
APPENDICES

1.0 PROJECT MANAGEMENT 1

1.1 Plans	2
1.2 Project Status.....	7
1.3 Report Document	12
1.4 Baselines & Reports	13
1.5 Weekly reviews	23
1.6 Keep In Mind	29

2.0 CONTRACTS 31

2.1 Internal Contract	31
2.2 External Contract	32

3.0 MINUTES 34

4.0 STANDARDS 68

5.0 PROTOTYPING 70

5.1 Test and Evaluation.....	71
------------------------------	----

6.0 XP 74

6.1 Survival Kit	74
6.2 Data Dictionary for Database.....	80
6.3 Data Dictionary for Java Classes	83
6.4 Task cards	93
6.5 User Tests Iteration 2 and 3	107
6.6 Problems & Solutions	110

7.0 INSTALLATION GUIDE 113

8.0 SOURCE CODE 115

8.1 Java	
8.2 JSP and HTML	
8.3 SQL	

APPENDICES

1.0 PROJECT MANAGEMENT

- Plans
 - Overall
 - Week plans
 - Document finalizing plan
- Project Status
 - 1 August
 - 1 September
 - 1 October
- Baselines and reports
- Keep In Mind Chart (KIM)

1.1Plans

FIGURE 1. Overall Plan Part 1

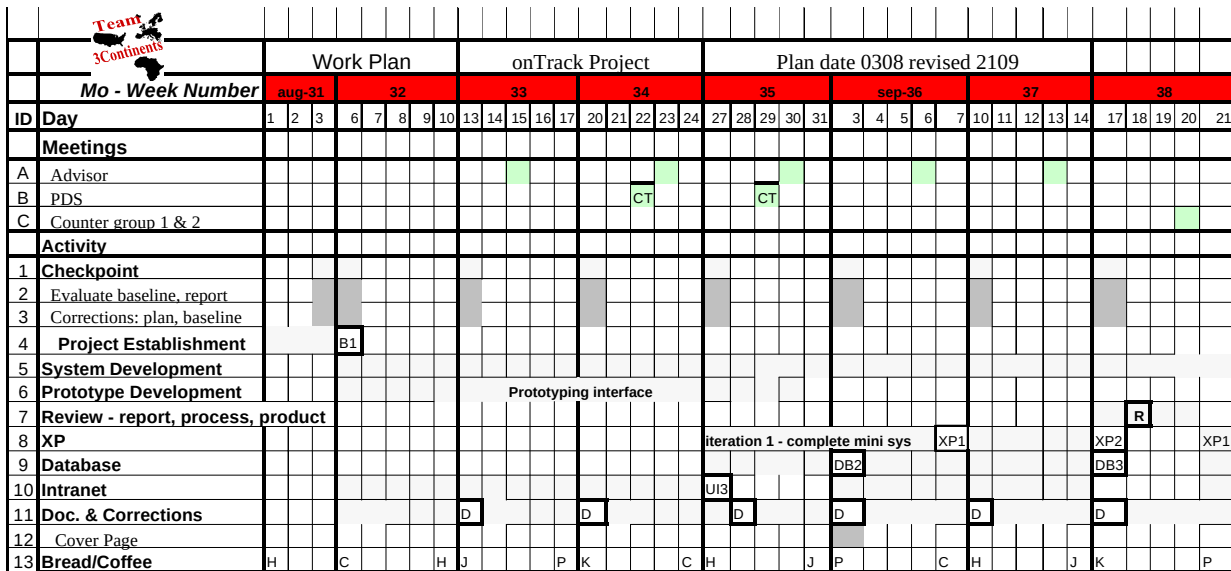
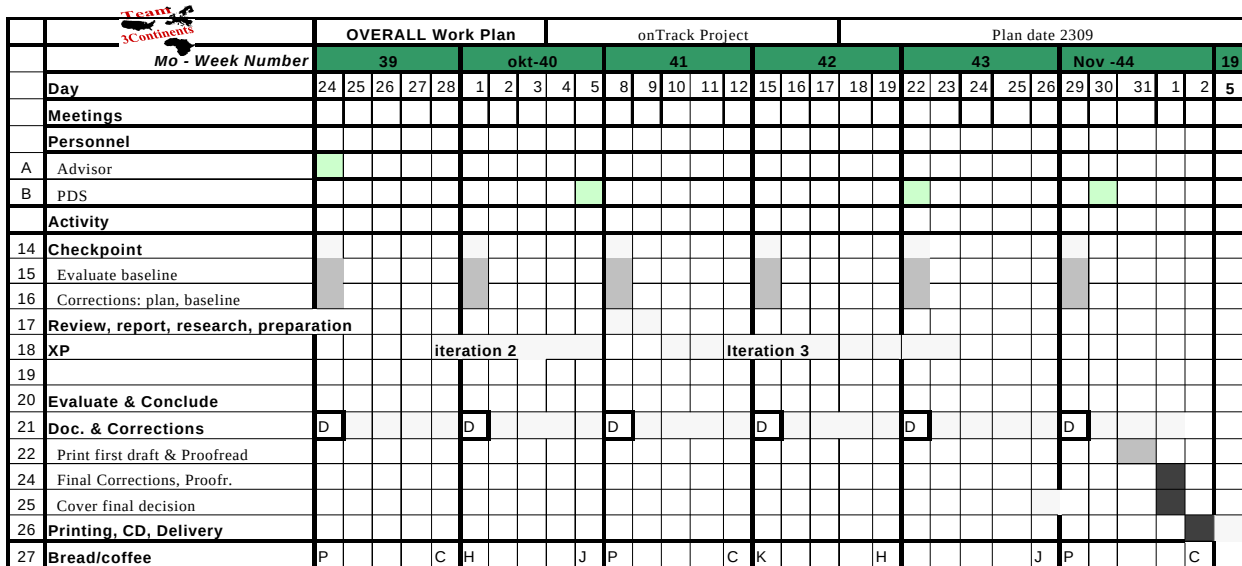


FIGURE 2. Overall Plan Part 2



Plans

FIGURE 3. 3 Week Plan 6th August - 24th August

		3 Week Work Plan					onTrack Project									
		Plan date 0308 - revised 1808														
		Mo - Week Number					aug-32					33				
ID	Day	6	7	8	9	10	13	14	15	16	17	20	21	22	23	24
Meetings																
Docs to advisor																
A	Advisor								1000						1400	
B	PDS					830								830		
C	Other														rhs1100	
Activity																
1	Checkpoint															
2	Evaluate baseline															
3	Corrections: plan, baseline															
4	Project Establishment	1														
5	Sys Meth Research															
6	XP, OO-DB															
7	SQLServer training															
8	Install Tools															
9	Analysis															
10	Initial Requirements	C														
	Requirements 2															
11	E/R															
12	Prototype Plan	J														
14	Database															
	DB conceptual design															
15	DB logical design															
19	Evaluate															
20	Prototype I															
21	Intranet															
22	Navigation															
23	UML Logical View															
24	Write Test															
25	UI Implement															
26	Test															
27	Evaluate															
28	User Documentation															
29	User Testing															
30	User Evaluation															
31	Evaluate & Revise															
32	Doc. & Corrections						D					D				

Plans

FIGURE 4. 3 Week Plan 27th August - 14th September

Team 3Continents		3 Week Work Plan					onTrack Project					Aug 27 - Sept 14				
		Plan date 2608 - revised 0809					version 1.1									
Mo - Week Number		aug-35					36					37				
ID	Day: red = done green not done	27	28	29	30	31	3	4	5	6	7	10	11	12	13	14
Personnel											CJ-free					
Meetings																
Docs to advisor																
A	Advisor				13:00											
B	PDS			830												
C	Seminar .NET (?)													1330		
D	Informal team meet					1300										
Activity																
1	Checkpoint															
2	Evaluate baseline															
3	Corrections: plan, baseline															
4	Prototype I															
5	Intranet UI	UI3														
6	Revise GUI Implement				UI3A											
7	Dev evaluation															
8	Prototype I user eval 2															
9	Technical Prototyping															
10	Test research, trial runs JUNIT						CH		All							
	JDBC										C	C				
11	Forte IDE						CH		All							
12	XP										XP1.0					XP1.1
13	Exploration															
15	Determine tasks				All											
16	Commitment															
17	Tasks- evaluate select								All							
18	Steering															
19	Select code team															
20	Write Unit Test															
21	Testing															
22	Coding									All		All	2x2			
23	Refactoring															
28	User Functional Test															
24	Database						DB2									
25	Implement DDL					JP				JP						
26	Populate w data															
27	SQL DML											All				
29	Doc. & Corrections	D					D					D				
30	Editor's review		P				P					P				
31	Cover Page First Draft															
32	Bread and Coffee	H				J	P				C	H				J

FIGURE 5. 3 Week Plan 17th September - 5th October

		3 Week Work Plan					onTrack Project					Sept 17 - Oct 5				
							version 1.0									
	Mo - Week Number	sep-38					39					oct - 40				
ID	Day: red = done green not done	17	18	19	20	21	24	25	26	27	28	1	2	3	4	5
	Personnel															
	Meetings															
	Docs to advisor															
A	Advisor			1400			1100									
B	PDS					1300										1300
C	Counter Group				900											
D	Kenneth				1245											
E	Informal team meet										1400					
	Activity															
1	Checkpoint															
2	Evaluate baseline						D					D				
3	Corrections: plan, baseline															
12	XP								XP1.1				iter 2			XP2
	Iter 1 complete tasks															
	Iteration 1 user functional test															
	Review iter 1															
16	Commitment															
17	Tasks- write, evaluate select															
18	Steering															
19	Select code team															
20	Write Unit Test															
21	Testing															
22	Coding															
23	Refactoring								iter 1	iter 1	iter 1					
	Integration & Test															
28	User Functional Test															
24	Database	DB3														
27	SQL DML															
29	Doc. & Corrections	D					DM					D				
30	Editor's review	P	P				P					P				
	Project Review															
31	Cover Page Second Draft															
32	Bread and Coffee	K	NOTE changes			J	P				C	H				J

FIGURE 6. 4 Week Plan 8th October - 2nd November

	4 Week Work Plan					onTrack Project					Oct 8 - Nov 5									
	Plan Date 0210					version 1.0														
Mo - Week Number	oct 41					42					43					nov-44				
	8	9	10	11	12	15	16	17	18	19	22	23	24	25	26	29	30	31	1	2
Personnel																				
Meetings																				
Docs to advisor																				
Advisor					1100								???							
Prosa	1000	1300																		
PDS												###					EVAL 9:00			
Informal team meet																			1200	
Activity																				
Checkpoint																				
Evaluate baseline						D					D									
Corrections: plan, baseline																				
XP		XP2						iteration 3							XP3					
Review iter 2																				
Commitment																				
Tasks- write, evaluate select																				
Steering																				
Select code team																				
Write Unit Test																				
Testing																				
Coding																				
Refactoring																				
Integration & Test																				
User Functional Test																				
Database		DB4																		
SQL DML																				
Doc. & Corrections	D					D					D									
Editor's review	P					P					P									
Doc Review																				
Report work																				
Cover Page Second Draft																				
Proofreading & Correction																				
Printing																				
Docs and Code on CD																				
Delivery																				
Bread and Coffee	J				P	K				C	H					J	P			

FIGURE 7. Document Finalizing Plan

ACTIVITY	nov-44							
	24	25	26	29	30	31	1	2
Advisor Meet	1300							
Corrections per advisor								
Complete Task Estimates								
Appendix								
Iteration 3								
Iteration 0-2								
Format Table of Contents								
Figures, Diagrams	completed, new ones? Put in doc							
Problem Def/Answers								
Conclusions complete								
Evaluations complete								
Layout Corrections								
Cover Decision								
Proofreading & Correcting			on disk - home					
Corrections								
Proofreading in evening					Hard Copy	Hard Copy		
Printing								
Docs and Code on CD								
Delivery								

1.2 Project Status

STATUS PR 1/8

EXTERNAL CONTRACT

Understanding of system functionality to date

- Assign applicant to course
- Create course
- Delete course
- Close course for more applicants
- Provide search capabilities
- Prevent doubles of same customer
- Produce printed MS Word documents merged with db data
- Information available to others than Torben
- Transportation instructions
- Locale information for area where course is held
- Hotels
- E-mail participant list

Project Status**SOME KNOWN REQUIREMENTS**

- SQLServer
- Primary keys representing course type
- Document Printing from MS Word
- GUI up to us
- Interface other Plass DB's for customer data

PROJECT ESTABLISHMENT DOCUMENTS

- Risks
- Internal Contract
- Interests
- Contacts
- Positions
- Bios

Missing

Tools with reason for selection
Standards
Plans – Overall, detail
Baselines
Working Methods

PROBLEM DESCRIPTION**Missing**

Problem Delimiting

COUNTER GROUP

No contact

WWW SITE

In progress

COMMON DRIVE AT RHS

ASAP

SERVER 120 DAY TRIAL EDITION

Obtained

MAJOR PLAN DATES

- Aug 3 – Proj estab complete
- Aug 8 – Proj estab to MC
- Prototype 1 complete Aug 14
- Week 38 – Counter group meeting
- Nov 1 – Final proofreading and report correction
- Nov 2 – Printing, CD
- Nov 2 - Delivery

STATUS PR 1 SEPT

**UNDERSTANDING OF
SYSTEM FUNCTIONALITY
TO DATE**

- Register applicants for courses
- Save historical course and participant data
- Print documents
- Control status of applicants
- Interface telephone system (outside project scope)

REQUIREMENTS

Due to prototyping new requirements were uncovered. Summarized these were:

General

1. Deltagere and Course - Must be able to see historic data
2. Printing - don't clip pages
3. Primary Key - If not used for anything, don't show on screen
4. Browser functionality not exploited - add more links
5. RULE 1 - when at least one deltager has been sent FINAL, all new apps are late
6. RULE 2 - no delete course which has participants
7. No other rules = no automation
8. Deltager - JUST ONE SCREEN For rediger and create
9. Fixed Menu - left side all screens
10. Separate search and edit
11. Possibly color code Course Name in Oversight over kommende kurser according to status
12. 2 letters for 2 situations and 1 letter for 1 situation
13. Revise field order
14. Watch consistency of layout

SCREENS

Kursus Admin

- Menu left
- Start screen opens with Oversigt overkommende kurser
- Links from Oversight to participant, course, firm
- Use Label Breve, not Formbreve

Kursus

- Explain date format next to text field (TF)
- Smaller TF for instructor
- Edit and create - one screen

Kursus type

² Course code = 5-8 char

Deltagere

- Move C5 field. This is a required field
- Use Kunde nr not name

Project Status

- Company name is search criteria

Firm

- Need person title field
- 2 address lines

Additionally

- ASP
- SQL
- minimal brug af session og cookie
- HTML 4 eller XHTML
- JMAIL (bruges til email)

PROTOTYPE I

The GUI prototype is accepted and complete , minus screen for printing

REPORT DOCUMENT

Complete:

- Project Establishment - need to revise Prob Descrip and delimit
- Requirements - (will be improved)
- Prototyping

In progress:

1. Systems Development
2. Test
3. Database
4. Tools
5. Server side code

TECHNICAL PROTOTYPING

Done

- Front Page
- GoLive

MAJOR PLAN DATES

Aug 3 – Proj estab complete

1. Aug 8 – Proj estab to MC
2. Aug 22 - User prototype evaluation
3. Aug 29 - User follow-up prototype evaluation
4. Prototype 1 complete Aug 29
5. XP Iteration 1 complete Sept 14
6. XP Iteration 2 complete
7. XP Iteration 3 complete
8. Week 38 – Counter group meeting

Project Status

9. Week 38 - Internal Project review
10. Nov 1 – Final proofreading and report correction
11. Nov 2 – Printing, CD
12. Nov 2 - Delivery

PROJECT TEAM

OK

- Quickly through phases Creation and Organization
- Cooperation steadily building
- No conflicts, some misunderstandings due to uncertainty and language
- Constructive debate, creative input, working for results

Needs improvement

- Higher degree of independent initiative
- Communication
- Assignments

EXAMPLE OF RESOURCE EXPENDITURE

This table was created to provide us with an idea of the economy of software development.

TABLE 1. Resource use August

		TOTAL
Manpower M	4	
Days D	23	
Approx Daily Hours at school effective H	5	
Total Man hours MxDxH	460	
Hourly cost (Dkk 26000/mo 142 hours)	175	
Sub total	Dkk 80.810	
Total Project Management & assignments daily hours	4	
total man hours = 23 x 4	92	
sub total	16.162,00	
Total cost		Dkk 96.972,00
Total Cost based on monthly salary		Dkk 104.000,00

STATUS UPDATE PR 1 OCT. 2001

CHANGES Since Sept 1

REQUIREMENTS

- Ability to create new types of letters and associate these with a new status type

Report Document

- Ability to edit the existing letters
- XP1**
- Completed. First functionality implemented as a skeleton system
 - Create database tuples of all necessary entities except participants
 - Print one letter type out - some required data fields are not yet included in this letter

1.3 Report Document

MID-WAY REPORT

- Foreword
- Introduction
- TOC
- Project Establishment
- Initial Requirements
- Client/Server architecture
- Server Side Scripting
- Methods & Tools
 - Development method
 - Tools
 - Technical Prototyping
- Iteration 0
 - Prototyping
 - User Interface
 - Test
 - Database
- References - **Not updated recently!!!**

Appendix

- Data Dictionary
- Project Management
 - 13.** Plans
 - 14.** Weekly reviews
 - 15.** Baselines and reports

IN PROGRESS

1. XP1 to be completed by Oct 9
2. XP2
3. Data Dictionary
4. User Manual

MAJOR PLAN DATES

1. Aug 3 – Proj estab complete

Baselines & Reports

2. Aug 8 – Proj estab to MC
3. Aug 22 - User prototype evaluation
4. Aug 29 - User follow-up prototype evaluation
5. Iteration 0 - user interface complete Aug 29
6. XP Iteration 1, skeleton system complete Sept 14
7. Sept 19 - Internal Project review
8. Sept 20 – Counter group (both T and K groups)
9. **XP Iteration 2 - Overview, participant, error handling complete October 5**
10. **XP Iteration 3 complete October 22**
11. **PDS evaluation October 30**
12. Nov 1 – Final proofreading and report correction
13. Nov 2 – Printing, CD
14. Nov 2 - Delivery

PROJECT TEAM

- XP pair programming helps maintain good working relationships
- Informal meeting for socializing outside the project working environment also appears to be help the team
- The ambition level since inception of project is as high as ever and the results to date reflect this. It can be said that as we go the level of ambition grows!! Still we are aware of the limits the project places upon us and are realistic in our goals.
- Despite great difficulties at times, everyone pushes on to solve problems and it seems we thrive on problems occurring and finding solutions to them. Not to say though that at our worst moments of frustration we hid it. Frustration was obvious, but it lasted only briefly and does not affect the outcome.

