

Incremental development for AXE 10

Lars Taxen* and Even-André Karlsson**

Abstract

To become more reactive and flexible to the market needs, i.e. shorter lead-times and more flexibility in handling late and changing requirements, Ericsson's AXE 10 development processes have been adapted to support incremental development. The basic principle is to develop each of the customer features in one executable and testable increment. In order to plan, coordinate and control projects executed in this way, an Incremental Development Method Package, IDMP, has been developed by UAB*** in close cooperation with support groups in development projects. In this paper, we define what we mean with incremental development and describe the main aspects of the package. We also report experiences from using the package in the CMS 30 phase 7 and HELIOS projects. A prototype tool which supports the package is described, together with the experiences from using the tool in the pilot project Combined Gateway. Even if the IDMP has been developed towards AXE 10, it has proven useful in non AXE 10 projects, as the principles are rather independent of development process.

1. Introduction

The telecommunication market is changing very rapidly mainly due to two forces: The deregulation with the entering of many new operators leading to more competition, and the proliferation of new technology, e.g. mobile communications, intelligent networks etc. This has put a demand on the suppliers to be more reactive and flexible to the market needs, i.e. shorter lead-times and more flexibility in handling late and changing requirements. This is a very challenging change considering the size, complexity and in service performance (up-time) requirements of telecommunication systems.

Ericsson has met this challenge by using some type of incremental development in many large projects. Experiences from these projects have been consolidated into a method package for incremental development, which is an extension of Ericsson's existing development processes.

2. What are increments?

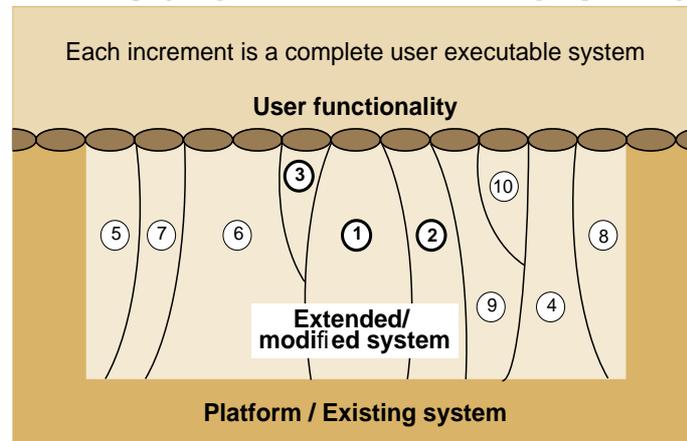
An increment is a well-defined, testable and rather independent functionality in the final system. An increment is preferably a feature offered to the customer. A group of increments is packaged in a build which together form a new executable system. The sequence of builds will provide a system with growing functionality. Usually the projects are organized in 3-6 internal builds with 1-2 month intervals before the system is finally delivered to the customer and put into operation, but we also have examples of intermediate deliveries to the customer.

*Ericsson Utvecklings AB, S-125 25 Älvsjö, email: Lars.Taxen@uab.ericsson.se

** Q-Labs, S-22 370 Lund, email: Even-Andre.Karlsson@q-labs.se

*** UAB has been responsible for the standard AXE 10 development Medax

The figure below illustrates the principle, in this case how 10 customer features are implemented to fulfil the project goals. The 10 features can be grouped in e.g. 4 builds, with



feature 1, 2 and 3 in the first, 4 and 6 is the second, 5 and 7 in the third, and 8, 9 and 10 in the fourth and last.

3. Customer benefits

Incremental development provides:

- Customer focus by emphasizing the customer features. The customer can also provide feedback on the intermediate builds.
- Requirements flexibility by allowing changes to features in later builds and also keeping the feature focus during each build.
- Reduced risks by having a system with gradually growing functionality.
- Early feedback through each build, both to the customer as well as the designers. In particular the continuous system test activity with feedback to design has proven beneficial for the final performance and quality of the system.
- Shorter lead-time by the overlap and concurrent work between different builds.

4. Evolution of the Incremental Development Method Package

Experiences from ERA, EED, ETO, EPA, MET, ETM, EIN and others, were consolidated into a requirement specification for an incremental development method package in the beginning of 1996. During 1996, the package was developed at UAB in close cooperation with the method group from CMS 30 phase 7 at ERA and the TTM 15 initiative associated with the HELIOS project at ETX. The R1 version of package was released late 1997.

