

Guided Exploration: an Inductive Minimalist Approach for Teaching Tool-related Concepts and Techniques

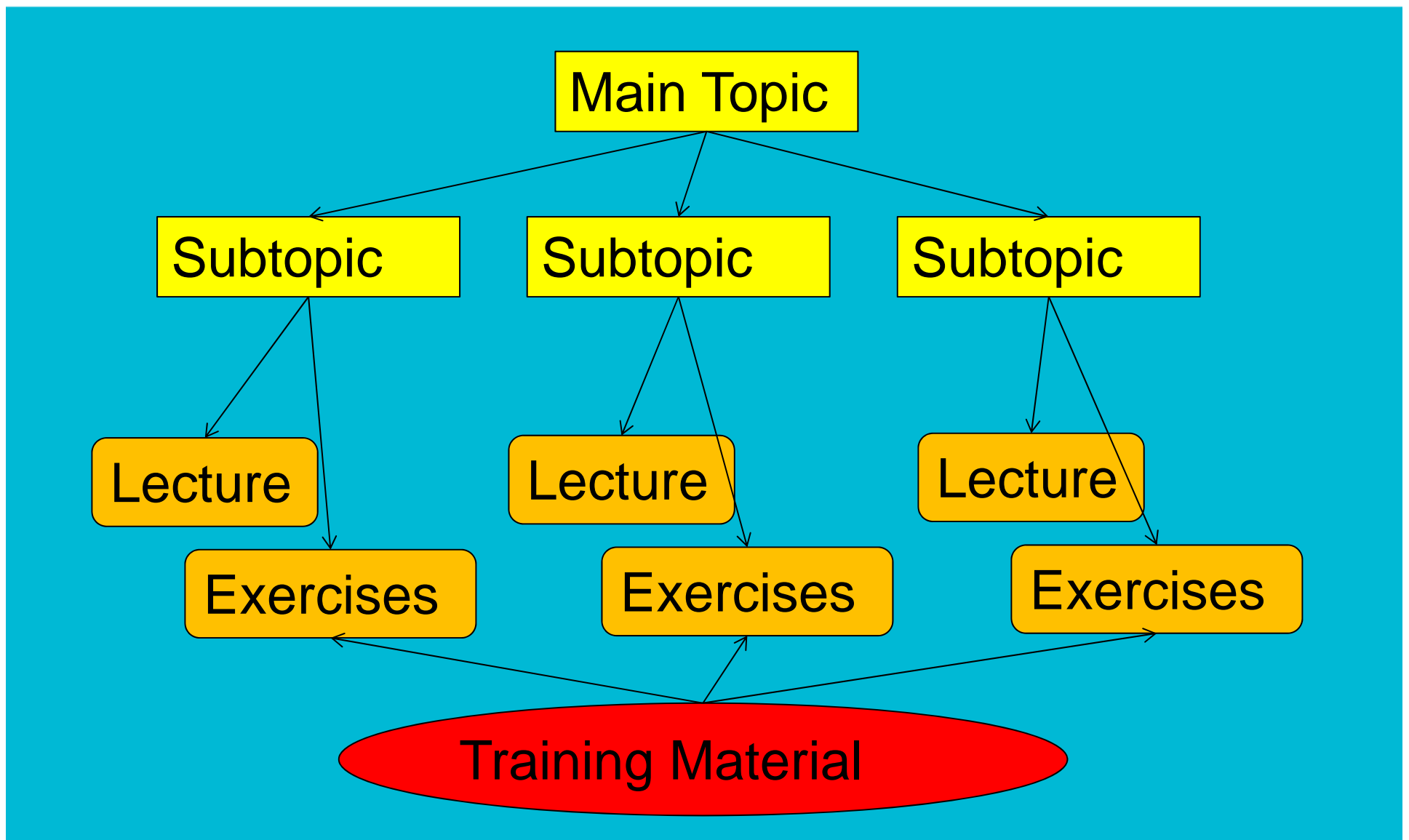
Christian Köppe
Rick Rodin

CSERC 2013
05-04-2013