EXPENDITURE

Dkk 192.000 (August * 2)

1.4 Baselines & Reports

Baselines are used as a means of checking project state at a specific checkpoint in time. A baseline evaluation report is written for each baseline to document the baseline evaluation and use for revision of baselines and plans.

PROJECT ESTABLISHMENT BASELINE

DESCRIPTION

- Project establishment completed.
- The team has been assembled several times providing the socialization needed at the inception phase. The meetings have been to discuss project and personal goals.
- Advisor meeting held to establish the relationship with him and inform of our initial plans.

- Meetings have been held with the sponsor organization to determine the nature of the project and in what form we will be able to work cooperatively with each other
- We have described PDS
- Produced a problem description and delimiting
- Done an evaluation and selection of system development method
- Interests and their relation to the project have been determined
- Risks determined, listed and prioritized according to a point system
- Tool evaluation and selection completed
- An overall plan has been created holding major deadlines and checkpoint

INTERMEDIATE PRODUCTS

- The project establishment document is the main product of this baseline
- Report introduction which includes the problem description and delimiting
- Overall project plan
- An internal and external contract

CRITERIA FOR EVALUATION

- Project establishment according to Professional Systems Development chapter 2
- Situation and Action - chapter 4
- Project establishment - chapter 5, Planning

PROCEDURE FOR EVALUATION

The project establishment document and report introduction are read through as a group and checked for completeness, systematic ordering of sections, typographical errors, inconsistencies.

We will evaluate the establishment phase asking ourselves the question - Are we now better prepared to deal with the project? If the answer is yes, we've done a good job. If the answer is no, we must rethink any part that may provoke a negative answer.

Corrections to the documents will be made as needed.

Approval by the project advisor will be considered acceptance of the project establishment.

**PROJECT ESTABLISHMENT
BASELINE REPORT AUG. 8****RESULTS:**

- Quantative - Project estab document although not taking every possible aspect of project establishment into consideration is a complete document.
- Qualitative - OK, approved by advisor and recommend to place in report versus appendix
- Deviations & Explanations - Adjust Company profile, more risks, improve standards, have separate chapter on tools.

Baselines & Reports

PROJECT SITUATION: **General** - very early stage.

In relation to timeplan

No deviation

Approaching important conditions

Result of XP research.

Availability of requisitioned software from school.

Action:

Next Checkpoint report:

20-08 document, first general baseline.

DATABASE DESIGN BASELINE PART 1

DESCRIPTION

This baseline constitutes the conceptual and logical design of the database.

**INTERMEDIATE PRODUCTS
AND PREREQUISITES**

- **Conceptual design**
 - Entities and relationships found
 - ER diagram drawn
 - Attributes and domains found
 - Primary and candidate keys found
 - Attributes documented
- **Logical design**
 - Undesired feature removed (multi value attribs, M-N, etc.)
 - Relations and foreign key attributes documented
 - Integrity constraints achieved
 - Normalization carried out

**CRITERIA FOR
EVALUATION**

Conceptual and logical design approved by advisor and PDS

(Connolly, chapters 7 & 8)

**PROCEDURES FOR
EVALUATION**

- Submission of database design - conceptual and logical - to advisor and PDS for review
- Internal review according to feedback from interests

DATABASE DESIGN BASELINE PART 2

DESCRIPTION

This baseline constitutes the physical design and the implementation of the database.

**INTERMEDIATE PRODUCTS
AND PREREQUISITES**

Physical design document, SQL script for creating tables and references amongst them. The tables are designed in the simplest possible way, not taking special consideration for futurization of the system.

**CRITERIA FOR
EVALUATION**

SQL script version 1 done, CRUD operations on the database possible through query tools.

(Connolly chapter 9)

**PROCEDURES FOR
EVALUATION**

Internal evaluation

**DATABASE DESIGN
BASELINE REPORT PART 2
SEPT 3****RESULTS:**

- Quantative - Conceptual, logical design. E/R diagram, db doc started. DB Physical not created. Scripts written
- Qualitative - OK. Advisor had suggestions. We may select to use the suggestion to futurize by adding tables not desired by PDS, but not to add extra tables holding only primary keys.
- In relation to plan - DB not created. As this takes little time it is not considered a problem.
- Deviations & Explanations -

PROJECT SITUATION

- General - At this point the project is progressing in the planned direction. The only setback was need to rework GUI. However some time was planned for for this activity as we are doing evolutionary prototyping. The feedback is considered very useful and although costs some time now, we feel it would have been devastating had this occurred at a point in time much later in the project, thereby the benefits of prototyping are achieved.

In relation to timeplan - on time

Approaching important conditions

User approval of GUI, XP start. Use of JUnit for testing if using Java for coding.

Action: Review design. Implement db

Next Checkpoint report: Sept 7 XP1

UI3 BASELINE**DESCRIPTION**

This baseline is concerned with the graphical user interface. It controls the layout as well as the documentation necessary to understand the user interface.intermediate products and prerequisites.

Baselines & Reports

INTERMEDIATE PRODUCTS	• Activity and navigation diagrams • xhtml codes • Test plan and test report
CRITERIA FOR EVALUATION	• Sequence of screens should match navigation map, which should have a flat hierarchical set-up to minimize mouse clicks • html codes should use xhtml syntax (html 4.0) and support IE browsers, as this is what PDS uses • Content of each screen should contain all required fields of input data and their manipulation buttons
PROCEDURES FOR EVALUATION	• Internal group test • External evaluation at PDS • Validation of xhtml codes with Tidy
UI3 BASELINE REPORT	
RESULTS	• Quantative - Navigation & Activity diagrams, Physical screens • Qualitative - Need to validate html code • In relation to plan - 1 week delay. Need new feedback from User org • Deviations & Explanations - After first evaluation, we needed to rework. This took only a day, but we were not able to meet with PDS again until a week later. However we were able to continue as we sent PDS screen shots for a first evaluation and feedback before actually demonstrating the updated Prototype.
PROJECT SITUATION	• General - We will need to validate the XHTML after the screens are approved. • In relation to timeplan - The GUI is 1 week behind plan, however this has not caused delay as other work could be continued and the GUI is not needed until that work is complete. • Approaching important conditions - Approval on 2nd User Evaluation. Start of XP phase Action: Reworking interface according to feedback from first user evaluation. New user evaluation scheduled. Next Checkpoint report: Aug 30 UI3A , Sept 7 XP1 baseline
XP1.0 BASELINE	
DESCRIPTION	This baseline will control the first part of the first iteration. It will involve all of the products as well as the processes. The starting point of the processes will be XP
INTERMEDIATE PRODUCTS	• Documentation of involved basic practices of XP. • Task cards with their function list. • Implementation of 2-task cards with the following result.

Baselines & Reports

	• Test writing, which means codes to be tested for the task. • java beans to control the functionality of the task. • JSP pages to generate the XHTML files (where applicable). • Connection of the DB.
CRITERIA FOR EVALUATION	• All codes should satisfy the code standards in the project establishment. • Test codes should contain all required methods to test the involved task. • Minimize the use of Sessions and cookies to fulfill the user requirement.
PROCEDURES OF EVALUATION	• JUnit for automatic test. • Internal testing by the group. • XP practices (where applicable).
XP 1.0 BASELINE REPORT SEPT. 7	
RESULTS:	• Quantative - None of the task cards have been fully implemented. We have java beans and tests. No connection to DB and no JSP. The XP basic documentation is not written. • Qualitative - The achieved results fulfill the baseline criteria. • In relation to plan - We are about 2 days behind. • Deviations & Explanations - Difficulties with technical prototyping - in particular Classpath settings and FORTE4j 2.0 IDE.
PROJECT SITUATION:	General Each day is a learning experience improving our abilities to make progress. However we did not place enough weight on the difficulties we would incur during technical prototyping.
IN RELATION TO TIMEPLAN	About 2 days behind
APPROACHING IMPORTANT CONDITIONS	XP2 and the user functional test which requires that we have a full functioning system in miniature, that is each element of the system is present in the first version, although the functionality is not built in full.
ACTION	Change of plan. Overtime may be necessary.
NEXT CHECKPOINT REPORT	Sept 14 - XP1.1

**XP ITERATION 1.1
BASELINE****DESCRIPTION**

This baseline will control the second part of the first iteration. It will involve all of the products as well as the processes. The starting point of the processes will be XP

INTERMEDIATE PRODUCTS

- Documentation of involved basic practices of XP
- Task cards with their function list
- Implementation of all task cards - Creates except Participant, Print with the following result:
Test writing, which means codes to be tested for the task
java beans to control the functionality of the task
JSP pages to generate the XHTML files (where applicable)
- Connection of the DB

**CRITERIA FOR
EVALUATION**

- All codes should satisfy the code standards in the project establishment
- Test codes should contain all required methods to test the involved task
- Minimize the use of Sessions and cookies to fulfill the user requirement
- User approval

**PROCEDURES OF
EVALUATION**

- JUnit for automatic test
- Internal testing by the group
- XP practices (where applicable)
- User functional test

**XP 1.1 BASELINE REPORT
SEPT. 26****RESULTS**

- Quantative
Code: Course, Course_Type, Firm, Location, Person, Print in JSP and Java.
Participant moved to XP2. Not enough tests written.
Document: XP intro and XP1 is documented in the main but not completed. Completed task cards are handwritten, not in electronic form.
- Qualitative: User approval after functional test, which was not a written prepared test, but more like a technical walkthrough.. We have used the XP practices of pair programming, writing task cards, evaluating time needed for tasks. We did not concentrate on JUnit testing in the XP1.1 baseline. Our own test at the last moment before the user evaluation revealed many errors due to changes and we were able to make a speedy recovery but with no time to spare before user evaluation.
- Deviations & Explanations - Our coding is rusty after summer vacation and a long school period without much coding. We basically have no idea how long time it takes to complete a unit as determined by task cards. Therefore our evaluation of time usage was way off. In this first iteration, we planned to do far more than possible and this put us under too much pressure to deliver by the time of functional test. Additionally we ran into many start-up problems with tools, which points to the need for an extended technical prototyping period when using unfamiliar or tools new on

the market. Another possibility could be to reduce the risk by sticking with familiar tools and changing only when there truly is a need for change.

Our misjudged evaluation caused us to deviate from XP with the goal of having a product to present at the user evaluation. Therefore did not do refactoring during the iteration.

PROJECT SITUATION**General****In relation to timeplan**

Approaching important conditions: XP2 user functional test Oct. 5

Action: Completed Task cards by 27/9. An intermediate phase between XP1.1 and XP2 will be used to refactor and complete tasks from XP iteration 1.

Next Checkpoint report: Oct 8

**DATABASE DESIGN 3
BASELINE****DESCRIPTION**

This baseline constitutes the connection between SQL Server and Java.

**INTERMEDIATE PRODUCTS
AND PREREQUISITES**

There has been established a connection between our Java code and the SQL Server, through the use of JTurbo JDBC driver. Technical prototype document, explaining the steps hereof.

**CRITERIA FOR
EVALUATION**

Connection established and functioning between Java and SQL Server.

**PROCEDURES FOR
EVALUATION**

Internal evaluation and test of database via java classes.

**DB 3 SEPT 17 BASELINE
REPORT****RESULTS**

- Quantative - Java code to connect to DB via JDBC. Data Dictionary. DB created. Can insert and retrieve from DB. Some tables need test data.
- Qualitative - Java code mixed in all classes instead of own DB class = bad design, needs refactoring.
- Deviations & Explanations - Authentication between Windows and SQL server was biggest problem

PROJECT SITUATION:

- **General** - Moving forward

Baselines & Reports

- **In relation to timeplan** - OK
- **Approaching important conditions** - See XP1.1

Action: Populate tables with data

Next Checkpoint report: Oct 5, XP2.0

BASELINE DOC MIDWAY

DESCRIPTION

This baseline is for the mid-way state of the project report.

INTERMEDIATE PRODUCTS

Following documents are finalized or up to date in relation to project phases:

- Intro
- Foreword
- Proj Estab
- Prototyping Iteration 0
 - User Interface
 - Testing
- Initial requirements
- Tools
- DB
- Server scripting
- Report is in a structured form

CRITERIA FOR EVALUATION

- Counter group feedback
- Advisor feedback
- Own standards

PROCEDURE FOR EVALUATION

- Internal Review
- Editor proofreading
- Reviews with counter groups
- Advisor Feedback

MIDWAY DOC BASELINE REPORT SEPT. 25

RESULTS:

Quantative - All docs as expected except task cards, figures and diagrams for C/S.

Qualitative -

- High according to reviews and corrections and structuring after reviews
- Project establishment needs a conclusion

Deviations & Explanations - none

PROJECT SITUATION:

General

- Progress is very good. Many of the initial difficulties uncovered during technical prototyping are behind us. Our coding is gaining speed as we get into the train of thought again after a summer without programming. Our understanding of the XP process is improving. We have learned much in the last few weeks time about organizing our work and this should benefit us in the next XP iteration.

In relation to timeplan: OK

Approaching important conditions: XP2 user functional test Oct 5

Action: task cards completed for XP1 2609. UML logical completed 2609

Next Checkpoint report: Oct 8 XP2

1.5 Weekly reviews

TABLE 2. Weekly Evaluation Week 31

Element	Comments
Agenda	If possible, 2 days before
Discussion	NC
Presentations	NC
Results	everyone felt we had gone far in the first 3 days and gotten many things in place
PM	second portion coffee too strong. Solved with ice water in mug
Suggestions in general	Docs to Carsten by thursday. He sends on to Paul after checking file names and backing up. If a document is specifically for editing we write in subject of mail to Carsten: For Editing.

TABLE 3. Weekly Evaluation Week 32

Element	Comments
Agenda	pm prioritize Individuals inform time needed for a presentation
Discussion	Off track on Thursday compared to agenda. Consequence was less time for Paul on XP
Presentations	Paul - XP, excellent presentation which helped us all gain considerable knowledge
Results	According to plan Improved XP Understanding Requirements Updated PDS - XP
PM	Organization is necessary and useful
Individual Work	OK
Team	Functioning without conflicts.
Suggestions in general	We need to keep in mind how to incorporate the information in in report. use KIM is one way and Essentials for Doc KIM
Website	Improvements made - more to come

TABLE 4. Weekly Evaluation Week 33

Element	Comments
Agenda	OK - good with several days agenda in advance Good idea of alternating subjects - gives time to read between days. Good we haven't stopped at middle but at points where it was acceptable to break off an activity.
Presentations	Jama prototyping: Presented a helpful view that stimulated discussion
Results	According to plan.
Individual Work	OK
Team	Coming together
Website	OK
Advisor meeting	Positive with good critique and atmosphere
Carsten's eyeglasses	Still not changed

TABLE 5. Weekly Evaluation Week 34

Element	Comments
Agenda	Paul complained about the upside-down agendas, but after an explanation, it became clear they were printed correctly, he just had to turn them 180 degrees
Discussion	The week presented lots of opportunity for discussion on the UI design, tools, prototyping and in particular the internal and user tests of the evaluations. We agreed that there was great value in testing, and prototyping has saved us from problems much further down the road, because using OO we would have uncovered the problems we did with the UI before it was too late. It also seems though that prototyping is a somewhat vague in what activities are part of the method and can lead to some consideration of looking for suitable combinations of methods. The week also covered E/R data analysis and database design. These too were subjects for many good discussions on the relationships, business rules and other considerations for the database.
Results	We are proceeding according to plan with a slight deviation due to the need to rework the GUI after the PDS evaluation. It will be re-evaluated the following work, but our work continues. The report document is well underway. Our database design is almost in place and should be implemented next week.
Individual Work	Ok
Team	No major problems. Some difficulty in maintaining work discipline such as all use tool agreed upon and standards.

TABLE 5. Weekly Evaluation Week 34

Element	Comments
Advisor meet	Very positive. We found the advisor to have a good attitude and comes with useful advice. He is open minded to the new ideas we are working with.
Carsten's eyeglasses	Same tape holding them together

TABLE 6. Weekly Evaluation Week 35

Element	Comments
Agenda	Difficult to keep times allocated for activities, but times are only indicative timeframes. Without these a discussion may be open forever.
Discussion	Another week with lots of good discussion. Good meaning we have put a lot of consideration into everything we do. Now the trick is to document it all.
Results	We feel satisfied with the weeks work. GUI approved by PDS. We are moving according to plan with no setbacks this week.
Team	Functioning well. No major problem. All still satisfied with PM.
Website	PM mentioned that it didn't seem the web-site was used as a communication medium as it is intended.
Advisor meeting	He didn't much from us so the meeting was a little thin in value. We realize he seems to point us in some direction each time via tips and hints on which some we will pick up on and others not. This time he started a discussion which re-opened the E/R diagram although after some time we did not feel the suggestion was applicable. Another point that has been repeated is the Spiral Model and we will try to work this into our report.
Carsten's eyeglasses	Nothing new

TABLE 7. Weekly Evaluation Week 36

Element	Comments
Results	Disappointing. We expected to have some tasks completed and ready for pair programming by following week. Our many difficulties with Classpaths stopped this
PM	Combine E-mails if possible. PM - OK if can be sure E-mail is checked nightly at certain time.
Individual Work	PM needs answers to when material will be ready. Ignoring requests is annoying to both PM and person being queried
Team	OK

TABLE 7. Weekly Evaluation Week 36

Element	Comments
Advisor	Not much this time. We had delivered little too. His doubts were calmed as far as if are approach uncovered all entities
Website	Minutes and baselines missing Need latest references too

TABLE 8. Weekly Evaluation Week 37

Element	Comments
Discussion	Often we discuss and don't make decision when they are needed. Waste of time. Must make final date if decision delayed
Results	Need more thorough testing. By Friday we were up-to-date on everything but code. A mutual wish between PDS and team to delay user evaluation held up completion of XP1 one week. Much time was lost due to the terrorist attack on WTC that got the world to stand still in shock.
PM	Still too many e-mails!
Individual Work	JDBC to work between JSP/Java and SQL Server solved for Monday. This was a week later than planned.
Team	All fight for solutions despite the problems
XP	We have tried a lot of it, but not systematic enough, just getting feet wet
Website	Paul says: What Web site??!!