The package was applied in CMS 30 phase 7, switching subsystem. The principles of the package was used in HELIOS. The experiences from these projects are described in detail in chapter 6. One major result was that the package needed tool support. Thus, a prototype was developed during 1997 in close cooperation with several pilot projects. This is further described in chapter 7. During this development, the construction planning in the package was modified. The rest of the package will be updated to release R2 during Q2 1998.

5. Main aspects of IDMP

In this section we go through the main aspects of the IDMP:

- Construction planning
- PROPS application
- Configuration management

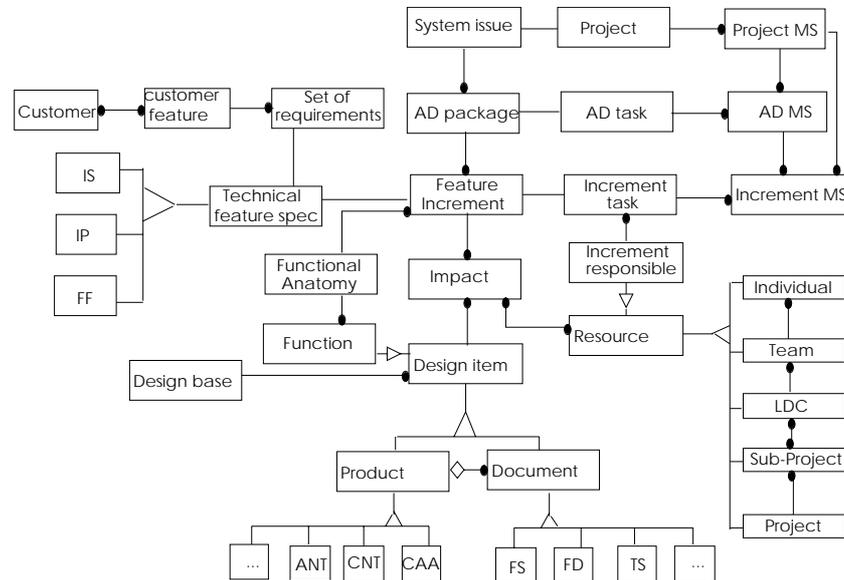
- Adaptation

Finally, we summarize the content of the IDMP.

Construction Planning: A carefully prepared construction plan is essential for a successful incremental development project. The construction planning process determines the possible increments, allocate them to builds and schedules the builds as well as the individual increments in time. The process is designed to satisfy:

- Gradually detailing in planning through the earlier phases of the project.
- Distribution of responsibility, both horizontally from total project to teams and vertically between subsystems.
- Adaptation of the model and processes to the needs of specific projects

The construction planning is based on a conceptual model, which describes the important concepts for incremental development, and how they relate to each other. This model has gained acceptance as the standard model for incremental development of applications within a large design community at BN and BR. The conceptual model is to a large extent independent of development model.

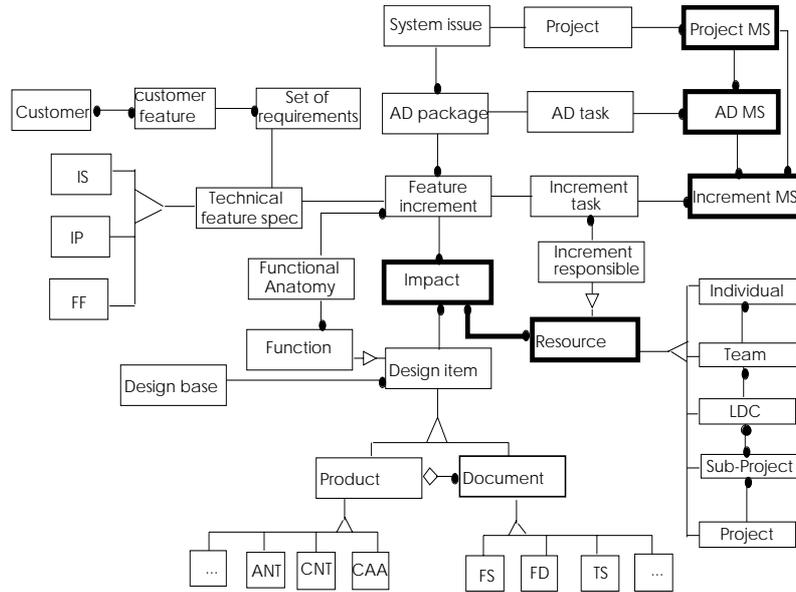


The construction planning is roughly split into the following phases:

- *Functional analysis*, where the number, scope and order of the increments are determined. Also, preliminary number of AD's and dates for these are set
- *Construction planning - Logical*, where the increments are allocated to AD's.
- *Construction planning - Real*, where the milestone dates are determined based on effort and resource estimation.
- *AD planning*, where the contents of the test dumps are determined.