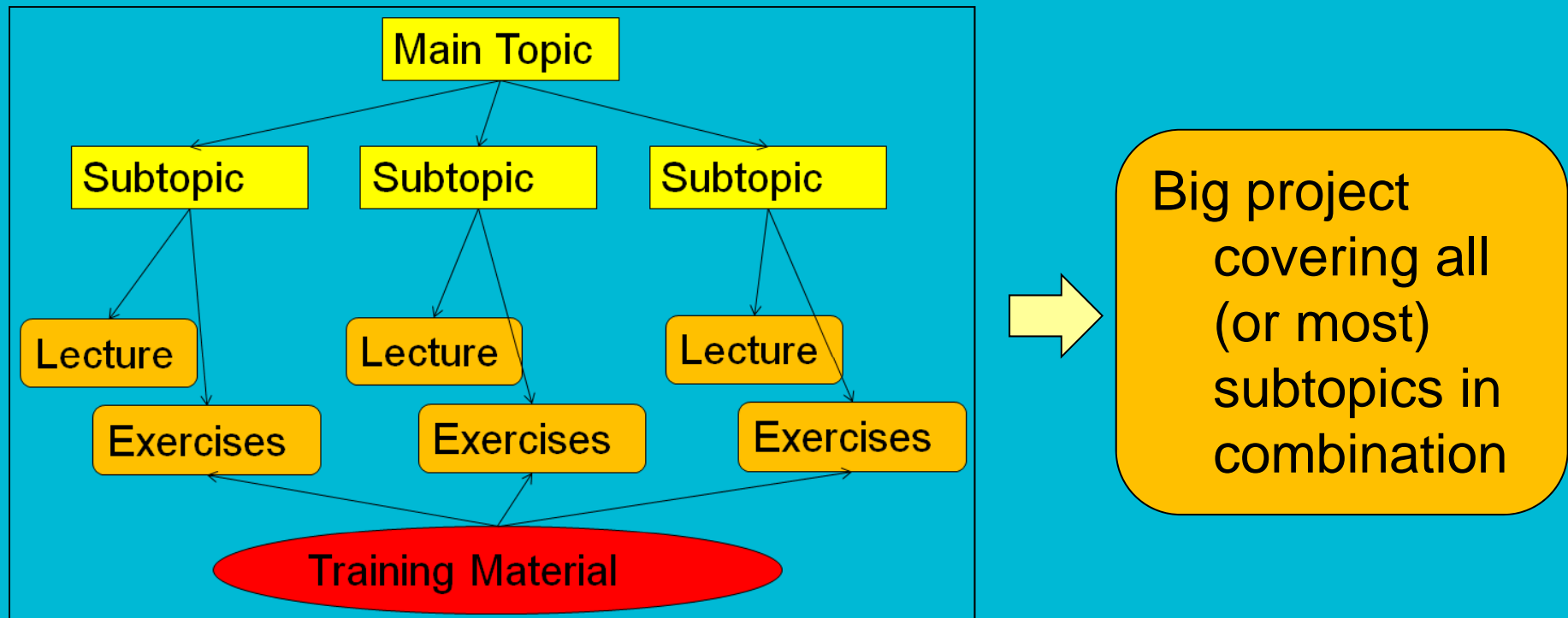


The Systems Approach



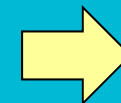
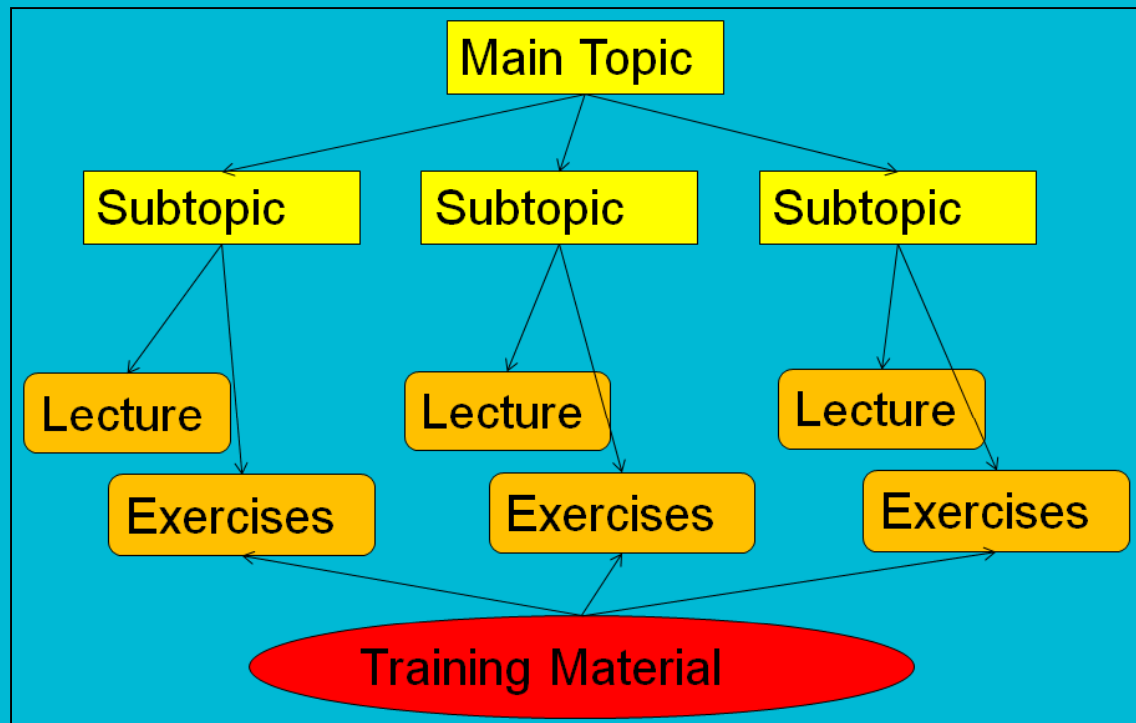
The Systems Approach

- Often accompanied by a larger project where all knowledge and skills have to be combined:



The Systems Approach

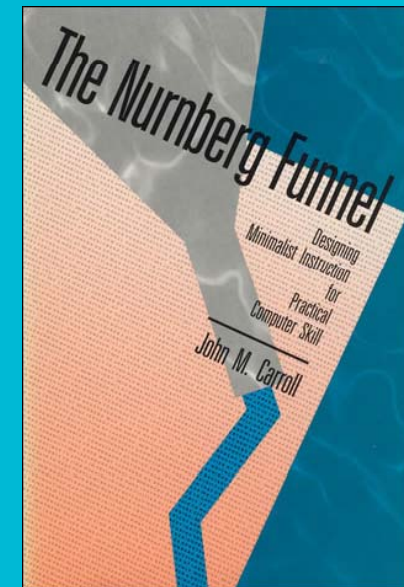
- What's wrong with this?



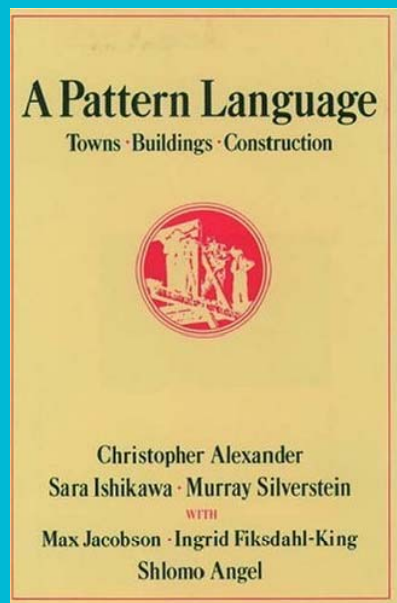
Big project covering all (or most) subtopics in combination