TABLE 9. Weekly Evaluation Week 38

Element	Comments
Advisor	Theme of meeting - report structure. Badly in need. New meeting for Monday about content. This helped us focus on structure point until Monday
Discussion	Still going strong. Generally discussions that bring a lot of questions to surface, but we sometimes forget decisions where necessary = wasted time for discussion. It's not always possible to decide at moment - this situation should be recorded on KIM
PDS User test	No written by user. More a technical walkthrough.
Results	We are satisfied. Software is moving forward and although code is not refactored and design is not OK, it is simple according to XP. We'll get it all by end of project
PM	OK work
Individual Work	Sometimes deadlines are not being kept. Requests for documents are sometimes ignored. PM would prefer to receive a date when something is definitely ready, than no answers.

TABLE 9. Weekly Evaluation Week 38

Element	Comments
Team	Productive. Solves problems. Need to listen more to partner when pair programming!!
Reviews	MaRT group - did a great job. Lots of good points which we focused on and corrected Kenneth - Unfortunately we did not get through his review of our work. We had a tight schedule that day and felt it was more important for him to get our feedback, but we got lots from him and also read his points he made on hardcopy.

TABLE 10. Weekly Evaluation Week 39

Element	Comments	Next week KIM
Advisor	We can have some doubt about his opinion of our project as it has many new facets.	
Agenda	Put a weekly Calendar on bottom	change agenda template
Discussion		
Document	Good to get structured. Need to remember to work on - as much time is planned for coding	FONTS, TEMPLATE
XP	Good with refactoring = improvement in readability and design. Screens corrected Testing - seems difficult to determine what to test and get the tests written rather than just writing actual code	We need to work on testing. Test on Task Card

TABLE 11. Weekly Evaluation Week 40

Element	Comments	Next week KIM
Results	We reached most of the expected results for XP2 although the Task overview was only partially complete.	Complete Overview as part of XP3 Complete unfinished parts of XP iterations such as printing in XP3
Problems	There were many technical problems. The worse was the expiration of the trial version of JTurbo- our driver for the the DB connection! Had we have realized at an earlier point that writing out exceptions was a HUGE help, the problem would have been uncovered and solved fast instead of the approx. 2 hours that were wasted.	Make sure to capture exception messages in output.

TABLE 11. Weekly Evaluation Week 40

Element	Comments	Next week KIM
Team	We have to learn to trust more in our capabilities. This point is determined by the fact that despite some code might run at one moment, we are doubt the next if there is a break-down, which often is caused by the unstable IDE Forte. The result is unnecessary changes in code and much lost time.	List Problems and Solutions as a reference
XP	Testing was very weak in XP2. Refactoring was strong. Writing task cards and estimating was strong - in particular since the scope of the iteration was properly judged.	Concentrate more on testing. Make it a required prerequisite to coding. If tests aren't written, explain why.

TABLE 12. Weekly Evaluation Week 41

Element	Comments	Next week KIM
Advisor	Good meeting. Very useful comments on a missing part for our XP Iterations - something that tells what the inputs and expected outcomes are for the iteration, in other words, What the hell are we doing!	Add to document points on What it is each iteration is about
Agenda	Calendar is improvement	
Code	Refactored with a Super Class eliminated lots of redundant code	
Document	Brought up to date with Iteration 1 and 2. Good that we got some document work done this week as it was become to distant an activity.	
PM	Should bring more home baked cakes Might have been useful to have posted a wall chart tracking how many and what tests were done to motivate testing	
Team	Very well functioning and fun group to work with, that achieves results.	
XP	Could be doing a better job with testing	

TABLE 13. Weekly Evaluation Week 42

Element	Comments	KIM
Code	Carsten says “OK OK OK” Good testing this week	
Document	Good that we used time this week on updating and putting together Sent to MC as one report file Just a few parts missing for what has been submitted like the new requirements for iteration 0	
Plan	Apparently not a very useful tool - nobody looks at it despite the fact it is E-mailed and hanging on workroom wall. Only point that gets noticed is anything that applies precisely to one individual - mostly on our plan this is the Bread Plan!!!	
Team	We are at a stage where things are functioning so well, that we are now really ready to continue with project - just when we are about to close down!	
XP	Testing accomplished	

1.6 Keep In Mind

The Keep In Mind (KIM) table follows. It was used as a wall chart in our work room and was quite helpful. Some points were not OK at the time of inclusion in the report.

Table 14: KIM

What	Who	Deadline	Status
Get Lotus Approach from PDS	C	ASAP	OK
DB access from word	C	ASAP	OK
Coding Standards	C	ASAP	OK
Junit	HC	27/8	OK 07-09
html tools	J	17-8	OK
db baseline1	C	21-8	OK
UI1 baseline	J	24-08	OK
Test/Proto	H	07-09	OK
Sys Dev meth	P	30-09	
Revise Proj Estab	H	07-09	OK

Table 14: KIM

What	Who	Deadline	Status
Doc Baseline	P		OK
JDBC Driver	C	31-08	OK per 07-09
List of final docs	C	02-09	OK
Minutes in PDF to H weekly	C,J,P	continuous	
Update PC's	C	01-09	OK
Central DB	C	24-09	OK
Central Code Pool	C	24-09	OK
Update Data Dictionary	C	continuous	OK per 30-09
Refactor Iteration 1	All	28-09	OK
Task Cards i E form	P	01-10	OK
html template	J	28-09	OK
User Guide	J	24-10	OK
Installation DOC	C	immediate	!OK
Project Estab Conclusion	H	immediate	OK
All chapters intro& conclusion			
TOC - how to format	P,C		OK
Arrange printing w Sanne		23 - 10	!OK
Remove IIS,ServletExect from tools	C	Immediate	

2.0 CONTRACTS

2.1 Internal Contract

PURPOSE

The internal contract provides rules and guidelines for the project team concerning working methods, socialization, commitments and conflict solving.

Team members agree this project will receive 100% of their attention and capabilities for the purpose of producing the highest quality products and an excellent experience in regard to the process.

Working hours

- Mon. - Thurs: 8:30 - 15:00
- Fri. 8:30 - 14:00

Place of work

Roskilde Business College. At PDS as needed and possible.

Informal meeting

At end of each month or phase we will meet outside school to discuss progress of the project. We will use this as means both of ensuring the social relations of the team members are positive and to look at project from a different perspective than the formal working atmosphere provides.

Decisions

It is an objective to reach consensus in the group after knowledgeable discussion and argumentation for viewpoints. Mediocre compromise should be avoided. We will listen with open minds to the opinion of others and be willing to argue for their viewpoints if the argumentation is valid.

Delivery of work

The initial requirement is that all work be delivered as planned. Should this not be possible, a request must be made at the latest 25 hours prior to scheduled delivery to the project manager for an extension. Material must be delivered as agreed on extension date. This situation will not be accepted more than twice by the team and will require a team discussion as to what action to take in respect to the person in question.

Minutes

A record will be made of meetings within the group and with PDS and our advisor.

External Contract

Internal Communication:

- Phone - latest 22:00, Harvey until 23:00.
- E-mail - latest 22:00

Leave home: Carsten, 8:00; Harvey, 7:00; Jama, 7:30; Paul, 7:00

Conflicts:

Internal Conflicts - We will use knowledgeable discussion as the first method to solve a conflict. If the conflict is unresolved, the project manager may make a decision to break an impasse. If the project manager finds he is not able to make a proper decision for whatever reason, the project advisor will be consulted and his word will be final.

External Conflicts - This refers to a dispute between the project team and the project sponsor. In this event we refer to the terms in the external contract.

2.2 External Contract

PROJECT CONTRACT

Contractor:

Plass Data Systems A/S (PDS), Ved Fauergaarden, Tåstrup Møllevvej 12A, 4300 Holbæk

Project Group:

3Continents(3Con)

TERMS OF CONTRACT

Contract Period: Aug 1, 2001 - Nov. 30, 2001

Final Product Delivery: Nov. 30, 2001

Deliverables: Written report. Prototype software on CD.

COPYRIGHTS:

3Con retains rights to resultant software. PDS is granted the right to use or further develop the software without incurring any costs provided the usage is within PDS. PDS may not sell or provide external organizations with the software without the permission of 3Con.

PAYMENT:

During the project period 3Con will not receive fees. Should PDS request 3Con to continue developing the product after the project period, a renegotiation of the contract would be required.

LIABILITY:

3Con is in no way judicially liable for any result, direct or indirect due to PDS's use of the delivered prototype.

External Contract

CONTACT PERSON:	PDS has assigned Preben Klavsen and/or Benny Bech as contact person(s) to 3Con. Harvey Shaw for 3Con.
MEETINGS/ COMMUNICATION:	PDS agrees to meet with 3Con as necessary, maximum once per week, though in exceptional instances if E-mail or phone are not sufficient to solve a problem an extraordinary meeting be held. Normal meetings to be scheduled 1 week in advance. In addition prototype testing and evaluation will require meetings which may coincide with any informative meetings mentioned previously. 3Con will inform PDS of the project's progress at least once weekly by e-mail, telephone or via the 3Con Website, www.rhs.dk/dmu_gruppe15.
FACILITIES:	PDS has agreed to provide 3Con with the use of minimum one PC with necessary software and a workspace. 3Con will request this at least 2 days prior to need. Handbooks and other literature are available to the 3Con group at PDS.
COOPERATION	PDS and 3Con will work together in such a manner as to solve all difficulties regarding the product or the process in order to ensure the resultant deliverables are of the highest quality and to the satisfaction of both PDS and 3Con.
PDS LOGO	3Con has permission to use the PDS Logo for the report and on the 3Con website. It will be stated in both report and on website that the logo is used with Plass permission and that the products are not in any way PDS A/S products.
PREMATURE TERMINATION OF CONTRACT	PDS may prematurely terminate this contract. To do so requires the following: • A formal meeting with 3Con to discuss and if possible resolve situation. PDS must call this meeting with one week's notice. • A written explanation of reason for termination including date for termination - no earlier than date of meeting. Should this situation arise 3Con retains the right to continue to work on the project until Nov. 30, 2001.

For Plass Data Systems A/S:

Date/Place: _____ Benny Bech _____

For 3Con:

Date/Place: _____ Harvey Shaw _____

3.0 MINUTES

MINUTES 2001-08-01

OVERHEADS BY HS

- Project status
- Project structure
- Organisation
- Working methods

Not much to write on business part, as this project is an internal development.

POINTS

MAJOR

- What systems development methodology is best suited for a relational database oriented project or is it a combination of methods best used?
- Prototyping

MINOR

- How will the database server MS SQLServer affect PDS course administration as compared to the Lotus Approach system now in use?
- Extreme programming

Harvey's exercise about "3 things that will give us 13":

- 1 Professionalism of report, consistency Innovative thinking
- 2 Own methods/taking best of others Thoroughness
- 3 Consistency in presentation Comprehensive preparation for presentation

Database design: Analysis vs given

TOOLS

Carsten will make tools list ASAP

SYSTEM DEVELOPMENT

THOUGHTS

Prototyping by JD

- Dynamic high-level language, e.g. Java
- Database programming: 4GL and database
- Throw-away prototyping (= revolutionary) is done to gain experience and to help plan future
- Evolutionary prototyping

XP + prototyping by PT

External Contract

- Make tests for classes, then program
- Delphi front-end for db is waste of time because Delphi is not web-capable. Are there any classes at all? Will analysis benefit anything? Intranet + server/db = 2 distinct projects?
- Any prototyping improves working spirit, because developers see a product! A goal to make something that works, compared to last project where we coded like hell the last two weeks.
- Combine prototyping to show what we have got using XP principles.
- First prototype is duplicating what they have in their system now.
- We should talk to Erwin as he probably is the most experienced system development person at school - especially about prototyping, as this seems like it started in the 70' - 80's. Where, when and how do we start with prototyping?
- Evolutionary for known requirements, revolutionary for unknown/unclear requirements.

WE NEED TO GET THE REQUIREMENTS DONE ASAP!!!

MINUTES 2001-08-02

REPORT

“....this is (yet another) unprecedented system”, explain the problems, pitfalls and advantages of evolutionary prototyping -solve them buy XP.

SYSTEM DEVELOPMENT

Fusion method and rational unified process are using prototyping

Method selection criteria (~why prototyping)

- Make sure requirements are right
- Improve communication
- Try something new and different
- Faster development of a product
- PDS expect a prototype not a finished system (proto vs OOAD)
- Web application => not OO applicable
- RDBMS => no OO relationship

We should list the steps of XP and prototyping side by side and see how we could combine the two to make out own method. Need system definition: SQL Server, intranet, purpose of system (fixed), pitfalls, etc.

XP-STEPS

- CRC-cards
- Write unit test
- Coding
- Testing

PROTOTYPING-STEPS

- Establish prototype objectives => Prototyping plan
- Define prototype functionality => Outline definition
- Develop prototype => Executable prototype (XP STEPS GO HERE)
- Evaluate prototype => Evaluation report

TENETS OF XP

- Keep it simple
- Code in small iterations
- Fast release cycle
- Unit/functionality testing a crucial part

Modifying the product after release = maintenance

Modifying the product during development = prototyping

OBJECTIVES / SUCCESS CRITERIA

Should be measurable

CLIENT/SERVER BY JAMA

Should we do validation as JavaScript on client or JSP or JavaScript on server. What server to run JSP on?

DATABASE PRIMARY KEY

Should show how many days the course lasts, e.g. MSWORD2-260901-2

REQUIREMENTS

Functional / Non-functional

- View participant list
- Print preliminary confirmation
- Print confirmation
- Print late application confirmation
- Print course certificate
- Print cancellation
- view list of courses
- CRUD operations for course, firm, participant, location, instructor etc. Add and remove for valid combinations
- Product layout => Organisational layout

MINUTES 2001-08-03

TABLE 15. Suggested table of contents

Column 1	Column 2
• Title page • Foreword • Introduction • Company profile • Problem description • Problem delimiting • Table of contents • Project establishment => trivial pages go in appendi • Resources • Contacts • Development methods • Development method • Prototyping, types, why evolutionary • Plan - Outline - functions - Implementation, XP why?	• XP - What, How, Applied • Initial requirements • Analysis • Evaluation • Methods • Tools • Process - Project conclusion, Bibliography, Glossary, Acronyms, Appendix • Contracts • Minutes • Code • Prototypes do what APPENDIX Description of the prototypes should go in appendix. The report contains the problems and process, why/how.

VOLUNTEERS FOR WORK

- XP + system development methods: Paul
- Prototyping: Jama
- Database (E/R) Front-end (Usecase)
 - Normalization
 - Conceptual db design
 - Logical db design
 - Physical design
- Write unit test
- Implementation
- Test
- Evaluation / feedback
- Sequence diagram
- Navigation map
- Logical view
- Write unit test
- Coding
- Test

External Contract

- Evaluation / feedback

Requirements: Carsten

Web + DB not OO applicable: Harvey

User interface design: Harvey

TIMEPLAN

- Harvey has timeplan (will e-mail to others).
- Meetings with MC Wednesdays.
- If we have informal meetings, class will end by 12.00 on fridays.

MINUTES 2001-08-06

Jama and Carsten read and corrected the project establishment for Harvey. Lots of things was messed up from Harvey's side - Remember to forbid Harvey to come on Mondays!

Baselines for prototypes

- DB
- Intranet/Frontend
- one per prototype

Requirements for 1st prototype

Send PDS the requirements we have and let them add/modify the list.

First prototype:

- Get interface right
- Usable and informative

Requirements are presented on the screen.

What/how to design screens.

System definition:

- Business things: Torben doesn't have to do his work at the same computer - he can do it virtually anywhere there's a internet connection

First prototype is purely interface design. The database will be developed concurrently, but has no relation to UI.

Prototype 1

- **Objective**

The prototype is able to present in logical and easy to use manner, data stored in the database to fulfill requirements

External Contract

- **How to achieve**

Navigation map

Logical view (Conallen ~UML)

XP implementation phase

Maybe we should use different feedback methods for each prototype...

- Schema
- Interrogation with physical punishment
- Nothing
- Something

MINUTES 2001-08-07

Jama will not be present Friday, as he is going to visit some family in Jylland.

SYSTEM DEFINITION

Functionality: Create participants, courses, firms, etc. (SEE FUNCTIONAL REQUIREMENTS)

Application domain: Administrator of Courses, internal interests.

Technology: A server running SQL Server 2000, Internet Information Server 5.0 with ServletExec 4.0 engine, clients running Internet Explorer 5.x.

Conditions: For further development: We leave PDS with a working prototype and documentation.

Objects/entities: Participants, courses, firms, locations and documents (in some form)

Philosophy: Course and participant administration, distribution of course information internally.

MINUTES 2001-08-08

TOOLS

Why do we use JSP instead of ASP, now that PDS is Microsoft Certified Partner?

Should all pages be JSP to make it consistent (instead of HTML)?

- Which is fastest?
- Which is better?
- Server issues
- Consistency
- REASON IT!

We have decided that all pages should be .JSP for consistency and functionality!

We got the software from IT-department, went and installed it on machine in room 117.

External Contract

Are we improving working practices or are we making it worse?

So far requirements are mapped to web-page layout.

Will PDS continue to use JSP or will they change everything to ASP? Are they only using ASP because they are Microsoft Certified.

Can we get additional fields on the course web page? (DOB in particular)

What will be the PK for participants already in the system?

Harvey has a paper with questions for PDS meet...

OTHER

Will there be / should there be a list of interested people who had their course cancelled. E.g. 3 participants get their MSWORD course cancelled, are there then stored in a table of interests, so they will be notified when a new MSWORD course is created.

FATCOP - Conditions: System should not change existing working practices, no matter how bad they are ;-)

What happens if there are too many or too few applying for a course?

We should leave options open / develop with further development in mind.

Quote by PT to put in report:

We want to document the requirements specs that we write and the responses to them, as detailed as possible, in order to illustrate the advantages of using our development method. Hopefully they will change requirements or we won't interpret them correctly to show the value of the whole thing.

Paul has gotten that XP book by Kent Beck...

MINUTES 2001-08-09

Prioritized list of requirements where they put the number

PDS's own list of requirements

What practices do they (PDS) implement in XP

Metaphor is what?

