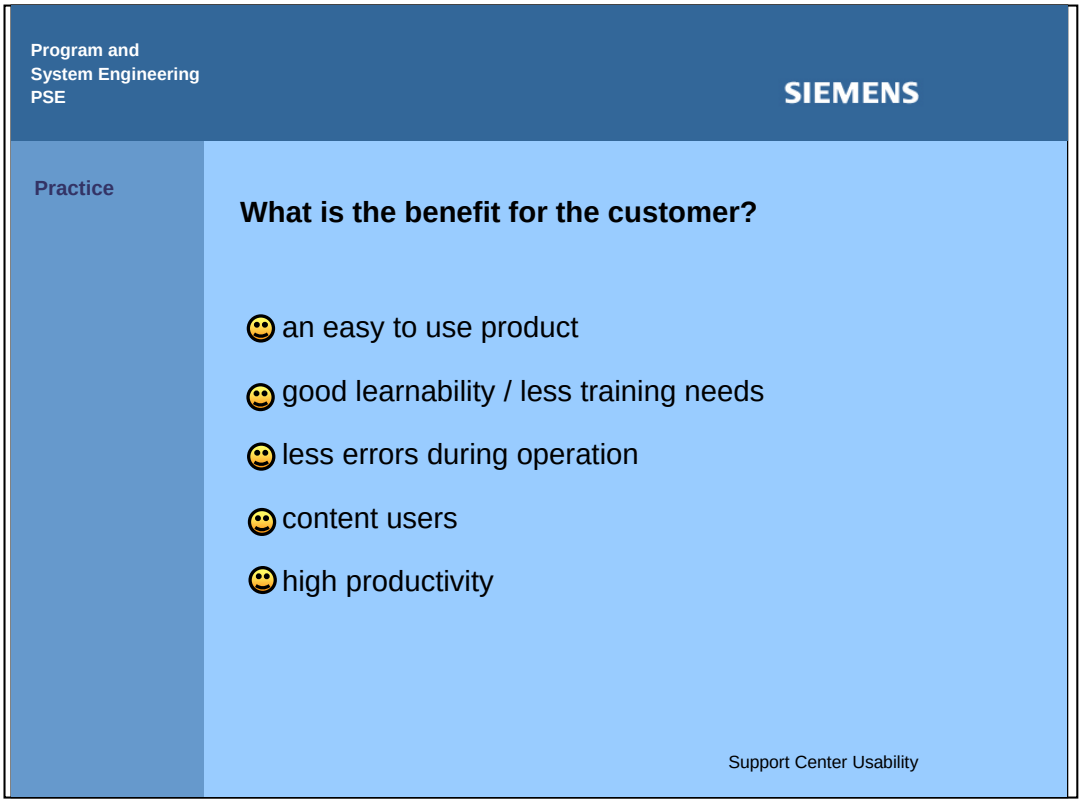


Those methods can be employed here, but keep in mind, that any problem found here costs much more in terms of time and money to eliminate than if it had been found at the beginning!
Usability Tests to verify that specified usability goals have been met (e.g. learnability)


Program and
System Engineering
PSE
SIEMENS
Practice
Requirements
Design
Development
Test
Operation
Operation
the ultimate test
the product goes out into the world
users work with it in real life
no laboratory situation anymore
Support Center Usability

Methods to employ

- interviews
- questionnaires
- evaluation of logs, hotlines
- input for new versions


Program and
System Engineering
PSE
SIEMENS
Practice
What is the benefit for the customer?
an easy to use product
good learnability / less training needs
less errors during operation
content users
high productivity
Support Center Usability

Practice

What is the benefit for the developing company?

- the **correct** product is being developed
- only those functions are developed the customer needs (KISS)
- usually development is done quicker and less expensive
- operation is less expensive)e.g. hotline
- more and more usability becomes a requirered feature
- usability becomes a marketing argument
- more orders due to high customer satisfaction

Program and
System Engineering
PSE

SIEMENS

Thank you for your attention!

Gerda Richter

Support Center Usability
Siemens PSE Austria

gerda.richter@siemens.com

Support Center Usability

Recommendations for the interested reader

- Usability Engineering, Jakob Nielsen, 1993
- Emotional Design, Donald Norman, 2004
- Requirements engineering, Linda A. Macauly, 1996
- Designing the User Interface, Ben Shneiderman, 2003 4th Ed.
- Leonardo's Laptop, Ben Shneiderman, 2003

- www.useit.com – Jack Nielsen's site
- www.usabilitynet.org – resources for practitioners

Support Center Usability