The Nurnberg Funnel

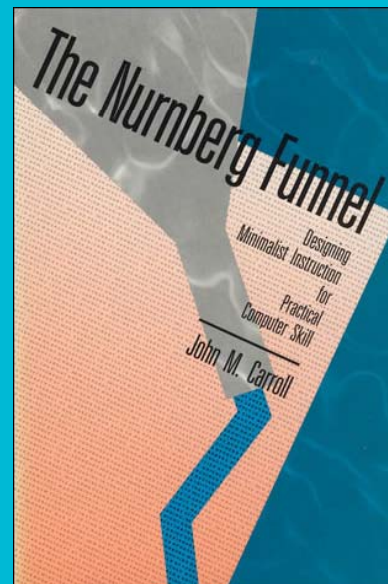
- By John Carroll (1990)
- Minimalism:
 - Allowing learners to start immediately on meaningfully realistic tasks.
 - Reducing the amount of reading and other passive activity in training.
 - Helping to make errors and error recovery less traumatic and more pedagogically productive.
- Huge overlap with inductive teaching



Meta-Patterns for Minimalist Training Design by R. Rodin



+



=

Meta Patterns for Developing a Minimalist Training Manual

RICK RODIN, Pace University, rck.rodin@verizon.net

1. Introduction

These Patterns are a guide to the creation of a minimalist training manual and are extracted from existing knowledge in The Nurnberg Funnel by John Carroll. The target audiences of these patterns are any company's organizational units, having a need to train new hires or cross train existing staff on the unit's particular work functions. In some cases new knowledge will be captured and crafted into training materials with this process. Using the eight "principles" of minimalism captured in these Patterns, one can "define" minimalist instruction. There is no deductive theory of minimalist principles; we cannot just crank out a training manual. Design never works this way (1). It would be appropriate to use an Agile process for creating the manual, guided by the eight principles embodied by this collection of Patterns. These Patterns will serve as guides to a process that will create the minimalist training objects; an Agile process is recommended. The company's organizational unit will form a team that will create minimalist Training Objects using this process guided by these Patterns. These Patterns formulate the basic tenants of Minimalism in the form of a pattern collection to guide the creation of Guided Exploration Training Objects taking the form of p-forms. The end result is a pedagogical method called "Guided Exploration." These Patterns differ from the established Pedagogical Patterns in that their target audience is groups of people at work in a typical office environment, performing on-the-job training. This training will be performed by the person doing the work, not a trained educator, for other employees, and not classes of students. The training materials produced by these Patterns are to support this purpose, one for which massive amounts of material are typically needed and very little of good quality actually exists.

1.1. Definitions

Minimalism is a style of writing that uses meaningful realistic tasks, reduces errors and error recovery, and emerged in the early 1990s, around 1990, and supports its effectiveness (1). Minimalism helps with errors. These three basic tasks, Getting Started Fast, Reading Training, Supporting Error Recognition Situation. According to Carroll, "A key idea of minimizing the extent for such design is an eclectic synthesis of small, self-contained units" (1). principles of "Less to Read," and Guided Exploration Cards, which

2. The Pattern Collection

This paper contains the following patterns:

- GUIDED EXPLORATION (starting pattern)
- TRAINING ON REAL TASKS
- GETTING STARTED FAST
- REASONING AND IMPROVING
- READING IN ANY ORDER
- COORDINATING SYSTEM AND TRAINING
- SUPPORTING ERROR RECOGNITION AND RECOVERY
- EXPLOITING PRIOR KNOWLEDGE
- USING THE SITUATION

Meta-Patterns for Minimalist Training Design



- Problems addressed by these also known from education
- Idea: apply the patterns in a course on administration system development (using Oracle APEX)
 - > some additional problems

Guided Exploration as Inductive Teaching Approach



- Material provided in P-Forms
 - P-Form -> pattern format (thus not a pattern)
- Any order of P-Forms possible
 - Default order provided
 - Dependencies included in context/resulting context
- Focus on concepts, not technique/tool
- Helps with errors -> link to original documentation
- Just-in-time teaching (on demand)

Name
Context
Problem
Forces
Solution
Resulting Context
What if I get stuck

Respect User Roles

Context: Your application has different types of users, and some parts of the application are only allowed to be used by specific users.

Problem: Providing all functionality to all users gives some of the users the possibility to work with parts of the application they're not allowed to work with.