What do you test - not everything, but possible breakable things

Capability Maturity Model!? Harvey will look at

Quotes use special paragraph formatting in FrameMaker

[LOOOOONG XP presentation by PT, took like a badjillion pages so I didn't bother writing them on the computer]

MINUTES 2001-08-10

PROBLEM DOMAIN QUESTIONS

- Kursister som tilmelder sig et overtegnet kursus? Salgsafdeling administrerer manuelt.
- Hvis undertegnet? Brev med at de ikke skal komme => salgsafdeling.
- Salgsafdelingen vil gerne have en venteliste med interesselidspunkt og deltager, som prioriteres efter hvor lang tid de har stået på listen.
- Hvis personer gifter sig => nyt efternavn!
- Nuværende systemer: DSI, C5, Approach, SQL Server
- Tabeller skal være fremtidssikrede
- Lotus Notes intranet
- IIS ikke noget problem
- Alt kører i Delphi (vil konvertere vores system til Delphi)
- Torben ser det som en fordel at systemet kører via browseren
- Vi har fået en lille maskine med IIS + SE til formålet
- Det skal være så nemt som muligt at tilmelde sig et kursus.
- Kortfristet tilmelding 14 dage før
- Kursusliste mangler beskrivelse/kommentat/kursusnavn efter kursusnummeret
- Sum af og max antal deltagere ville være rart
- Firmaer ligger i telefon-systemet
- Ingen instruktør-tabel, bruger kun initialer
- Tabeller/kolonner/variabler skal enten være på dansk eller engelsk - IKKE BEGGE!
- Diplomer udskrives hjemmefra

PDS ON XP

- Par programmering bruges 2 timer pr. uge
- Involvering af kunde bruges ved kravspecifikation og prototyping
- Support-afdeling fungerer som testbrugere
- Nye folk bliver sendt ud til kunder
- Storycards bliver brugt sammen men UniBank; omprioriterer krav specifikation
- 15 versioner på 3 mdr.
- Metafor ~ Jesu lignelser; forklare et problem på en anden måde.
- Syntax (.dk?) bruger 20-30% af XP
- Bør ikke teste trivialiteter

2 slags test: (hos PDS)

- Planlagt test - de dele man ved man har ændret
- Systematisk test - de basale ting, daglig brug af systemet
- (Monkey test) - Prøve at teste for umulige ting, fx. bogstaver i et tal-felt

External Contract

- Brug stack-dump i java
- Forskelligt hvem skriver storycards
- Benyttet: Overordnet kravspecifikation, kritisk modspil for kravspecifikation over for kunder.
- Vi vil gerne prioritere storycards, hvis i selv skriver dem!
- Umuligt at tidsestimere hvis man er ny eller hvis man er i et nyt område

3 tidsestimeringer: (førte til liste med over 200 punkter)

- Optimistiske
- Realistiske
- Pessimistiske

- Fejlhåndtering 50% af tiden
- Forvent det uventede!
- Hav altid en superklasse til næsten hvad som helst
- Laver altid prototype med userinterface
- Exploratory prototyping i realistisk omfang
- Stress-test (technology prototype)
- Skrive små tests
- Klassediagrammer: Bliver brugt uformelt og til diskussioner, også til indbyrdes kommunikation
- Send spørgsmål på forhånd!

Design kriterier er 2 ting:

- Visuelt - må ikke virke fremmed, skal være genkendeligt
- Bagved - overskueligt

PDS har skærbilled-test med indbyggede fejl - der blev fundet flere end der var lavet!

MINUTES 2001-08-13

Weekly review (HS).

Documents should be renamed _final if it is the final and latest version instead of date.

REPORT BASELINE

- Description
- Intermediate products
- Criteria for evaluation
 - Docs = predefined standards
 - Consistency of content
 - Advisor approval
- Procedure for evaluation

External Contract

- Proof reading
- Editor updates report status
- Delivered to advisor monday in PDF format

Points for the agenda must have some time estimates

Systems development overview

- No OOAD
- Good evolutionary idea
- Prototyping advantages
- Disadvantages of prototyping are solved by XP
- Our version of XP
- Experiences with this method?
- Relatively smooth transition

Write consequences of bad design things that will happen if they just convert the old system to SQL Server.

There will be small phases in between the major development phases to breathe and think! Add flexibility to the plan.

Can we test JSP with JUnit?

- Do we need Beans to do it?
- How do you test JSP pages?

CF + HS will look at JUnit! Need some classes to use JUnit. Ask Anders to look at it.

MINUTES 2001-08-14

Is it evolutionary prototyping when PDS will throw it [the final prototype] away and re-do it in Delphi? The answer would be YES, because what ever they do with it, is none of our business and out of our scope.

Refactoring: When processing the code, refining and simplifying the code for higher code quality and overview.

XP-exploration: Find out what a system should do = story cards = req. spec.

Out the idea about educating system developers article by Lars Mathiassen in the beginning of the the system development chapter.

- THEN WE DID THE USER SCREENS - (Jama has them on paper)

List if companies should be narrowed when selecting an e.g. dentist-specific product.

Write about why HTML is good enough for what we are doing.

Physical DB design gives much to write on SQL Server 2000?

MINUTES 2001-08-15

MEETING WITH MC

- Good language
- “Red thread”
- Steering of project != development method
- “Customer training courses” should go in problem description
- DB development method:
 - In the beginning there was no method
 - Then came the Bachmann method
 - Then introduced E/R (BAD: no functionality)
 - Structured Analysis
 - The result of E/R and SA was Integrated System Development
- Prototyping = OOA (?)
- What methods and how
- Tools - do better discussion
- [To Carsten] You can get 3 pair of frames for glasses for 1000,- at Nordens Optik!
- Putting methods, etc. in the project estab (150 pages ok!)
- Tools - why and alternatives
- SQL Server vs InterBase - reasoning choice
- Positions: What about alternative group settings, e.g. rotating PM?
- Counter group - Haven't heard nothing from them yet!
- Risks - not enough on list, risk that we want to incorporate everything, risk as a tool
- Our project could look much more like the spiral model
- Project establishment: Approved by MC! Don't send him it again!

MINUTES 2001-08-16

Make whole chapter on Tools - Project establishment has only a list

Talked about testing of UI

*** WHAT ELSE DID WE DO THIS DAY ***

*** THAT'S ALL THERE IS ON MY PAPER ***

MINUTES 2001-08-17

DB Baseline due for monday 20/8 (Carsten).

Write about the tools we have tried before making the decision on tools we have. Does the methods recommend a set of tools?

Coding standards in Java and SQL (Carsten).

Write text for DB design; You can picture the result of the DB design, so there is no need to do all the steps. Explain the ER syntax used only (partial participation, etc.) Database definition, Course dates are multivalued attributes.

External Contract

Q4PDS: What number system are they using for their customers?

Can you delete a course from DB and still keep the participants in a table with the course_no only (~ on delete = no action). Deleting the courses reduces the size if the DB tremendously over time. You can still see the name of the course the participant has taken, because the Course_type table exists. What if a course_type is changed, e.g. MSWORD97 becomes MSWORDXP? Is there any reason for storing old course types?

LOOKUP Courses => LOOKUP Course_Types IF Course_Types.Duration THEN
LOOKUP Course_Dates

Diagrams of what we have for them

MINUTES 2001-08-20

Data dictionary go in appendix.

Use Adobe GoLive 5 for HTML instead of FrontPage

- Conclude that FP sucks and GL works
- FP weird code and GL fast!
- Tidy to check XHTML syntax

Don't write further on documents sent _for_editing! Editor will discuss/review documents edited every monday. Takes approx 2 hours in order to be thorough.

Tools should NOT reflect economic perspectives!

You => PT = _for_editing

PT => you = _edited

You => Librarian = _final

Time is a valuable asset, so throw-away prototyping is not applicable to our project.

Do we only include requirements for further development? - Only requirements for our project/model/our viewpoint

DB TEST PLAN

- Validate transactions
 - INSERT INTO Applicants VALUES
 - DELETE
 - UPDATE
- JSP/DB integration
- JSP web server

Lots of broken links on the pages.

When does a course become removed (date)?

External Contract

“Afholdt status” necessary or just check date?

We thought that trivial testing wasn't needed - but this test shows the value of it and the advantage of writing test prior to coding. One person dictating and one person writing was.

Course_Edit: Check boxes instead of radio buttons.

Printing from Word is piece of cake, after looking in Jama's JSP book (setContenttype = application/msword). Opens the page in Word even though the contents of the source page is in HTML

MINUTES 2001-08-21

We (and PDS) are missing some rules for the standard letters.

Split relationship between Applicant and Course up into Participant (course_no, applicant_no, order_no, state)

Breve (on course_administration screen)

- Tilmeldinger (2 venter)
- Bekræftelser (4 venter)
- Endelig bekræftelse (0 venter)
- etc.

Automatic update of states

Is there a waiting period between states/letters?

MINUTES 2001-08-23

GENERAL POINTS

- Try to maintain a calm transition from one screen to the next
- “Fortryd” button should be called “Slet felter”
- Start page should be “Kommende kurser” with links to the others
- Would like overview of locations (NOT GONNA HAPPEN)
- Should use browser functionality more effectively
- Could give different colors for courses with different status
- Try to keep similar format to make user interface recognizable
- Icons and tabs?
- Separate search from edit portions
- Make sure that fields are in the correct order
- Make sure that links work (Imbarresing one!)
- Save course and participant history
- Link => “Format page for printing”
- Hide irrelevant primary keys
- One screen for participant (create + edit)

- Fixed menu bar on the left side of the screen

SPECIFIC POINTS

- Course Administration: “Brev” in stead of “Formbrev”
- Course Create: Get max deltagere from Location as default value, show desired date format for entering, 4 fields for dates
- Course Edit: Add phone and C5 order no and better separation of edit course and participants (HAS BEEN TAKEN CARE OF - SEE NEW LAYOUT)
- Location: Text areas missing: “Forplejning”, “Overnatningsmuligheder”, “Praktiske oplysninger”, “Kommentar” that SHOULD NOT be printed, “Kørselsvejledning”
- Participant create: Customer no and Customer name should both be there, C5 number should go below Course_no, C5 number should be entered if a course is chosen
- Participant edit: Search by firm name
- Firm create: Get SQL script from Benny ASAP!!!!

Starting point for user interface was “we don’t care” which turned out not being true.

“User interface requirements” => in requirements document

“Coming courses” page should have links for each participant - click on the course or firm to view/change.

ADVISOR MEETING

- Too many files sent (Carsten’s bad!)
- On the right track (not onTrack, or?)
- Parallel activities
- Evaluate chapters separately
- Look at ER for Course_type and Course again
- Consider further use of the system
- Make table for some of the relations and not just the entities, is what he usually does
- Add flexibility to system - don’t listen to what the users say. If you have implemented the future/further use of the system, and the users come to you and say they want this and that without knowing you have already implemented it, it’s VERY easy earned money...hehe
- What is basis for choosing just exactly this prototyping model?
- New methods that we haven’t learned anything about is good ~ higher grades
- Don’t explain XP - just use it!
- XP is just a new buzz word for prototyping
- We do technical prototyping ~exploratory
- What part of the system is responsible for testing the UI (?)
- Equivalence classes - Do research! (Is one of the ways to do black box testing)
EXAMPLE: Instructor - 1) search for one that exists 2) search for one that doesn’t
- Prepare SQL for results of 0, 1 or many matches

External Contract

- Test: Problems and solutions
- Next meeting: Thursday 30/8-2001 at 1300 hrs.

MINUTES 2001-08-24

Technical protoyping - test the tools we use; SQL Server, Word, Frontpage. We were flexible enough to change to GoLive instead of sticking with Frontpage and refactoring the dirty code (~unreadable code).

Designed user interfaces and reinstalled sw on machines all day.

MINUTES 2001-08-27

Problems switching with GoLive

Data dictionary for everything code related

DB baseline

UI baseline

Tools (we will use?): Visual UML & UMLPad

Docs due friday:

- Storycards (All)
- ASP vs JSP (CF)
- Tools update (CF)
- Testplan (HS)

How much of XP do we want to include?

What are we using and what are we not using?

Thursday: Create database, user real data, no “Mr. Piggelwiggely”

“We experience difficulty writing the use cases, because we had to regenerate ignorance about the way things are going to be done, beause we’ve made the user interface already, however maybe it’s a better way...” (PT)

Brainstorm of stories is moved for tomorrow, AS PAUL FORGOT HIS BOOK!!!

Test plan (PDS wednesday 29/8-2001)

- Carsten show and tell
- PDS write down questions meantime
- Questions from PDS
- Torben tries it out
- Print out screens and write comments on them directly

MINUTES 2001-08-28

Mail XP-report to Jama and Paul, since they have thrown it away!

Make PDFs for minutes with one week per file

Database

- What have we done to ensure migration from Approach to SQL Server?
- Data Transformation Services - CF look at / write about
- Alternatives
- Constraints; max deltagere, don't delete a course with participants
- Participant doubles are permitted
- Storing taken courses for each participant
- Size of DB they don't care about - it's cheaper to buy a new harddrive
- Improve, extend, futurize their current db
- Torben inherited the old DB
- Approach ~visual DB clicking you way through the process - PROBLEM?

TABLE 16. ASP vs JSP comparison

	ASP	JSP	Applicable
Platform	WinNT (Win9x)	Independent	No
Standard	Closed	Open	No
Commercial	Yes	No	No
Server	IIS or PSW	Many	No
Components	COM	JavaBeans	No
Extensible	-	Own tags / less scripting	No
Scripting	VBScript + JScript	Java	Yes (+JSP)
DB access	ODBC, ADO	JDBC, ODBC	No
Learning time	Short, but completely unknown	Long, but we already know alot of it	Yes (+ JSP)

MINUTES 2001-08-29

Demonstrated the prototype

PDS went through the screens and gave us feedback

Started doing stories/tasks

Lunch

More stories + correction of screens

STORY: CREATE COURSE

User choose "Kursus" from the menu. In the "Søg/opret" box, the user puts in the desired course number adn clicks the "Søg/opret" button. (The page for creating a course will be loaded with empty fields). The user then fills out the relevant or known fields and clicks the "Gem" button.

What would the difference have been, if we had made PDS write the stories first? We had the existing use cases in the form of the user screens given. The user involvement in the project is not as big as it should have been according to the XP idea. It is a step back if we have to write the stories. We are adjusting (to?) the method.

MINUTES 2001-08-30

How does XP go in hand code from other places (such as Anders' RecordSet2Table)?

Technical prototyping: Does a tool fulfill our requirements, what do we need tool to do?

Split up the TOOLS section - and put the HTML tools into it.

Simulate stress on web + db server; make program that loops HTTP GET ANALOGY: You start with the basic things (in a car: frame, wheels and engine) and then add extras (A/C, power steering, etc.)

There is enough elements of risks in the project so we choose a language we are familiar with.

Coding standards: HTML (done) Java/Swing graphical element names.

JSP has cleaner code than ASP, ASP is disorganized - no datatypes

ADVISOR MEETING

- We are NOT doing a textbook, but a project - so not so much theory
- Don't mix general comments in the conclusions you make during chapters
- Keep down the introductory theory
- Comment only theory where it gives some problems
- Only 1 or 2 pages on XP theory
- Write only about thoughts
- Is the text too small?
- Result of prototyping is a navigation diagram
- Reference missing to test document in prototyping document
- Students + developers mixup
- Discuss relations and make a conclusion
- This is the drawback of not doing OOA: Lots of iterations in ER
- Problem domain still a bit unknown
- Technical prototyping eliminates high risks of new tools
- Do it where tools might cause problems
- Squeeze it in the relevant prototype section
- ASP vs JSP: MC prefers JSP, client/server functionality placement not dealt with yet
- Big company => Own methods
- Small company => No methods

ASP/JSP • Learning time

- Who knows what
- The risk being reduced if we choose JSP
- Point out the good and bad points
- Lots of similarities
- We chose JSP because...

DB

- Looked at options, but it is highly unlikely...
- Another way might have been to do this...
- Same as with the 1-M that with time could become M-N and therefore you add an entity in between
- The design we have done is based on their business constraints!

MINUTES 2001-08-31

Coding standards: Reference Sun's standards (find link for the references!)

New link in the menu on the left side of the screen: "Breve" opens an overview of all the letters that are pending for printing, batch printing possible.

Where do we place the functionality to check the data that is entered? JavaScript on the client or Java on the server?

Do error-handling while coding!

Next PDS meet: Systematic check list for Torben, thorough!

Why or why not are we using sessions?

Thin or thick client? Jama will explain in document

DB

- Table for printing - HOW?
- Standard letters are to be stored in the database as a long HTML-formatted string
- When do you use stored procedures vs application functions vs views
- Technical prototype of JDBC driver

XP

- Story categories (good for user prioritizing):
- Necessary for the system
- High business value
- "Would be nice"

MINUTES WEEK 36**From 03-09-2001 To 07-09-2001**

The following section will describe the activities and discussions that take place in the

RHS either in the group room or on the PCs.

03-09-2001 • We will have two counter group. Kenneth and Thomas's group probably on 20-09.

- PM explain project status.
- Recommended to have an extra BaseLine for the DB as we had have some BLs before. BL will be the end of this iteration.
- XP1 will include 2-3 screen functionality.
- E-mail from PDS.

Session: Even though is not much applicable in this case (Project), we will try to use some where in the beans to record the user's history.

ASP/JSP: No ASP and will make a reason why we shouldn't. Send to them some (3) point to justify why we use JSP (Carsten). Again if they insist to do so, then go back to the contract.

- Review of documents: We end-up to make clear the difference/similarity of XP and evolutionary prototyping. previous thoughts was that the implementation of the evolutionary

prototyping will be XP, but you can keep separately these two subjects.

After discussion we conclude to implement with XP all left iterations. The product of each iteration will be prototype!!!.

- PDS's new requirements: We will implement as much as we can and where applicable.
- Discussion of cover page. Harvey's sample will be temporary.

2001-09-10

- For the counter groups, we will check the report on Monday (17th), and we will submit to Kenneth on same day. On tuesday we will submit to the one counter groups (Thomas's).
- DOC:
 - Some changes are made in System Development Methods based on Advisor's meeting. Why UI prototyping became first thing to do? This discussion should be included in the above mentioned section.
 - Technical prototyping: Document format should consist general approach then some detailed experience from our discussion.
 - Server Side Scripting: Where to put this section? This will be placed same place with Client/Server section (From jama) and this part will be called Web Architecture. The seminar of .NET will be mentioned this part, but we will not give a consideration according to our development processes.
- XP1:

- We didn't succeed the goals of the baseline. We will make a separate section to explain what is the reason for unsuccessfulness.
- Does the XP practices are new in the system development literature. This discussion's conclusion will be included in the appendix.
- We minimize to make sessions and cookies to fulfill the user requirement. There is no reason to use cookies as there is no login ETTTTTCCCC. Instead of session we will use page and application scopes (where are necessary).
- 117 for coding:
 - How to connect the DB. Do we make through the JSP, do we make a generic bean connection, do we make a connection pooling or do we a java file and import the class file in each JSP file.
 - At the beginning we will make java bean+connection/communication of the DB and use the class bean in the JSP file. We unsucceed the above point, then we make the following observation.
 - Test the DB connection to call openDBConnection method in the JSP file.
 - Execute an SQL statement in between calling storeInDB method.
 - Close the connection by calling the CloseDBConnection method.