The conceptual model is used to indicate what information is produced in each phase. As an example, during the Construction planning - Real, the impacts from each increment on

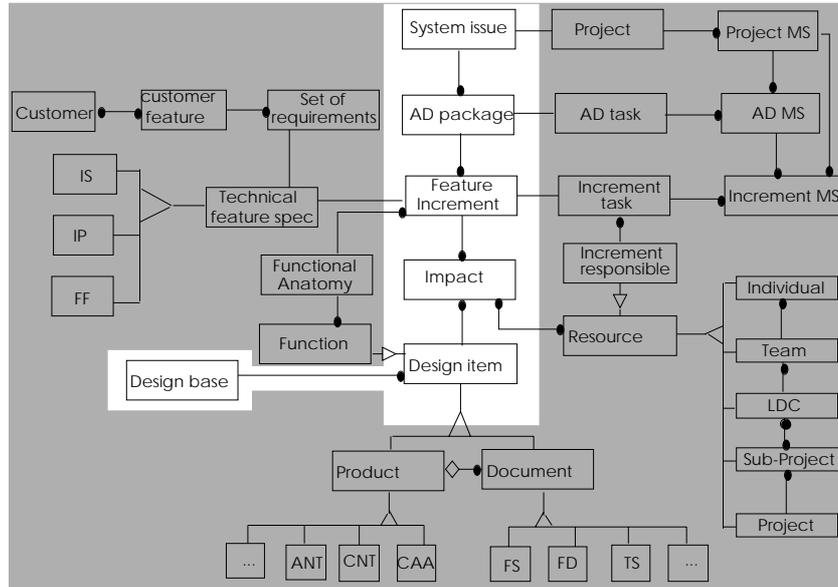
design items (products and documents) the information concerned is indicated as in the figure below:



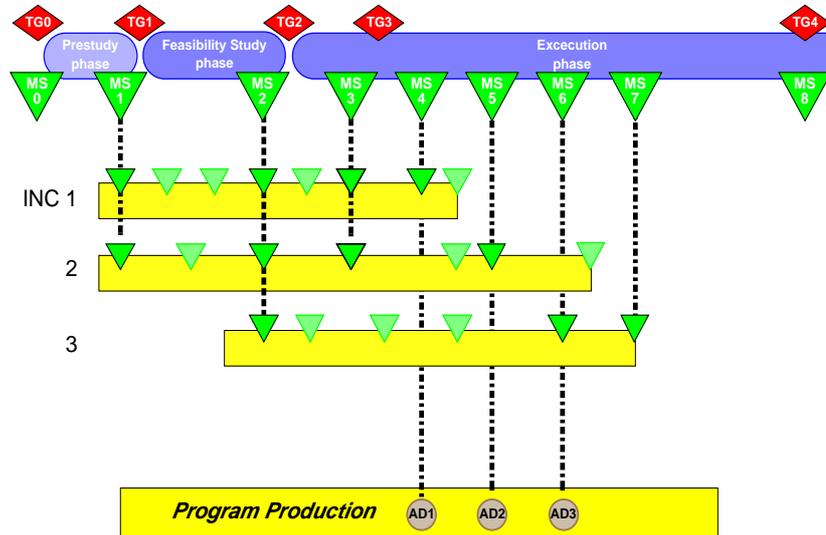
We also use the conceptual model to indicate what information is documented in different documents. For example, in the Increment Impact Matrix, we document what design items are impacted by each increment. An example is found below:

							AD		AD 1		AD 2			
							Increment Name		Flexible Charging		Software Supervision		Statistics Improvements	
							Increment Responsible		LMF		LMF		EPL	
							Tech. feature spec / type		IP 15941		IP 15941		IP 15941	
							Tech. feature spec / nbr		101/FCP 101 006		102/FCP 101 006		103/FCP 101 006	
							Tech. feature spec / rev		PA2		PA5		PA6	
							Total effort per incr.		364		521		250	
Subsystem	Resp.	Prod / Doc Type	Prod / Doc Number	Prod / Doc Title	Base rev.	Total effort per item								
ANT210..	LMF MKY	CNT	2382037	HRON	R2	150	45 / R3				105 / R4			
ANT210..	LMF CAK	CAA	179506	HSD	R5A	500	100 / R6A		200 / R6A		200 / R6B			
ANT210..	EPL PED	CAAZ	1072179	HCITOP	R1A	50	50 / R2A							