Forces:

- Different stakeholders are often interested in different data. An employee who's responsible for entering data might not be interested in monthly reports on sales, and a manager might not be interested in the amounts of items of a specific order, but in the total sales numbers per month.
- Something with user rights/who's working with the application

Solution: Identify all different user roles and their interests. Respect the user roles by adapting the application so that only the parts of the application are shown or available which are relevant for the current user or he/she is allowed to use.

The implementation is dependent of the specific user roles and their interests and rights. A data entry employee should not have the opportunity to add new employees or to run management reports. Standard ways of implementing this are:

- Deactivating or removing menu items,
- A check when the user tries to go to a page if he or she is allowed to go to that page,
- Deactivating or removing page items, like e.g. text-fields or selection lists,
- Setting a default value of an item based on the user role,
- Changing properties of an item, like e.g. from optional to required.

Resulting Context: The user only sees the parts of the application he or she is interested in. If the user is not allowed to use certain application parts, then these are not available or reachable for the user.

What if I get stuck/Hints:

- Remember: Authentication and Authorization (Authentication has to do with granting access only for known-users. Authorization is granting access to parts of the application.)
- For documentation see UserGuide, chapter 13, paragraph "Establishing User Identity Through Authentication" and "Providing Security Through Authorization".
- Authentication is defined and used on application level. Authorization is defined on application, page, region and/or item level.

Fulfillment Criteria:

- No application parts are available — or reachable — for a user which do not match the interests of the user or do fit the rights the user has.





Validate Your Data

Context: The data administration is implemented by ADMINISTRATE SIMPLE DATA and ADMINISTRATE COMPLEX DATA. There are some restrictions on some of the data; e.g. a field is required or another field has to follow a certain format.

Problem: The data in the application can become corrupt or inconsistent if the restrictions on them are not taken into account. Leaving fields empty that need a value or entering a date or email-address in the wrong format can lead to exceptions.

Forces:

- Users often are not aware of all data restrictions, they need help with them.
- If incorrect or corrupt data get into the database, it is difficult to repair or clean them afterwards.

Solution: Implement all simple validations in the user interface, like required fields or simple format checks.

Resulting Context: Only valid data can be entered in the application. The user gets immediate feedback if some data are required but not filled in or if the used format is not correct, this also improves the usability of the application. If the validations become more complex, apply ENFORCE BUSINESS RULES.

What if I get stuck/Hints:

- Datatypes are also restrictions on data, using the correct datatypes often offers some automatic checks.
- Documentation concerning text input for this topic can be found with the keyword: "Validation". In the UserGuide see chapter 8, paragraph "Validating User Input in Forms".

Fulfillment Criteria:

- All data which are entered in the application are validated according to the previously defined criteria.



P-Form list

- Use sample application
- **Functionality Description**
- **Administrate complex data (4)**
- **Administrate Simple Data (5)**
- **Application Cursor**
- **Enforce business rules (6)**
- **Get Data in a Report (3)**
- **Good Naming**
- **Help with value selection**
- **Navigation Support**
- **New Application**
- **Provide a Database**
- **Respect User Roles**
- **Selection Report (2)**
- **Stop Possibility**
- **Uniform Design**
- **Use default values**
- **Validate your data**
- **Visualize data (2)**

Evaluation

- Datagathering via
 - Teacher observations
 - Weekly questions
 - Questionnaire
- Some interesting findings

Evaluation



- 12 unique groups responded to weekly questions
 - 3 default order, 7 different (+2 other)

Team	Order of P-Forms	Answers to “Why did you apply the P-Forms in this order?”
T1	1,2,4,19,9,17,14	“We found this order quite intuitive.”
T2	1,2,3,4,5,7,19,8,11,9,14,17,13,12,10,15	“We applied the P-Forms in this order because we needed it for the functionalities we made.”
T3	2,3,4,14,5	“We applied them in this order because it was a quite natural order.”
T4	3,2	“We did the P-Form Functionality Description first, because we felt it was necessary to get that one done first. You cannot start a project without knowing the requirements, because if you do, you do not know what you are building and you will deliver an unusable application.”
T5	2,4,19,14,5	“First we must have the database, otherwise we could not work with APEX.” ... “So that there will be a structure in the app”