We fail the second point and again observe where the problem lies.

- We use main method in the Course_Type_test.java file and run it "import java.sql.*; and throw in main the exceptions throwed (used) in the methods called in main". We printout some control lines in the catch clauses. and we get them. Now we get that the problem is SQL statement. From this result we use simple SQL statement to insert a record to the DB. This statement is placed in the main method (not to use storeDB method) then we succeed access the DB. From this point we make incremental improvement, and finally we succeed to insert and access a record to/from the DB using the form then JSP to call the bean class which will communicate the DB.
- Today as we had have some deviations for our plan caused by the technical prototyping, we make clear what will be tomorrow's plan.
- So far we have the following problems.
 - No printer connection "Will be solved with Bo"
 - Different Java, JSP and HTML files on both machines.
 - DB. Should we keep in one machine (Physically) or place both machines.
 - No access for commandrive.

2001-09-11

- Printer problem will be solved alone as Bo suggested some steps to do.
- Task Cards Check: We compare the task card and the code steps that we did yesterday (course_type_create).
 - The Course_Type_Create.jsp file should be change. html file, as there is no functionality in it, but just it calls the jsp file. The other idea is the separation of the java codes with html ones.

- How did we implement to test the DB records, to avoid the duplicate records?. Do we make a separate bean to do so? NO. DB capability with the primary key will check and gives an error.
- After checking the task card we find that we were very close the implementation of it, and we will make the necessary correction for the rest of the task cards.
- Paul and Harvey continue to change the task cards according yesterday's work.
 - They finish to alter the task cards. and they make a table to estimate the engineering time needed for each task.
 - Then they start making Person_Create task with needs to work with Firm_Create task card. There is impossible to create a person with out Firm.
- Carsten and Jama goes to implement Firm_Create task card. The following are steps to implement Firm_Create task card.
 - We copy and paste previously created files "course_type_submit.jsp, Course_Type_test.java, Course_Type.java" to make the correspondent files for Firm_Create task, then include the missing variables and methods for these variables. Compile them but we couldn't. Problem is import junit.framework.*;, set the classpath in control panel doesn't compile, copy junit.jar file to the common folder of the files (c:\myjava\lib\ext) doesn't work, delete the classpath c:\junit3.7\junit.jar doesn't work, set the classpath of forte4j-fastjavac compiler c:\junit3.7\junit.jar NOW we can compile then run the Firm_test.java file in the dos prompt (c:\myjava>) writing "junit.TestRunner beantest.Firm_test" and it tests.
 - For the DB connection we did same steps as yesterday. Yesterday there were an error with calling the closeDBConnection() method with out closing the resultset and statement and also free the memory space making both equals null. I we try to insert a duplicate record to the DB the system wil neither accept nor crash, but we get SQL exception then we have to catch this exception with user friendly message.
 - In the Firm_create.java bean we use setProperty in one line to avoid repeating lines of each text field, but we have to have same names of bean variables and text fields in the form.

2001-09-12

- It was an unnormal day because of America Under Attack event. We start 10⁴⁰.
- So far we done 2-create task cards (Firm and course_type) completely and two others (Person and Location) partially. we will try to finish those task cards and then start writing remaining task cards.

2001-09-13

- First step of today will be to make clear what to test for the tasks.
- For the DB we should have a SQL script to create a metadata to the DB.
- The following is brainstorming of what to test.
 - connection != null.
 - getFromDB is never called with PK field empty.
 - Form parameter names = bean parameter names.
 - Inserting doubles gives error message.
 - Input bean = bean.getFromDB.

External Contract

Redirect happens/taken to right/attended page

Fields are valid (date, customer_no).

Not null fields are filled out.

Tidy validation.

Cursor positioning after submit.

Relevant error messages.

Input data should be checked "Data format".

After submit of a form refresh same form with empty.

Search without results gives error message.

- Some key codes will be included in to the report and the rest of the codes will be placed to the appendix as usual.
- DB-connection/close test should we make a separate class or are there any other ways to do so?
- We will make a generic class to connect to the DB using JTurbo then another class DB-test, which contains the test methods. This test will be next iteration.
- Field validation and data format will be controlled with JavaScript, but the question is how is the risk of this for the group? "probably high". There are some other options such as using JSP tag libraries, and also we can use java codes "servlet" to do so. AS the knowledge of the group for java is better than JSP and JavaScript then we will implement with JAVA.
- What are the other possible error messages that needed to display? we will make a brainstorming.
- Do we make all task with their test, or should we first finish all tasks and then their tasks? We couldn't integrate not passed codes in to the code pooling, therefore we have to make each task card with it's test and include in the coding pool when it passes the test.
- Jama and Harvey continue writing print and course create task cards, while Carsten and Paul continue to finish the implementation/coding of location create task card. Rest of the day we were fighting with Person (harvey/jama) and Location (Paul/Carsten) task cards.

2001-09-14

- We postponed the meeting with PDS as we haven't finished product to show them.
- So far we have two fully finished task cards "firm and course_type creates", two partially finished "location and person creates" and three written task cards "course, print and register".
- Why some task works perfect and others doesn't? the answer can be one or more of the following reasons.

Data we are trying to insert in the DB is not acceptable according datatype of the attributes and relations between the tables.

SQL statement syntax is different in the DB windows (SQL analyzer) and in the java codes "pattern of single and double quotes".

MINUTES WEEK 38

From 17-09-2001 To 24-09-2001

The following section will describe the activities and discussions that take place in the

RHS either in the group room or on the PCs.

17-09-2001 • There are two possibilities to the DB.

Login through windows authentication or to use a user name and password in the SQL pages.

- Some times there is discussion without a decision in the group. To solve this problem we have to stick that discussion unless we make a decision or we have to write on the KIM list “on the wall”.

- The review of the last week’s process we find out that there are some solved problems,

but there is no clear way how did we succeed those problems. Therefore what was wrong with our processes?.

Do we use hit and run process coding or did we make hierarchal building (step by step)? We agree to make hierarchal coding from down to up as we are using XP practices i.e. not to make all codes one time and check it where the problem lies.

- Baselines of DB3 and XP2. we checking the quality as well as the quantity of these baselines.

XP2: Before checking this baseline we have to make sure whether we succeed or not the XP1. The difference between both lies description and intermediate products.

DB3: Why is necessary to have some changes in the DB design? Because of iterations. If those changes are necessary we will make some alternatives and the decision of one of them depends the group discussion, user requirement advisor’s advice and our implementation capability.

- Print all documents to put in order and make ready for counter groups delivery.

24.09.2001

Question about thought of comparing prototyping to OOAD in problem description.

Problem delimiting should explain this is what we are looking into, not what we have excluded.

Pair Programming can be good to build a knowledge base, but it requires resources and why should a firm pay 2 programmers to do one job?

Need to make requirements that are verifiable

Explain how functionality for each iteration was determined at beginning of each Iteration chapter.

Make a User Manual. What does the system do, how do you do what you want to etc. Should be something that can be read independently of report. Include screen shots, navigation diagrams etc.

External Contract

Function List for appendix

DB section in Iteration 0, explain this does not reflect working practices but best place to put.'

Error Handling- could make a DB table of errors. Get these into a hash table when system starts

No meeting next week

Exam - no theory, just questions like how would system look if it was distributed?????!!!!!! or what is difference between C++ and java packages!!!! (none C++ doesn't have packages!!!!!!)

25.09.2001

How deep is use ful the advisor's advise. Its worthwhile, but we will evaluate how to apply it.

What is the use of interfaces as they are only a signature of methods? One of the main use is to keep the consistency of the signatures of the methods used in different classes, and each class can implement those interfaces as they want, but the condition is to write all the signatures of the implemented interface.

Navigation diagram seems complex. Is it possible to make simpler? According the user requirement and the test of user interface prototyping this is what they need for.

We need a user manual, which describes

- How to navigate in between different screens.

- Required field to fillout.

- The relation between search, create, edit and delete as they are in same screen.

- Data types of some fields if they are necessary.

Do we need to make a class diagram. If so there are different possibilities.

- JVision creates from classes to class diagram.

- Each iteration can have one class diagram.

- Common class diagram of interfaces, super classes, common classes, and relation between same level classes.

- As our project is not OOAD we will not rush the decision to make a class diagram(s). It depends our need, advisor and if the end-user requires more explanation of the systems.

After going through XP1 problems and solutions we have to stick to make a test integration and refactoring.

- Follow the written task card.

- Pair programming, one coding the other recording and corrections, and should have on hands, data dictionary, task cards, written test, and all necessary items to implement the task.

External Contract

If there is deadlock on a task card, call assistance from others.

To make easy the working process, we have to have on hands all survival kit.

Test should be done each class to migrate into the coding pool.

Create a generic Db class, should we import this class or extend. We will import as there can be some other classes to extend. This class will contain all necessary common methods to communicate with the DB. These methods will be designed in a way to use in all the beans. What will be the return value of select() method, this will affect the print task, which we are using in several beans. So does this method's return value can hold all data from those beans to printout. Return type will be SET as it has many methods, but the question is how will return the set value as you already closed the DB. The value of SET will be placed in a variable before closing the DB.

26.09.2001

As Db class used in by each class as import we have to make sure the test succession of it, otherwise it will affect all the other classes.

All Database involved codes should be removed from each bean and JSP files, but still there will be some necessary codes in each bean and JSP file.

Lower section of the course screen we have to make a separte task card called "Participant management".

Refactoring is next step to do, to make the codes good look, easy to read and well organized.

Why PDS has to firm tables in their DB? question to PDS.

Jama and Paul will make XHTML refactoring.

Harvey and Carsten Db class test.

27.09.2001

As we need to handle some errors do we implement with JavaScript or JAVA. If JS we need to make few functions, but again how does this affect the architecture (thin client) and group knowledge of JS. We can make an JSP error page and use it each JSP file, the error page uses each bean to handle the errors from them.

Error handling will be made by JAVA and JSP, but NOT javascript.

First error to handle is those can take place in course_type and can be.

Are the required fields filled out.

Non duplicate record in the DB.

We split the duration field of course_type in two "start_time and end_time". Remenber to change all related files "XHTML, JSP, bean, database in course_type table".

28.09.2001

Time estimates of the next coming task cards are more than the available, then we will focus first three tasks, and if we succeed ontime we will proceed next tasks.

As we're behind of our target we will do some coding activity "refactoring" at the home even though it contradicts the XP practices. The home work should make any deviation to the system design.

To remove unnecessary DB codes from each bean, we have to make sure the Db class test and functionality.

MINUTES WEEK 40

10/1

Group went over project review/status and that was okay.

Group went over weekly review and we need more discipline writing and using tests.

User manual in FrameMaker and HTML format. This is an important part of the project with respect to thoroughness. Jama says that the user manual will be comprised of the introductory manual and reference manual. Functional description and installation document to be included in report.

Not everything was working 100% but we needed to continue in order to get this iteration's functionality done.

We need more proactive problem handling by all group members.

We use Tomcat/Forte solution for development purposes and IIS/Servletexec for development. We need to change the tools document accordingly.

Coding – continued Participant, database class integration.

JTurbo evaluation expired.

Important to include exceptions object from database in our exceptions message otherwise it is not enough information.

There was a problem that contact person was all in one word in the database.

Needed to change participant data types from bit to int, and from char to varchar.

10/2

We talked about refactoring possibilities

- Making a super class over and above the database class
- Maybe look at if-sentences versus case

We also talked about corrections of the screens

- In course_type there was something with alignment for materials and comment
- Jama needs to add the refresh idea to the explanation of the person page in the user manual
- Need to put stars next to required fields
- Something wrong with select methods for course_type
- We need to look at firm number in firm because when left blank it equals zero
- Default should be new person for person
- Drop down boxes maybe should have a sorted set in order to have a static ordering

We talked about putting solutions of XP1 in XP2 and evaluation of these solutions.

Need to change the getFromDB methods to include order by name in XQL.

10/3

We need to look into the value of interfaces in our code.

Harvey brought up refactoring idea with the get and set methods in the classes and we need to look into this later.

C and P continued with Participant and H and J continued with error handling.

Possible error message – max persons equals persons?

Should not be able to create course in the past.

Refer to page 531 for new date format. We decided that for them to put in the date it needed to be in this format with slashes 3/10/01. Then we would convert and compare it in the database.

Carsten solved Harvey's/Jama's coding problem and Harvey tried to kiss Carsten. Carsten said: "Hey sorry, I'm not that way. No matter how many times you try, give it up and stop sending me flowers."

We need to remember to check to see if a course is held for more than two days.

C and P had problems today because the XQL sentence that they used was not of the standard format and they were missing a standard solution.

10/4

We decided that we needed to look at the overview today.

Jama did not come today due to a doctor's appointment so Carsten worked on overview while H and P decided to go through each page and make sure that everything is working for tomorrow when we go to Plass.

We had many problems with the computers.

Anders came by and tried to help us but it was not easy because of the machines. H and P discovered a way to forego using the `getBeanType` method, which solved a big problem when using the `getFromDB` method.

Carsten was able to get most of the overview done but without links.

10/5 – Day at Plass.

- Move customer number in on overview
- Move 2 out of 10 also to the right
- No column headings on overview
- No city entry necessary – get from postnummer
- Pages should notify somehow that you did not press slet felter
- “Focus”
- Should go back to first field
- Keep duration as it is now
- Error messages should be identical in format
- Should not erase data when you put in the wrong text
- Course boxes should have values already showing
- In course page links to the overview page do not work
- Should be able in person to create multiple people for a firm without having to search for a firm every time
- Error message in register for max persons and should not already be in that course

Overall they liked it. When asked what was most important for the next iteration Benny said error handling.

We did a weekly review and we talked about the weekend’s assignments. Harvey/code, Paul/XP, Carsten/error handling, Jama/user manual.

MINUTES WEEK 41

10/10

This week the minutes start on 10/10, but on 10/8 the group stayed home to do document work and 10/9 the secretary was sick, but document work was continued.

We went over the XP1 and XP2 documents that Paul brought in.

We also discussed what to do about comments from PDS about changing the course page to include registration of participants. We decided to leave it as is.

We discussed things that were missing in the document.

We made corrections to documents for delivery to Claudius.

External Contract

Carsten sent his error handling document to Claudius later from home.

Jama worked on the logical view text and diagram.

The day was cut short because of a meeting that Den Danske Bank was holding, so we stopped at 1 o'clock. Some of the members went to that meeting.

10/11

Today we decided to look at task cards for:

- Course
- Overview (includes altering the page layout according to acceptance test)
- Error handling for Participant
- Print
- Person_edit
- Course_date

MC came and postponed our meeting until Friday at 11 o'clock because he got the documents too late.

We decided we needed a refactoring of the Course bean to include an attribute that would keep track of current number of participants. This was in order to reduce database access. The original idea was to make a SQL statement that would return the number of tuples in a given course.

Carsten chose to use the select to stink method.

Harvey pronounces "donkey" as "dunkey" (well, he's from New York).

We added test to the task cards to make sure that we did them.

Carsten updated and made sure that both machines were working for the next day's pair programming.

We made a list of daily work practices to add to the survival kit.

10/12

C and J worked on programming course_date task.

P and H worked on coding the Print task.

Both groups forgot to test even though we just looked at daily work practice sheet. We therefore had to go back and write tests and run the test suite before we started any more code.

Meeting with Claudius

External Contract

- Error handling belongs in iteration 2
- Do not explain why we use JSP in logical view text
- Link diagram is just one way to show navigation and Claudius wants it with arrows
- Add how we had problems with too much work, new method, etc.
- Add exactly what we did during the iterations – step by step and outcome
- Next Friday we need to deliver our full report to this point to Claudius
- In iteration 1 maybe we should add whether dropping pair programming was beneficial
- Claudius thinks that story cards demand an unrealistic devotion on behalf of the user
- We need to document what a normal task card looks like
- Claudius is afraid that the parts will not fit together
- Somewhere we should explain that our data dictionary is more than that
- Maybe we have not specified what the classes behavior is and we therefore had to look at the dictionary
- Last paragraph of refactoring in iteration 2 – tone down
- Claudius thinks that we should say in our report that, “We were really missing Lars Mathiasson’s architectural design”
- Put in report what kind of projects are best suited for XP
- With respect to putting the database class in a different package than the others what you gain over inheriting is with inheriting you lose a little performance. In chapter 4 of JPL, there is a good example of some program code that shows the flexibility of using classes in other packages
- Next meeting on 25

After the meeting

H and P figured out that the problem with test of course type was comparing the string “8:30” to “8:30 “ and therefore gave an error. We must change the data type in the database.

H and P added a delete method to the database class in order to delete inserted tuples from test classes (course type).

C and J made tests for course (course dates) and added a signed persons to the bean and the database for course.

We added all tests to the test suite and successfully ran them.

MINUTES WEEK 42

10/15

- Went over new format for iteration chapters to include what was done – approved by group members, nice addition
- Briefly discussed error-handling document and additions needed
- We addressed the following document topics:
 - Format for code chapter – two columns sideways

- Print out full report for each member and take home – with respect to changes, do them in a second file or on paper
- Changes over the weekend, both to the database and the code, resulted in the test not running that ran before we left on Friday. My personal opinion is that we should stay later at school to do changes to things if they need to be done, otherwise it defeats the purpose of pair programming. Maybe we should alter the plan to include these little tasks if they are necessary
- After lunch, C and J finished course_date task and P and H wrote tests for Participant and experienced problem with broken file error again
- Harvey left at 2:45 and C, P and J ran the test suite, which is now up to 13 tests with three failures. This is okay though, because these were expected due to the fact that the methods were not yet implemented. They were also successfully tested with same errors after integration of the day's work

10/17

- When C, J and H left yesterday they could use the Firm successfully and print a letter. Today we couldn't. There was some mix-up again with integration. Thereafter, C and J continued work on overview and test on whether converted date from database was correct. P and H wrote test for change printed DB. P and H refactored some code. We used to have a store in DB method, a delete from DB method and update DB method and combine them all to be contained in a general update method. This was possible due to the fact that the database has the same update method for each of the three instances. This is all in the superclass, each bean will still have delete, insert, update, but each will call updateDB in superclass
- After lunch, we had problems with the DB. Because we made changes to the DB script, we had a mysterious change to the database. Suddenly the participant table had no connections to the other tables. We therefore wasted a lot of time with errors we encountered thereafter, because we couldn't understand why things were not working. We found this out by referencing a database generated diagram of the tables
- P.S.: We need to figure out where to put max_persons

10/18

- Today, Participant wasn't working because we needed to make State a foreign key from the standard letter table. We therefore needed to implement the standard letter table first today
- We discussed how we need to focus on just this iteration's work
- It is interesting to evaluate what the value of a thorough requirements specification is in XP. It has many times confused us thinking about future possibilities of just some brainstorming comments made by PDS. Non-functional requirements of course are still of high value because they do not change (normally) throughout the project

External Contract

- Tomorrow we are supposed to send the report to Claudius and we each have some things to add so we will work in shifts on the document while others program
- P and H finished Print and started on Person_edit
- C and J worked on Participant management and also worked on many not task related articles

10/19

- To begin with C and J continued work with Participant management and P and H polished the document for delivery to MC
- After lunch, C and J continued work on Participant and wrote a test for Participant for a state change
- P and H continued edit_person and completed that as far as changing the name is concerned
- Jama left at 1.00 and we did the weekly evaluation

MINUTES WEEK 43

10/22

Last day of coding before code delivery to Plass.