The Increment Impact Matrix can be looked upon as a projection or view of the following parts of the conceptual model:



PROPS Application: For incremental development projects there are milestones on two levels: the project level and the increment level. The project level milestones must be defined for each project, and they should be connected to information necessary to take the necessary business decisions. If the development method for each increment is standardized, the definitions of the increment milestones can be the same for all increments. If the feature



increments are short it is recommended to leave out some of the milestone definitions to avoid having too many milestones. Also for feature increments which are developed synchronously towards the same build, we could use the same milestones.

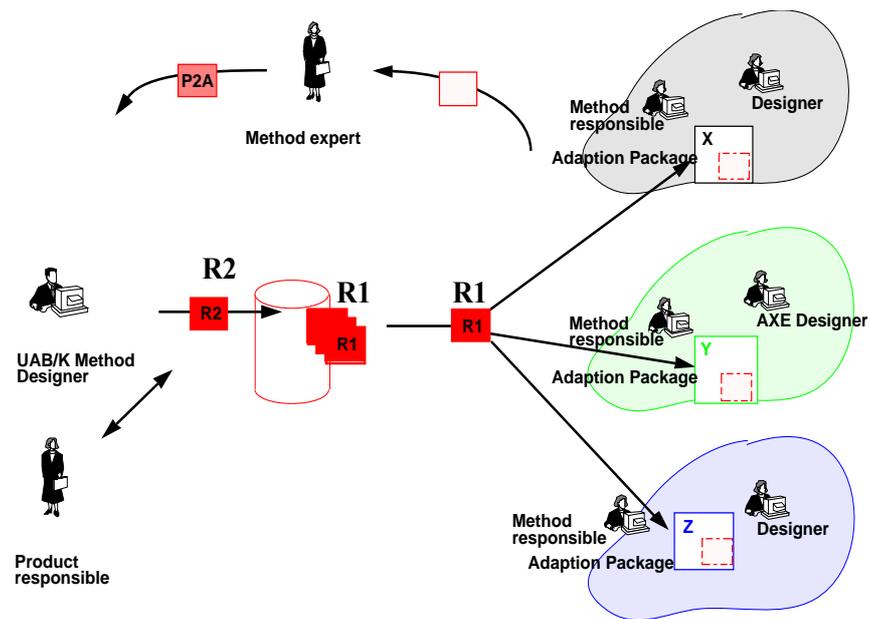
Configuration Management Rules: The purpose of these rules are to make it possible to plan the increments in the sense that we must know what revision of documents and products belong to which increment. The basic principle is to step the revision letter for a design document for each increment impacting the document.

Standard vs Adaption: The package is a standard package, which is meant to be adapted to local needs at LDC's / projects. The adaptations needed are usually minor. This principle of adapting a standard product¹, has some very attractive consequences:

- The project will get an adapted method (and tool support) very fast.
- A specific adaption does not have to be coordinated with any other adaption.
- The project will feel that they have contributed to the method (and tool) properties, thus the acceptance of the support of will be easier.
- It makes it possible to balance between a collective asset (the standard product) and individual assets (the adaption)
- The maintenance of the combined standard - adaptations product will be less compared to the case where each LDC / project maintains its own support product.

The experiences found in adapting the standard product in many projects, can be incorporated in later revisions of the standard product. Thus, the combined standard - adapted product will be steadily improved:

1. This applies to the tool support as well, see chapter 7.



ID Method Package Product Description: The Incremental development method package contains the following elements:

- Incremental development for AXE 10 - General Description
- Incremental development for AXE 10 - Wall-chart
- Definition and planning of increments, work instruction
- Preparing the Increment Specification Documents, work instruction
- Configuration management in Incremental Development, work instruction
- Incremental development with Medax and current tools, work instruction
- Increment dependency matrix, document instruction
- AD (Build)-plan, document instruction
- Functional Anatomy Description, document instruction
- Increment Task Specification, document instruction
- OH-slides and Teacher's Guide

In addition the following documents are provided:

- Incremental development—Guidelines and experiences
- Incremental development and PROPS (Ericsson's project management process)

This packages was developed in cooperation with several pilot projects, and was released for general use late 1996. It is based on experiences from about 10 incremental development projects.