Evaluation



	\bar{x}	M	σ
Q1: I liked using the P-Forms.	3.21	3.00	1.10
Q2: I would have preferred more classical lectures and exercises.	2.95	3.00	1.23
Q3: I understood the concepts of administration systems.	3.98	4.00	.56
Q4: I would like to have more courses that use P-Forms.	2.79	3.00	1.05
Q5: I learned APEX very well.	3.26	3.00	.86
Q6: I liked to have a well-structured order of tasks as with the default order of the P-Forms.	3.55	4.00	.97
Q7: I liked that I was able to choose my own order of P-Form applications.	3.60	4.00	.99
Q8: The P-Forms helped me to understand what I was doing.	3.17	3.00	1.10
Q9: My new knowledge will help me to implement better administration systems in the future.	3.07	3.00	1.05
Q10: I can use APEX easily for implementing other administration systems.	3.07	3.00	1.07
Q11: The P-Forms were easy to understand.	3.71	4.00	.97
Q12: I missed a book or reader in this course.	2.90	3.00	1.27

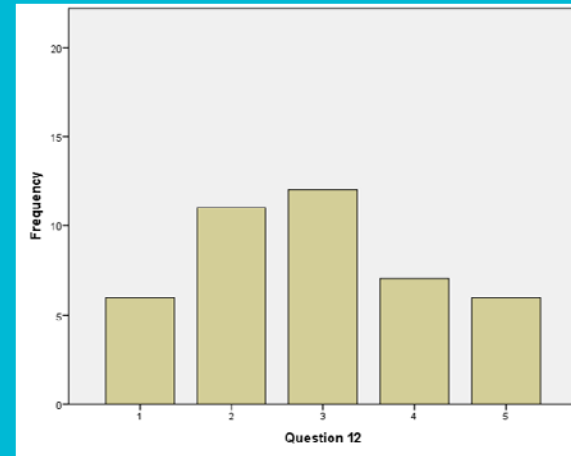
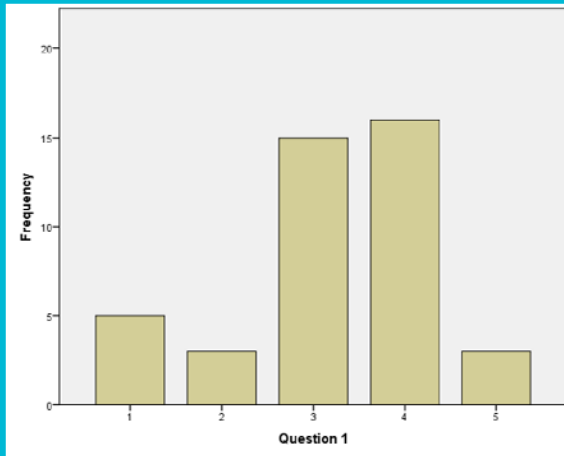
age	n	\bar{x}
≤ 20	20	2.80
21/22	14	3.43
≥ 23	8	3.88

Table 3: "I liked using the P-Forms" - Mean per age

age	n	\bar{x}
≤ 20	20	2.85
21/22	14	3.29
≥ 23	8	3.75

Table 4: "The P-Forms helped me to understand what I was doing" - Mean per age

Evaluation



I liked using the P-Forms I missed a book or reader in this course

- First course with this approach
- Some students used the P-Forms only for checking fulfillment of the grading criteria
- Dependence on pre-education

Evaluation



- Interesting: I did not give this course, but my colleague
- Overall beneficial (even though very critical in the beginning)
 - Just-in-time teaching worked well -> order of covered topics different from original one!
 - Clear grading criteria
- Unhappy with students beginning late
 - > weekly required P-Forms (Continuous Activity)

Summary

- Promising approach
- Not much extra work for adapting existing course
- More “fine-tuning” necessary
 - Partial dependence on pre-education



Thank you!

Mail: christian.koppe@hu.nl

Web: www.koeppe.nl

LinkedIn: <http://www.linkedin.com/pub/christian-köppe/9/52/81>

Twitter: @ckoeppe