- We made sure that we started with tests
- Tests were not as simple as they seemed. The tests are often inter-related and running one test could break another. What was needed was to run the set-up method that would reset the values for the test objects
- C and J found that it was important to keep the same C5 number when changing from one course to another while working with Course_administration
- H and P finished the Show_letter page and afterwards tried to add error handling to the Firm page in the event that a firm was searched for and not found

10/23

- Today, we had our last presentation of the product with Plass. We met early at Plass and fixed all the unit tests first
- We then went through the functional test and we found that in Participant_register we needed a while loop to add more than one participant
- We discussed the need to go through each file's code on overhead for the group so that each member has an understanding of the code
- We then collectively tried to tackle the problem of making our own error handling not dealing with SQL exception but couldn't figure it out
- We made new acceptance test format to have them test all functionality

Comments from the gallery:

- Seems like there were empty spaces in text boxes for location – that could be why it starts somewhere else

External Contract

- The same was also true for Course_type with text boxes
- Something wrong with course page tab order – didn't retrieve comment from database
- Failed save under firm create, but that was because they changed the date
- Person page has wrong tab order – problem saving comment because it erases existing comment and starts weird place
- Register – tab order for second register pages
- Overview page – they screwed with special cases until it finally didn't work
- They really liked the print page

We apparently didn't test thoroughly enough with our own functional test. We tested the original functional test but not the comprehensive one.

Next meeting at 9:00 next Tuesday.

10/24

- Today and every day from now on, will be dedicated to report augmentation
- We first made problems and solutions list
- We then took a long time to discuss presentation possibilities. So far we concluded that it should consist of:
 - Intro – Harvey
 - Demo – Harvey
 - Testing – Carsten
 - Architectural design – Jama
 - Refactoring – Paul
- After lunch, Harvey looked at requirements from iteration 0 and updating appendix
- We discussed a list of things that needed to be addressed in the document and did them

Meeting with Claudius

- Overall impressive, good
- Overview is still problematic
- Missing numbering of chapters
- Loses overview because diagrams needed an explanation of why there are no diagrams before this point
- Evaluate initial risk list in conclusion chapter
- Say something about XP and DB design
- Liked tools and technical prototyping
- Some kind of paragraph in between tools and technical prototyping
- Evaluation of tools in the conclusion of that section and a better conclusion – more specific to prototyping

External Contract

- Client/server chapter – no “what is a tear” – re-evaluate
- Recommence collective chapter for diagrams

10/25

- We talked about adding a chapter called “the outcome”, which includes the diagrams and list of completed functionality
- We will send the following to Claudius on Monday:
 - The outcome chapter
 - The conclusion chapter
 - Iteration 3 chapter

10/26

- Last day – hooray!
- C finished data dictionary
- P wrote outcome chapter
- H and J worked with appendix
- Later, P and H worked on various document tasks and Jama tried to make class diagram with JVision
- Discussed one person cleaning up code over weekend
- Also stopped by the print lady but no-one was home
- C and J left early and H and P continued fine tuning

Things that were left for Monday:

- Carsten works on alternatives section in database chapter
- Finish iteration 3
- Add personal evaluations
- Do conclusion
- Finish survival kit
- Finish outcome chapter

4.0 STANDARDS

By establishing standards for writing our documents, program and database code we provide a proactive means of increasing the quality of deliverables vs. a reactionary approach of quality control, where quality is the result of controls and corrections. It is far simpler and requires fewer resources to build-quality as compared to fixing for quality. In the following section there are descriptions and templates for standards.

CODING STANDARDS AND CONVENTIONS

Having a clearly defined ruleset for code eases the overview and understanding for developers as well as other interests.

HTML

W3C's XHTML 1.0 or HTML 4.01.

JAVA

Java code has been writing according to "The Java Language Specification, second edition" and the extract hereof, "Java Code Conventions".

SQL

There does not really exist any specific rules for how to write SQL. The online books installed with SQL Server does, however give some ideas of how it should be done.

```
CREATE TABLE Person
(
    person_no          SMALLINT NOT NULL IDENTITY(1,1)
    customer_no        SMALLINT NOT NULL ON DELETE NO ACTION,
    name_              VARCHAR(30) NOT NULL,
    comment            VARCHAR(50),

    PRIMARY KEY(person_no),
    FOREIGN KEY(customer_no) REFERENCES Firm(customer_no)
);
```

*Chapter Title = Report
Title Format*

Report Purpose = body in BOLD

4.1 Section Heading = Heading 1

HEADING 2

This is body text

Insert a reference as author last name, page (Shneiderman, 150)

Footnote¹

SubSection body BOLD

Add a Format A table. Heading Row in Black 50%.

Table Heading Text Arial 10 in white

TABLE 17. Table head

Title 1	Cell Heading Text Arial 10 in white
cell body	

FIGURE 8. Use Figure format

FIGURES

To stretch figure to full page width, right click in Anchored Frame, select Anchored Frame, de-select CROP and Apply.

1. See Appendix Contracts

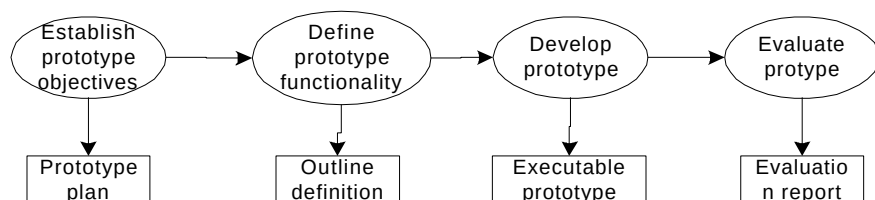
5.0 PROTOTYPING

Prototyping is an interactive technique in which users are actively involved in the mocking-up of screens and reports. The purpose of a prototype is to show the end-user the possible steps for the system.

Prototyping is not a suitable technique for all types of application projects according to their size and topic. The topic and the time frame of our project is applicable to use the prototyping as methodology.

STAGES OF PROTOTYPING

- **Objectives:** The objectives of prototyping should be made explicit, the presence of misunderstandings and misinterpretations of the expressed functionality. The output of this phase will be documentation of prototyping plan.
- **Functionality:** The next stage is to decide what to put into and, perhaps more importantly, what to leave out of the prototype. The output of this stage will be to outline a definition.
- **Development:** This stage will cover all tasks to implement the body of the prototype.
- **Evaluation:** This is the most important stage. It focuses on three basic issues: what is good about the prototype, what is bad about the prototype, and what is missing from the prototype. After evaluation of the prototype you have to either scrap parts, modify parts, and even add new parts or is no longer generating any new requirements.



APPROACHES TO PROTOTYPING

Consulting different literatures you will find different criteria of prototype classifications, which are based on the different purposes for which a prototype might be used. The following section describes different types of prototypes according to the latest literature¹.

- **Throw-away prototyping:** The principal function of this prototype is to clarify requirements and provide additional information for managers to assess process risks. The objective is to validate or derive the system requirements. You should start with

1. Software engineering 6th. edition by Ian Sommerville

those requirements that are not well understood because you need to find out more about them.

- **Evolutionary prototyping:** The objective is to deliver a working system to end-users. This means starting with the user requirements, which are best understood and which have the highest priority.

In addition to above mentioned prototypes you can find names of other kinds of prototypes, but most overlap in some way the two we have mentioned.

SELECTION OF AN APPROACH

It is generally not sufficient to take the existing situation as the one and only starting point for setting up PDS requirements. Additionally, we conducted several meetings with PDS to clarify the requirements. We found that despite our efforts we were still slightly unclear about the requirements, particularly with regard to the design of the user interface.

It was necessary to evaluate what type of prototyping would be most appropriate for our situation. We concluded that the choice would be evolutionary prototyping based on the following points:

- Time frame: We do not have the time to make a prototype that we will not use.
- Requirements: Major points of the requirements are well understood.
- Close similarity between evolutionary prototyping and XP, which will be our future process development.
- The production environment of the prototype is integrated with that of the final product.

There are two main advantages to adopting this approach to software development.

- **Accelerated delivery of the system:** Software support is made available quickly. In some cases, rapid delivery and usability is more important than details of functionality or long-term software maintainability.
- **User engagement with the system:** The involvement of users in the development process does not just mean that it is more likely to meet their requirements. It also means that the end-users of the system have made a commitment to it and are likely to want to make it work

5.1 Test and Evaluation

Forms and procedures used for evaluating the user interface prototype.

PROTOTYPE I

Internal Evaluation

Developers should split into 2 groups. Each group has a “user” and an observer. *Users* follow this script:

- Go to Course Administration index page

- Go to each link
- Simulate a form fill-in keeping mouse usage to a minimum by using the tabs
- Submit the form¹

Observers note the following by recording Yes/No and a number or description where applicable:

- Are all primary screens present according to navigation diagram?
- Are there links from Course Administration to all screens?
- Are there dead links?
- Does the tab jump to correct box according to a logical order as we have determined?
- How many mouse clicks were used to get to each page?
- Does the submit result in a return to the Course Administration page?

Experience and observations can be quite different depending on the point of view of the individuals involved in the evaluation. Therefore positions should be exchanged after the first run. Finally notes are to be compared between the groups and revisions made immediately .

Prototype I Evaluation Form

The first table that follows is an evaluation of the navigation between screens in the GUI. Each of the filled in tables is a composite of the results recorded by the 2 groups at the Developer and User evaluations. The first two tables are from the developer evaluation and the second set of tables is from the User Evaluation session.

TABLE 18. Prototype I Evaluation - Navigation

Test date: 20-08-01	Test environment RHS	User: Observer:	
Action	Link to page OK or reason not OK	Link to Home OK or reason	No. mouse clicks
• Open Course Admin	OK		
Go to links			
• Course create			
• Course edit			
• Course_type_create	Wrong File Name	Wrong file name	
• Course_type_edit			

-
1. Simulation with a link was used as there was no server connection

TABLE 18. Prototype I Evaluation - Navigation

• Location_create		Wrong file name	1
• Location_edit	Missing File		
• Participant create		Wrong file name for link	
• Participant edit	Wrong File Name		
• Firm create	OK	OK	1
• Firm edit	Missing File		

The next table displays the second form for Prototype I, UI Evaluation of the Form-fill in features.

Table 19: Prototype I Evaluation - Form fill-in

Form Fill-in:	Button	Label	Text Field/ Text Area	Layout
• Course create		Emphasize text for Label stating date separator is a ;		Fields and labels closer
• Course edit	Fortryd button not working			
• Course_type_create			Materials field should be TA	
• Participant create				Too much scrolling
• Firm create				Rt Col Label too close to lt TF. TF sizes should vary.
OTHER				All screens, vertical line too long

User evaluation

The same script and form for recording observations as for the internal evaluation was used. In addition the user had prepared his own criteria in advance, presented in the next table, to judge the GUI.

6.0 XP

6.1 Survival Kit

A DAY IN THE LIFE OF AN XP PAIR PROGRAMMER

- Retrieve replicable task card
- **BEFORE** you start coding, are all tests written?
- Does the test suite run without errors?
- Are all written tests added to the test suite?
- Follow task cards as much as possible
- Remember to write down refactoring ideas
- Remember to write down time started and ended
- Remember to add times to wall chart
- Do not change classes that were not understood to be changed by all group members without notifying the other pair
- Stop at 2:30! – don't start any new major code portion close to this time
- At the end of the day, re-run test suite on integrated code
- Update problems and solutions list when a new one has come acrossed

SCREEN SHOTS

FIGURE 9. Overview



Oversigt	Kursus nr.	Firma	TV	Deltager	CS	Status
Tilføjet	011101	MSW0800	3 dage	Microsoft Word 2000	Homo	8/10 deltagere
	011234	0000	1 dage	Basics	Homo	abc 1/10 deltagere
Personer	2000	navn		12345678	herbert	Forløb ikke startet
	123456	78900000	1 dage	Physical Therapist program	Homo	PPP 1/10 deltagere
Kurser	1000	Home Roskilde		49004900	dave	Forløb ikke startet
Kursusleder						
Kursusleder						
Udskriv						
Print						
Hjælp						

FIGURE 10. Person

The screenshot shows the 'Person' form in the Kursus Administration web application. The browser window title is 'Kursus Administration - Person - Microsoft Internet Explorer'. The address bar shows 'http://localhost:8080/entask/jsp/Person.jsp?customer_id=100'. The form has a sidebar with navigation links: Översikt, Tilmelding, Person, Kartor, Kartusledar, Kartusshjelp, Utgåv, Föreläs, and Hjälp. The main content area is titled 'Person' and contains a 'Säg till Kunde' button. The form fields are: Kunde (Marion Roskilde), Nytt kund nr (empty), Name (dropdown menu showing 'Valg for next person'), and Kommentar (dropdown menu showing 'Valg for next person'). There are 'Gem' and 'Tilbage' buttons at the bottom.

FIGURE 11. Register

The screenshot shows the 'Tilmelding' form in the Kursus Administration web application. The browser window title is 'Kursus Administration - Tilmelding - Microsoft Internet Explorer'. The address bar shows 'http://localhost:8080/entask/jsp/tilmelding.jsp?customer_id=100&customer_id=100&search=0&search=0&search=0'. The form has a sidebar with navigation links: Översikt, Tilmelding, Person, Kartor, Kartusledar, Kartusshjelp, Utgåv, Föreläs, and Hjälp. The main content area is titled 'Tilmelding' and contains a 'Säg till Kunde' button. The form fields are: Kunde (Marion Roskilde), Tilmelding (Algebra 2), Personer (4000 Roskilde, 4900 4900), Kartor (Name, Ordre nr., and Valg), Kartusledar (dane, 999, and Valg), Kartusshjelp (donald, 999, and Valg), Utgåv (Alice, and Valg), and Föreläs (Shedart, and Valg). There are 'Gem' and 'Tilbage' buttons at the bottom.

FIGURE 12. Kursus

FIGURE 13. Location

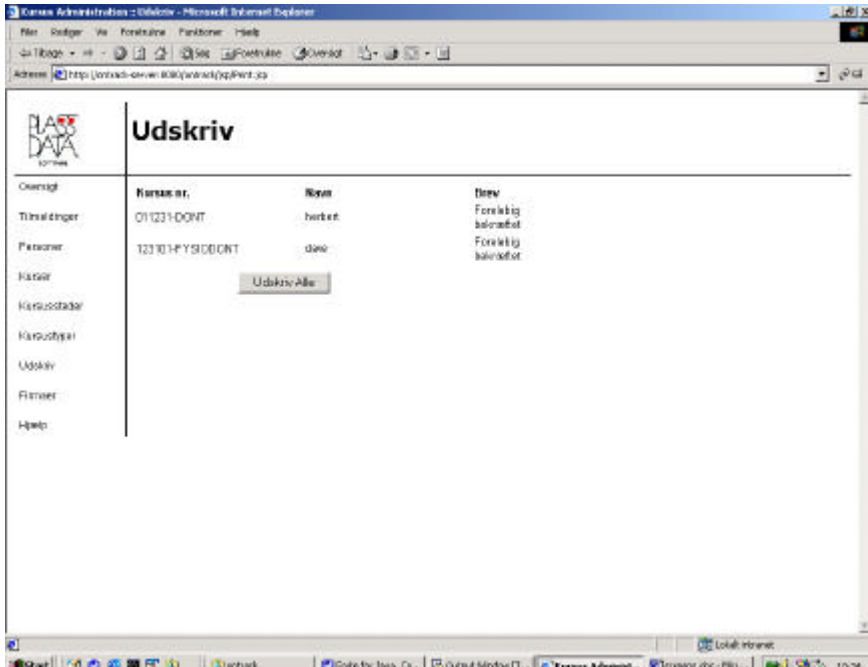
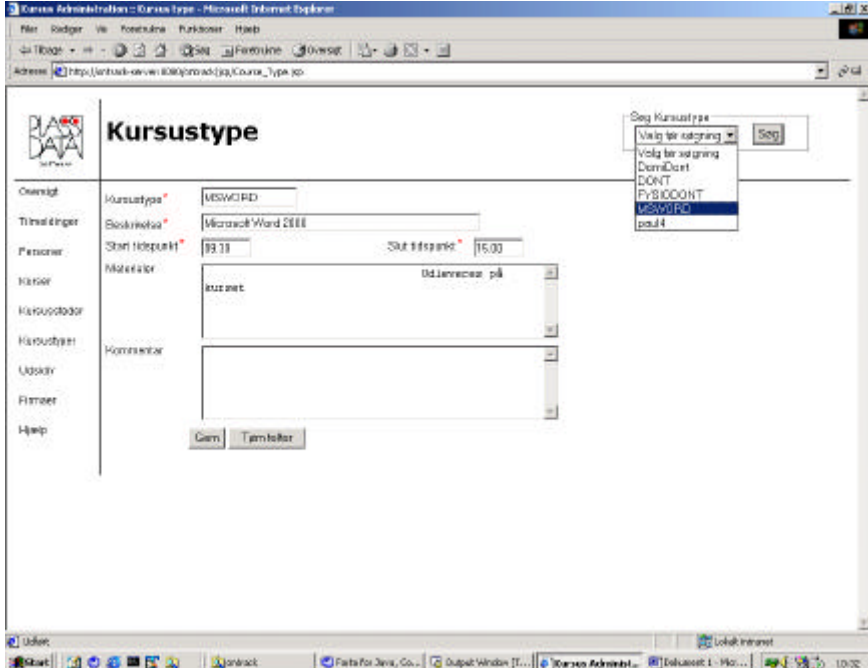


FIGURE 16. Firm

Kursus Administration - Firma - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://localhost:8080/portal/jsp/Firm.jsp

Firma

Sag Kunde/Firma
Kunde nr [2000] [Sag]

Geopgft: Kunde nr [2000]

Tilfjelling: Navn [newFirm]

Personer: Adresse 1 [newSt] TE [02145678]

Kursus: Adresse 2 [newSt2] Fax [07654321]

Kursussteder: Post st [4000] e-mail [new@new.dk]

Kursustyper: By [Roslunde] Kontaktperson [Jens]

Udskriv: [Gem] [Tilfjelling]

Primær

Hjælp

FIGURE 17. Main help

User Manual Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://localhost:8080/portal/jsp/Help.jsp

User manual online help

From this page and the available connections you will find out the necessary help to use this system.

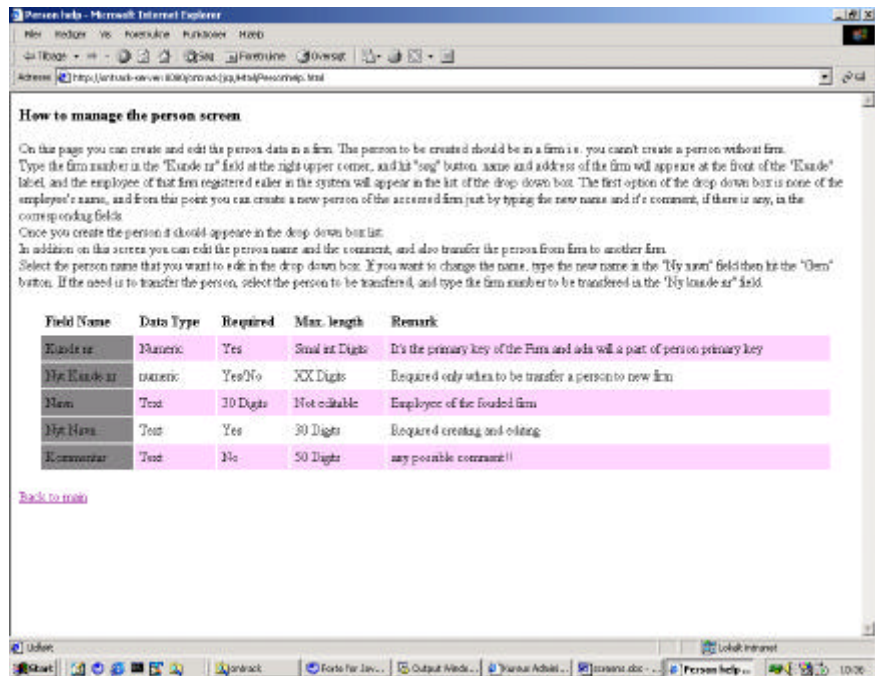
The help facility is based on the functionality of each screen. As user when you doubt what to do for each screen, don't waste your time to navigate and to look around different screens, just click the link for that screen and find out how to use it. Going the link of each screen you can find out data type, if it's required and whether it is required field or not of each screen.

To use this system with the following manual links you have to follow the [registration](#) guide.

- [Course type](#)
- [Course](#)
- [Exam](#)
- [Location](#)
- [Registration](#)
- [Printing](#)

[Back to main](#)

FIGURE 18. Person help



6.2 Data Dictionary for Database

TABLE 20. Tables

Name	Description
Course_Date	The specific dates for when a course is held
Course_Type	The type/kind of course that is held
Course	A course is an instance of a course type
Firm	The firm/customer that has bought software from PDS
Location	The location at which a course is being held
Participant	When a person is signed up for a course or has participated in a course, he/she becomes a participant
Person	A person employed by a firm
Standard_Letter	A generic letter sent to participants throughout the time from they apply for a course, until the course is held

TABLE 21. Course

Attribute	Datatype	Null	Default	Key
course_no	VARCHAR(30)	NOT NULL		PK
course_type	VARCHAR(15)	NOT NULL		FK ref Course_Type(course_type)
max_persons	SMALLINT	NOT NULL	Location (Max_persons)	
location	SMALLINT	NULL		FK ref Location(location_no)
room_no	VARCHAR(5)	NULL		
instructor	VARCHAR(30)	NULL		
comment	VARCHAR(250)	NULL		

TABLE 22. Course_Date

Attribute	Datatype	Null	Default	Key
course_no	VARCHAR(30)	NOT NULL		PK FK ref Course(course_no)
date_	DATE	NOT NULL		

TABLE 23. Course_Type

Attribute	Datatype	Null	Default	Key
course_type	VARCHAR(15)	NOT NULL		PK
description_	VARCHAR(30)	NOT NULL		
start_time	VARCHAR(5)	NOT NULL		
end_time	VARCHAR(5)	NOT NULL		

TABLE 23. Course_Type

Attribute	Datatype	Null	Default	Key
materials	VARCHAR(250)	NULL		
comment	VARCHAR(250)	NULL		

TABLE 24. Firm

Attribute	Datatype	Null	Default	Key
firm_no	SMALLINT	NOT NULL	IDENTITY ^a	PK
name_	VARCHAR(30)	NOT NULL		
address1	VARCHAR(30)	NOT NULL		
address2	VARCHAR(30)	NOT NULL		
postcode	CHAR(4)	NOT NULL		
city	VARCHAR(30)	NOT NULL		
phone	CHAR(8)	NOT NULL		
fax	CHAR(8)	NULL		
email	VARCHAR(30)	NULL		
contact_person	VARCHAR(30)	NOT NULL		

a. customer_no is generated in different thousands, depending on which customer group they represent, e.g. real-estate agents = 2000-2999, dentists = 3000-3999, etc.

TABLE 25. Location

Attribute	Datatype	Null	Default	Key
location_no	SMALLINT	NOT NULL	IDENTITY	PK
name_	VARCHAR(30)	NOT NULL		
address	VARCHAR(30)	NOT NULL		
postcode	CHAR(4)	NOT NULL		
city	VARCHAR(30)	NOT NULL		
phone	CHAR(8)	NOT NULL		
fax	CHAR(8)	NULL		
email	VARCHAR(30)	NULL		
www	VARCHAR(30)	NULL		
food	VARCHAR(250)	NULL		
overnight	VARCHAR(250)	NULL		
practical	VARCHAR(250)	NULL		
directions	VARCHAR(1200)	NOT NULL		
comment	VARCHAR(250)	NULL		
max_persons	SMALLINT	NULL		

TABLE 26. Participant

Attribute	Datatype	Null	Default	Key
course_no	VARCHAR(30)	NOT NULL		PK FK ref Course(course_no)
person_no	SMALLINT	NOT NULL		PK
order_no	VARCHAR(10)	NOT NULL		
state	VARCHAR(30)	NOT NULL	“Ej bekræftet”	FK ref Standard_Letter(state)
printed	SMALLINT	NOT NULL	1	

TABLE 27. Person

Attribute	Datatype	Null	Default	Key
person_no	SMALLINT	NOT NULL	IDENTITY	PK
firm_no	SMALLINT	NOT NULL		FK ref Firm(firm_no)
name_	VARCHAR(30)	NOT NULL		
comment	VARCHAR(30)	NULL		

TABLE 28. Standard_Letter

Attribute	Datatype	Null	Default	Key
state	VARCHAR(30)	NOT NULL		PK
letter	VARCHAR(4000)	NULL		

6.3 Data Dictionary for Java Classes

TABLE 29. Classes

Name	Package	Description
Course	ontrack.beans	A course
Course_Admin	ontrack.beans	The superclass for all the beans
Course_Date	ontrack.beans	The specific dates for when a course is held
Course_Type	ontrack.beans	The type of course that is held
Db	ontrack.util	Takes care of database connectivity
Firm	ontrack.beans	The firm/customer that has bought software from PDS
Location	ontrack.beans	The location at which a course is being held
Participant	ontrack.beans	When a person is signed up for a course or has participated in a course, he/she becomes a participant
Person	ontrack.beans	A person employed by a firm

TABLE 30. Course

Attribute/method	Data type	Comment
course_no	String	
course_type	String	
location_no	int	
room_no	String	
max_persons	int	
instructor	String	
comment	String	
assigned_persons	int	Calculated on the fly
date1	Date	
date2	Date	
date3	Date	
date4	Date	
days	int	The total number of days this course lasts
Course()	N/A	Constructor
Course(String course_no, String course_type, int location_no, String room_no, int max_persons, String instructor, String comment, int days, int assigned_persons)	N/A	Constructor w/ dates

TABLE 30. Course

Attribute/method	Data type	Comment
Course(string course_no, string course_type, int location_no, String room_no, int max_persons, String instructor, String comment)	N/A	Constructor w/o dates
Course(String course_no)	N/A	Constructor
setCourse_no(String course_no)	void	
getCourse_no()	String	
setCourse_type(String course_type)	void	
getCourse_type()	String	
setLocation_no(int location)	void	
getLocation_no()	int	
setRoom_no(String room_no)	void	
getRoom_no()	String	
setAssigned_persons(int assigned_persons)	void	
getAssigned_persons()	int	
setMax_persons(int max_persons)	void	
getMax_persons()	int	
setInstructor(String instructor)	void	
getInstructor()	String	
setComment(String comment)	void	
getComment()	String	
setDays(int days)	void	
getDays()	int	
setDate1(String date1_string)	void	
getDate1()	java.util.Date	
setDate2(String date2_string)	void	
getDate2()	java.util.Date	
setDate3(String date3_string)	void	
getDate3()	java.util.Date	
setDate4(String date4_string)	void	
getDate4()	java.util.Date	

TABLE 30. Course

Attribute/method	Data type	Comment
string2Date(java.util.Date date)	String	Converts a string date of the format 'YYMMDD' to a java.util.Date
date2String(String str)	java.util.Date	Converts a java.util.Date into a string date of the format 'YYMMDD'
date2SQLDate(java.util.Date date)	java.sql.Date	Converts a java.util.Date into a java.sql.Date
sqlDate2Date(java.sql.Date sqlDate)	java.util.Date	Converts a java.sql.Date into a java.util.Date
equals(Object obj)	boolean	Compares two Course objects
storeInDB()	void	INSERTs this Course in the Course table
getFromDB(String course_no)	Course	SELECTs the Course from the Course table that has this course_no
getAvailable()	List	SELECTs the course_no from the Course_Date table that are in the future only, if the Course last several days, it will only appear once
getOverview()	List	Same as above, but gets all the fields for each Course
getAllFromDB()	List	SELECTs all the Courses from the Course table
toString()	String	Returns course_no, course_type, location_no, room_no, max_persons, instructor

TABLE 31. Course_Admin

Attribute/method	Data type	Comment
db	Db	
error_list	List	Contains the errors found during methods
sql_string	String	The SQL string to be used on the database
bean_type	String	The name of the bean type to be processed
Course_Admin()	N/A	Constructor
updateDB()	void	Performs an INSERT via Db.SQLInsert()
getFromDB()	Object	Performs a SELECT via Db.SQLSelect()
getAllFromDb()	List	Performs a SELECT via Db.SQLSelectAll()
getIntFromDb()	int	Performs a SELECT via Db.SQLSelectInt()
getSQLString()	String	
setSQLString(String sql_string)	void	
getBeanType()	String	
setBeanType(String bean_type)	void	
checkErrors()	boolean	Return true if there are errors in the error_list, otherwise false
getError_list()	List	Returns the list of errors

TABLE 31. Course_Admin

Attribute/method	Data type	Comment
addToErrorList(String s)	void	Adds an error string to the error_list
getIter()	Iterator	Returns an iterator for the list of errors

TABLE 32. Course_Type

Attribute/method	Data type	Comment
course_type	String	
description_	String	
start_time	String	
end_time	String	
materials	String	
comment	String	
Course_Type()	N/A	Constructor
Course_Type(String course_type, String description, String start_time, String end_time, String materials, String comment)	N/A	Constructor
setCourse_type(String course_type)	void	
getCourse_type()	String	
setDescription_(String description)	void	
getDescription_()	String	
setStart_time(String start_time)	void	
getStart_time()	String	
setEnd_time(String end_time)	void	
getEnd_time()	String	
setMaterials(String materials)	void	
getMaterials()	String	
setComment(String comment)	void	
getComment()	String	
equals(Object obj)	boolean	Compares two Course_Type objects
storeInDB()	void	INSERTs this Course_Type in the Course_Type table
getFromDB(String course_type)	Course_Type	SELECTs the Course_Type from the Course_Type table that has this course_type

TABLE 32. Course_Type

Attribute/method	Data type	Comment
getAllFromDB()	List	SELECTs all the Course_Types from the Course_Type table
toString()	String	Returns course_type, start_time, end_time, materials, comment

TABLE 33. Db

Attribute/method	Data type	Comment
con	Connection	
driver	String	The JDBC driver
url	String	The URL to the SQL Server
username	String	
password	String	
Db()	N/A	Constructor
openConnection()	void	Opens the connection to the database
closeConnection()	void	Closes the connection to the database
getConnection()	Connection	Returns the connection
SQLInsert(String query)	void	Performs an INSERT specified by the query
SQLSelect(String query)	Object	Performs a SELECT specified by the query, and returns the first row of the ResultSet
SQLSelectAll(String query)	List	Performs a SELECT specified by the query, and returns all rows of the ResultSet
SQLSelectInt(String query)	int	Performs a SELECT which MUST use “AS integer” in the query, and returns this integer
SQLDelete(String query)	void	Performs a DELETE specified by the query

TABLE 34. Firm

Attribute/method	Data type	Comment
customer_no	int	
name_	String	
address1	String	
address2	String	
postcode	String	
city	String	
phone	String	
fax	String	
email	String	
contact_person	String	
Firm()	N/A	Constructor

TABLE 34. Firm

Attribute/method	Data type	Comment
Firm(int customer_no, String name_, String address1, String address2, String postcode, String city, String phone, String fax, String email, String contact_person)	N/A	Constructor
setCustomer_no(int customer_no)	void	
getCustomer_no()	int	
setName_(String name)	void	
getName_()	String	
setAddress1(String address1)	void	
getAddress1()	String	
setAddress2(String address2)	void	
getAddress2()	String	
setPostcode(String postcode)	void	
getPostcode()	String	
setCity(String city)	void	
getCity()	String	
setPhone(String phone)	void	
getPhone()	String	
setFax(String fax)	void	
getFax()	String	
setEmail(String email)	void	
getEmail()	String	
setContact_person(String contact_person)	void	
getContact_person()	String	
equals(Object obj)	boolean	Compares two Firm objects
storeInDB()	void	INSERTs this Firm in the Firm table
getFromDB(int customer_no)	Firm	SELECTs the Firm from the Firm table that has this customer_no
getAllFromDB()	List	SELECTs all the Firms from the Firm table
toString()	String	Return name_

TABLE 35. Location

Attribute/method	Data type	Comment
location_no	int	
name_	String	
address	String	
postcode	String	

TABLE 35. Location

Attribute/method	Data type	Comment
city	String	
phone	String	
fax	String	
email	String	
www	String	
food	String	
overnight	String	
practical	String	
directions	String	
comment	String	
Location()	N/A	Constructor
Location(int location_no, String name_, String address, String postcode, String city, String phone, String fax, String email, String www, String food, String overnight, String practical, String directions, String comment)	N/A	Constructor
getLocation_no()	int	
setLocation_no(int location_no)	void	
getName_()	String	
setName_(String name_)	void	
getAddress()	String	
setAddress(String address)	void	
getPostcode()	String	
setPostcode(String postcode)	void	
getCity()	String	
setCity(String city)	void	
getFax()	String	
setFax(String fax)	void	
getEmail()	String	
setEmail(String email)	void	
getWww()	String	
setWww(String www)	void	
getFood()	String	
setFood(String food)	void	
getOvernight()	String	
setOvernight(String overnight)	void	
getPractical()	String	

TABLE 35. Location

Attribute/method	Data type	Comment
setPractical(String practical)	void	
getComment()	String	
setComment(String comment)	void	
getDirections()	String	
setDirections(String directions)	void	
equals(Object obj)	boolean	Compares two Location objects
storeInDB()	void	INSERTs this Location in the Location table
getFromDB(int location_no)	Location	SELECTs the Location from the Location table that has this location_no
getAllFromDB()	List	SELECTs all the Locations from the Location table
toString()	String	Returns location_no, name_, address, postcode, city, phone, fax, email, www, food, overnight, practical, comment, directions,

TABLE 36. Participant

Attribute/method	Data type	Comment
course_no	String	
person_no	int	
order_no	String	
state	String	
printed	int	1 = true, 0 = false (for compatibility with SQL SMALLINT)
Participant()	N/A	Constructor
Participant(String course_no, int person_no, String order_no)	N/A	Constructor
getCourse_no()	String	
setCourse_no(String course_no)	void	
getPerson_no()	int	
setPerson_no(int person_no)	void	
getOrder_no()	String	
setOrder_no()	void	
getState()	String	
setState(String state)	void	
getPrinted()	boolean	
setPrinted(boolean printed)	void	
equals(Object obj)	boolean	
toString()	String	returns course_no, person_no, order_no, state, printed
storeInDB()	void	INSERTs this Participant in the Participant table
deleteFromDB()	void	DELETES this Participant from the Participant table
updateDB()	void	UPDATES this Participant in the Participant table

TABLE 36. Participant

Attribute/method	Data type	Comment
getFromDB(String course_no, int person_no)	void	SELECTs the Participant from the Participant table that has this course_no and person_no
getPrintList()	List	SELECTs the Participants from the Participant table that has not been printed (printed = 0)
getStates()	List	SELECTs the possible participant states from the Standard_Letter table
checkErrors()	boolean	Returns true if no errors occurred, otherwise false
getError_list()	List	Returns the list of errors

TABLE 37. Person

Attribute/method	Data type	Comment
person_no	int	Autogenerated in DB
firm_no	int	
name_	String	
comment	String	
Person()	N/A	Constructor
Person(int customer_no, String name_, String comment)	N/A	Constructor
setPerson_no(int person_no)	void	
getPerson_no()	int	
setCustomer_no(int customer_no)	void	
getCustomer_no()	int	
setName_(String name_)	void	
getName_()	String	
setComment(String comment)	void	
getComment()	String	
equals(Object obj)	boolean	Compares two Person objects
equals(int person_no, int customer_no, String name_, String comment)	boolean	Compares this Person with the fields of another Person
storeInDB()	void	INSERTs this Person in the Person table
updatePerson()	void	UPDATES this Person in the Person table
getFromDB(int person_no)	Person	SELECTs the Person from the Person table that has this person_no
getAllFromDB()	List	SELECTs all the Persons from the Person table
getFirmPersonsFromDB(int customer_no)	List	SELECTs all the Persons from the Person table that has this customer_no
fillSelect(int customer_no)	String	Build an HTML selectbox with Persons using the method above
toString()	String	Returns person_no, customer_no, name_ and comment

TABLE 38. Standard_Letter

Attribute/method	Data type	Comment
state	String	
letter	String	HTML formatted string
String getState() void setState(String state)		
String getLetter() void setLetter(String letter)		
void getFromDB()		Retrieves the bean with the name_ specified by setName_() from the Standard_Letters table
void getFromDB(String name_)		Retrieves the bean with the name_ given as parameter from the Standard_Letters table
void storeInDB()		Stores the bean in the Standard_Letters table
equals(Standard_Letter standard_letter)		Compares itself with the standard_letter given as parameter

6.4 Task cards

TABLE 39. Engineering task card - Show_Letter

Story number: _____	Task description: Show Letter	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Open letter in word • Call get person • Call get firm • Call get course • Call get course_type • Call get course_date (not yet implemented) • Call get location		• Uses existing participant bean • Set printed = 1 after printing and • updateDB()	
Software engineer's notes:			
Possible Test: • printed = 1 after printing • Participant is removed from the printing list (see Print task) Implement this task after course_date			
Things yet to do:			
Comments:			

Task cards

TABLE 40. Engineering task card - Overview 2

Story number: _____	Task description: Overview 2	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Loop - call get course - where course number is shown, generate link • List course only once, even if it lasts more than one day • Show the number of days a course lasts along with the course number etc.		•	
Software engineer's notes:			
Need to rearrange column order (sort by date) and other requests from last acceptance test			
Things yet to do:			
Comments:			

TABLE 41. Engineering task card - Course_Date

Story number: _____	Task description: Course Date	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
(Used on Overview and Register)		- Add date fields to Course bean - getFromDB() variant that takes the dates from the Course_Date table and the fields from the Course table - getFromDB() variant that only gets courses in the future (necessary for Overview and Register) - Methods for date conversion: - HTML form to java date and vice versa - Java date to SQL date and vice versa	
Software engineer's notes:			
Possible Tests: "yymmdd" string converted to SQL smalldatetime and vice versa - Future courses only shown - use GETDATE() in SQL			
Things yet to do:			
Comments:			

Task cards

TABLE 42. Engineering task card - Participant_Error_Handling

Story number: _____	Task description: Participant Error Handling	Task estimate: _____	Date: _____
		Software engineer: _____	
Errors			
• Search for non existing firm • Attempt to register for a full course • Try to register person to same course twice • Order_no must have value			
Software engineer's notes:			
Possible Tests:			
• Put Harvey at keyboard and observe if it catches all his errors			
Things yet to do:			
Comments:			

Task cards

TABLE 43. Engineering task card - Print

Story number: _____	Task description: Print	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Loop - Call get person, course using attributes from participant - Show course_no, name, state (list consists of participants where printed = 0) - Make person name link to Showletter.jsp along with necessary parameters		• uses existing participant bean	
Software engineer's notes:			
Possible error messages: • you must select one for printing			
Things yet to do:			
Comments:			

TABLE 44. Engineering task card - Person_Edit

Story number: _____	Task description: Person Edit	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Make default selection in person drop-down to “- Create new person -” • If “- Create new person -” has not been chosen, then update the person in context • Compare which fields have been altered		• UpdateDB()	
Software engineer's notes:			
Possible Tests: • Use equals method to verify change			
Things yet to do:			
Comments:			

Task cards

TABLE 45. Engineering task card - Register_Participant

Story number: _____	Task description: Register_Participant	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Call Get Firm data (name_ and phone) via customer_no. • If not found, give error message • Call Get list persons related to that firm • Call Get list of available, coming courses • Call Create participant bean for each participant that has been checked to attend a course • Call Store participant bean		• Create Participant bean with applicable variables • Set bean fields (also checks fields) • Get bean fields from DB (for testing) • Store in DB • (Equals x 2) for testing	
Software engineer's notes:			
Error messages: • required field empty • invalid inputs • exists already once committed			
Things yet to do:			
Comments:			

Task cards

TABLE 46. Engineering task card - Person_Create

Story number: _____	Task description: Person_Create	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Call Person_Create.JSP • Call get Firm parameter from form • Call set Firm bean • If not found then error message • New form? • Call open DB • Call get bean fields from DB • Call show applicable bean fields (name, add, tel., customer_no.) • Call get parameters (name, customer_no., comment) • Call set Person bean (name, customer_no., comment) • Call store in DB • Call close DB		• Create Person bean with applicable variables • Set bean fields (also checks fields) • Get bean fields from DB (for testing) • Open DB • Store in DB • Close DB • (Equals x 2) for testing	
Software engineer's notes:			
Error messages: • required field empty • invalid inputs • exists already once committed			
Things yet to do:			
Comments:			

TABLE 47. Engineering task card - Overview 1

Story number: _____	Task description: Overview 1	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Call get list of coming courses • Loop: Call get a specific course • - Get dates and location • - Print: course_no + dates + description_ + location + instructor + current/max persons		• Get list of coming courses	
Software engineer's notes:			
Error messages: • required field empty • invalid inputs • exists already once committed			
Things yet to do:			
Comments:			

Task cards

TABLE 48. Engineering task card - Course_Type_Create

Story number: _____	Task description: Course_type_Create	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Call Course_type_Create.submit.JSP • Call get parameters from form (course_type, description, materials, comment, time period) • Call set bean (course_type, description, materials, comment, time period) • Call open DB • If PK exists, generate HTML error message • Call store in DB • Call close DB		• Create Firm bean with applicable variables • Set bean fields (also checks fields) • Get bean fields from DB (for testing) • Open DB • Store in DB • Close DB • (Equals x 2) for testing	
Software engineer's notes:			
Error messages: • required field • empty invalid inputs • exists already once committed			
Things yet to do:			
Comments:			

Task cards

TABLE 49. Engineering task card - Location_Create

Story number: _____	Task description: Location_Create	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Call Location_Create.JSP • Call get Location parameters from form (name, add, post, city, tel., fax, e-mail, www, food, overnight, practical, comment, directions) • Call set Location bean (name, add, post, city, tel., fax, e-mail, www, food, overnight, practical, comment, directions) • Call open DB • If PK exists, generate HTML error message • Call store in DB • Call close DB		• Create Location bean with applicable variables • Set bean fields (also checks fields) • Get bean fields from DB (for testing) • Open DB • Store in DB • Close DB • (Equals x 2) for testing	
Software engineer's notes:			
Error messages: • required field empty • invalid inputs • exists already once committed			
Things yet to do:			
Comments:			

Task cards

TABLE 50. Engineering task card - Firm_Create

Story number: _____	Task description: Firm_Create	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Call Firm_Create.submit.JSP • Call get parameters from form (customer_no, name, add 1, add 2, post, city, tel., fax, e-mail, contact person) • Call set bean(customer_no, name, add 1, add 2, post, city, tel., fax, e-mail, contact person) • Call open DB • If PK exists, generate HTML error message • Call store in DB • Call close DB		• Create Firm bean with applicable variables • Set bean fields (also checks fields) • Get bean fields from DB (for testing) • Open DB • Store in DB • Close DB • (Equals x 2) for testing	
Software engineer's notes:			
Error messages: • required field empty • invalid inputs • exists already once committed			
Things yet to do:			
Comments:			

TABLE 51. Engineering task card - Course_Type_Error

Story number: _____	Task description: Course_Type_Error	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• COURSE_TYPE.JSP: • Check if error messages exists in bean, and if there is write them out • Call submit page • COURSE_TYPE_SUBMIT.JSP: • Call applicable set-methods • Call store in db • If any errors exist, redirect to the course_type page		• Set bean fields and check for invalid/missing fields and write errors to a list in the bean • needs boolean method CheckForErrors() • Get bean fields from DB (for testing) • Store in DB • (Equals x 2) for testing	
Software engineer's notes: Error messages: • required field empty • invalid inputs • exists already once committed			
Things yet to do:			
Comments:			

Task cards

TABLE 52. Engineering task card - Course_Create

Story number: _____	Task description: Course_Create	Task estimate: _____	Date: _____
		Software engineer: _____	
JSP		Bean	
• Call Personal_Create.JSP • Call get_course parameters from form (course_no., course type, dates) • Call set Course_bean (course_no., course type, dates) • Call open DB • If PK exists, generate HTML error message • Call store DB • Call close DB		• Create Course bean with applicable variables • Set bean fields (also checks fields) • Get bean fields from DB (for testing) • Store in DB • Open DB • Close DB • (Equals x 2) for testing	
Software engineer's notes:			
Error messages: • required field empty • invalid inputs • exists already once committed			
Things yet to do:			
Comments:			

6.5 User Tests Iteration 2 and 3

TABLE 53. User functional test - Iteration 2

Test	Expected result	Result
CHECK DB AFTER EACH SUBMIT		
General		• Pages should notify somehow that you did not press slet felter • “Focus” for cursor after submit • Error messages should be identical in format using JavaScript alert boxes • Should not erase data when you put in the wrong text • Course select boxes should have default values already showing
Open the Course Overview	Data appears. Links not functional	• Move customer number in on overview • Move 2 out of 10 also to the right • No column headings on overview
Open Location		
Submit without entering name, directions	Error messages: Name missing, Directions missing	OK
Enter all, submit	Refresh to Location	OK, but if error on submit data should still be in form to prevent re-entry
Open Course_Type		
Submit without filling in Course_Type	Error message: Course_Type missing	OK
Submit without filling in Description	Error message: Description missing	OK
Fill in with existing course_type	Error message: Course_type missing	OK
Fill all in and submit	Refresh to Course_Type	• Keep duration as it is now with start and end_times
Open Course		

TABLE 53. User functional test - Iteration 2

Test	Expected result	Result
Submit without filling in Course no	Error Message: Course_no missing	OK
Submit Course	Refresh to Course	Links to the overview page did not work
Open Person		
Submit search for firm on person page without filling in a firm number	Error message: Customer number missing	OK
Submit search for firm on person page with a correct firm number	Firm name appears on form and list of persons in Select Box	OK
Submit Person without entering a person name	Error message: Person name missing	OK
Submit Person	Refresh to Person Page	Should be able in person to create multiple people for a firm without having to search for a firm every time
Open Participant		
Submit without selecting firm	Error message: Missing firm no	OK
Enter firm no and search	Firm data with person list	OK
Submit without entering Order no	Error message: Order no missing	OK
Submit after select first person	Refresh to Overview	Error message in register for max persons and should not already be in that course

TABLE 54. User functional test - Iteration 3

Test	Do following	Result
Open Course Overview	Click Location Link - Create	
Click Course_Type Link	Create	OK. course_type created

TABLE 54. User functional test - Iteration 3

Test	Do following	Result
Click Course Link	Create Search for the course	Tab order incorrect OK - data for comment and room no not shown (not implemented as part of the iteration) System dates long in future cause crash ^a
Click Firm Link	Create Submit a second time to provoke error message	Crash ^b . After debugging, all was OK Error message for PK in JavaScript Alert box
Click Person	Create Torben Benny Bendt	
Click Tilmelding	Register 3 persons to same course Search	After search focus should be on first person. Tab order incorrect
Check Overview for new participants	Click Link for the course the participants were added to. Delete participant from the course	OK
On Course Page	Change persons course. Open Course Overview Change a persons status. Check the change	OK
Click Print Link	Click the person Status was changed for	OK
Go back to Print Page	Check person is removed	OK
Click Person Link	Update a Name Check Drop down box for name change Change relation to firm Search under new firm and check drop down for that person Change comment – check in Database	OK Changing a name without changing comment deletes the persons comment. If statement missing in code

TABLE 54. User functional test - Iteration 3

Test	Do following	Result
Click Location Link	Create	Cursor jumps from Fax to Search Cursor position in text areas wrong Null in text fields for null values Using “ “ marks around submitted text causes that text not to show after a search

- a. Due to expiration date of JTurbo trial version
- b. At this point we broke off the test to fix the problem, which was a small change done the previous day. The error was not uncovered because we did not test thoroughly after the change and overconfidence. This emphasizes the need for having tests written and internal functional testing.

6.6 Problems & Solutions

Solutions are discussed in the iteration section of the report following the iteration where the problems occurred.

TABLE 55. Iteration 1

Problem	Solution
Contradicting interests - Customer = Software/ School = report	Balance work and goals
DB inconsistency	Single DB. Use data dictionary
Debugging	Change one and only one thing a a time, check result. Move in when sure the problem is not here
Difficulty thinking simple	Train way of thinking
Everything works on each computer, not when combined	Daily integration exactly as XP suggests. Team that wants to integrate code, does it. All tests should run. If there is a problem, the integrating team fixes it.
Many new tools - did not allow enough time to learn	For future - plan more time for technical prototyping. Choose familiar tools when possible. Look for simplicity
Not familiar enough with JSP	More coding
Recording time usage	Chart on wall. Each team enters hours used each day on each task

TABLE 55. Iteration 1

Problem	Solution
Refactoring	Plan coding day - Xhours coding, X refactor, X integration so that all are finished by 14:45 at latest. Must refactor every day
Rusty with coding	More coding
Stick to the task	Stick with what is supposed to be done according to task cards, write down the good ideas for later reference
Things don't work when installing at company	Plan to and arrive early enough before the scheduled test, install, check and fix
Too many discussions are too long	Determine goal of a discussion. Set a time limit and use it. When is it time to decide and move on. When has every sensible point been discussed?
Too slow to act	
Underestimated time needed / Overly optimistic about how fast we can produce results	Use more common sense and experience
Versions	Daily ACE file created with our filenaming convention
Writing tests	List what tests we want and write them. If ideas for new ones come up, write them
Writing the code	Test a little code a little - the now famous XHSC method - make one text box work, when it works take another, when it works - another, if the rest should function in same manner add all, otherwise one at a time Test SQL in Query analyzer - then in code

TABLE 56. Iteration 2

Problem	Solution
Difficulty determining errors	Write out exceptions
Downloaded software trial version expired making it impossible to connect to DB and resulting in a loss of about 2 hours time.	Write out exceptions
Forte is an unstable environment	Check other IDE's and evaluate the risk of changing at this late date in project or accept the bugs and report to SUN.

TABLE 56. Iteration 2

Problem	Solution
Inconsistencies in data types, exceptions, errors, other difficulties during coding	Communication. Create a Survival Kit - a reference for programming which includes Data Dictionary, Screen Shots, Task Cards - both the actual and empty ones, A Problem and Solution list.
Testing	List tests required and write them
Version control	Allow more time for integration of files at end of day. Concentrate on this point

TABLE 57. Iteration 3

Problem	Solution
Cursor Position in text areas	No white space between Text Area start and end tags
Tab order in forms Not everything tested internally prior to user test	Internal functional testing, correct and no changes immediately preceeding user functional tests
Dependency on unit tests	Don't rely completely on their success. If a method doesn't have a unit test, your tests don't let you know if the method works!! SO either unit tests for everything or both unit and other test types needed.
Underestimating coding capability	When a unit that worked yesterday or a minute ago starts malfunctioning, believe in the codes quality and don't start changing it to fix an error. There is a good chance the problem is technical in nature, such as a malfunctioning server or IDE.
Forte "broken file" error	We don't know what causes this to suddenly occur fairly often, but deleting the class and temporary .java~ files often solves problem. If necessary stop/restart Forte.
School net down during autumn break	Discuss with sys admin
Task estimates way off	More experience estimating = always estimate tasks
Dynamic test data	Static data so it is not always necessary to change test code
Working on one DB via net	Use independent 100% equal db's in regard to their design
Session beans	Provide different id names if it is necessary to have several instances of the same class in sessions

7.0 INSTALLATION GUIDE

Running the onTrack course administration and participant progress tracking system, requires Java and a web server capable of running JSP pages. Following is an installation guide for

- Tomcat 3.2 (running through Forte 3.0 CE)

GENERAL PREREQUISITES

- Java 2 Standard Edition v1.3.1
Follow the standard installation procedure, and add "C:\jdk1.3.1\bin" to your PATH.
- The code
Unpack the files from X:\myjava.exe (where X is you CD-ROM drive) to C:\myjava.
- SQL Server 2000
While installing, you are asked which kind of authentication you wish to use. Make sure you click "SQL Server Authentication".
When the installation is done, start the Enterprise Manager.
Click on the server, and expand the tree.

Creating the database

Click on the "Databases" folder, and right-click in the right window and choose "New database...". Enter the name "PDS" and click Ok.

Creating the user

Click the "Security" folder, then the "Logins" folder, and right-click in the right window and choose "New login...". Enter the name "ontrack" and click on the "SQL Server Authentication" and enter "ontrack" in the password field below. In the "Database" select box in bottom, choose the PDS database you created before. Click on the "Database access" tab in the top. Check "Permit" for "PDS" and in the "Database roles" check "db_owner".

- JTurbo 2.0
Unzip the file. The placement of the JTurbo.jar file depends on which web server you are using, so you will be told later what to do.
- Classpath
Add "C:\myjava;C:\myjava\JTurbo.jar;" to the CLASSPATH.
If you wish to use JUnit for testing, also add "C:\myjava\junit.jar".
Remember to separate all files/directories with ";".

**TOMCAT 3.2 (RUNNING
THROUGH FORTE 3.0 CE)**

Prerequisites

- The above mentioned prerequisites
- Forte 3.0 Community Edition

Copy the JTurbo.jar file to "C:\forte4j\lib\ext" folder.

At the first run of Forte after the installation, you are prompted to specify your working directory. Make this C:\myjava.

When Forte is done loading, Go to the "File" menu and choose "Mount filesystem...". Click the "Add local directory", enter "C:\myjava" and click "Ok".

Click the "Editing" tab just below the menu bar, and then click in the "Explorer" window.

Now expand all directories in your way to "C:\myjava\ontrack\jsp".

Double-click on "Overview.jsp" to view the code in the "Source Editor" .

Now click on the "Execute" button below the menu bar (looks like "play", you can also press F6).

Your browser now opens, and the system is running

8.0 SOURCE CODE

8.1 Java

8.2 JSP and HTML

8.3 SQL