The method package has been supported by a series of seminars during it's introduction. The largest one attracted 90 people, and was a three days seminar with the first day focusing on the methods, the second day covering project experiences and the last day with work-groups looking into special aspects of incremental development, e.g. daily builds, project planning and tracking, configuration management, tool support, etc.

6. Experiences

6.1. The HELIOS project

HELIOS was large development project (475 000 man-hours) that run between March 96 to Sept. 97. It was comprised of nine subprojects distributed over seven countries in Europe. It had about 180 feature increments (called assignments) packaged into three builds.

HELIOS decided to go for incremental development early in the project, but there was a lot of discussions about how to use it. Some of the results were:

- The early definition of the Logical Construction Plan was very useful. It helped a lot in planning, and gave a lot of structure to the project.
- Difficult features in early builds was good.
- Incremental development cut the lead time, but increased cost due to more administration when blocks were opened in more than one build. Also, the review effort increased.
- It was very difficult to follow up the project in the traditional way. They tried to see each of the builds as a small project with all the milestones. This led to a milestone passage about every second week. In addition to that, a lot of changes had to be handled during the project. The result was that it became impossible to follow up the project in the traditional way. To cope with this, HELIOS decided to follow up on the main project level only those documents needed by others. The rest of the follow up was left to the responsibilities of the subprojects. They also tried to synchronize the main deliveries from the subprojects.

6.2. The CMS30 phase 7 project

This was a smaller project of about 60 000 man-hours distributed to two design centres i Sweden and Finland. The purpose was to deliver 25 features to the Japanese mobile operators. It ran for about 6 months. Two features were removed and five new features added during the project. Some of their results were:

- Changes were easy to plan. Adding and removing features was quite easy.
- Much better control of costs: the cost of each feature was known.
- Neither MS3 nor MS4 could be used. ID needs a totally new approach to milestones. Conventional model of milestones must be totally forgotten!
- Passing tollgates was a major head-ache. It was very hard to pass a tollgate, especially TG2 (where the project scope shall be known in all aspects) if you only can foresee the future 3 months ahead due to the contextual insecurity.
- PRIM needs to be updated for each increment (otherwise writing a TR would be impossible). This caused a lot of extra effort.
- SIGMA handling is tricky.
- No tool support for configuration management caused a lot of manual work.
- Lots of inspections!
- There can not be just one TG2 in turbulent projects like this one.
- TG3 placement is even harder than TG2.
- Using one IP as an increment specification helps project planning, but caused too much inspection overhead.
- Confessing that the project you start with is not the same project you finish with is a step forward. ID can be seen as a confession to this.

7. Tool support for managing incremental development

The purpose of the tool support is to manage all information that is needed to plan, coordinate and control incremental development projects. This will give us the possibility to work in a different manner where information rather documents are managed. For example, it would enable the generation of reports from the information when needed, rather than maintaining and updating documents.

During the second half of 1997, pilot projects at EPK and ETM were started to get feedback from LDC's on the IDMP and to collect requirements for a tool supporting the construction planning of increments. In parallel, selected requirements were implemented and tested by building a prototype called CPIprot (Construction Planning prototype). Prototyping promised to be the best approach to get useful, reliable and fast feedback from the pilots and other projects. After the first version of the prototype was available (around October 1997), it was presented to the pilot projects at EPK and ETM. In addition several demos at different LDC's and at the 2nd Advanced Incremental Development Seminar in Stockholm have been given. Also, AMC phase 6 has decided to be a pilot, and JDI has also show interest in the tool.

The prototype was developed in cooperation with the company Technia AB and is build upon a commercial object-oriented PDM (Product Data Management) system called 'Matrix' (developed by Matrix-One Inc.). The Matrix platform was chosen because it is very flexible, has a powerful script language and has a graphical and a web interface. Furthermore it has a client/server architecture and so supports the distributed projects at Ericsson. It is also compliant with the CORBA standard.

The conceptual model is directly implemented in the tool, which makes it possible to browse the information exactly as it is shown in the conceptual model. Since each project has its own characteristics and different ways of working, the conceptual model and tool support has to be adapted to each project. These adaptations are rather small and can mostly be done quite fast. The adapted model is implemented in the prototype which is easily possible, due to the fact that the platform of the prototype is object-oriented.

A sample of the features implemented in the prototype are:

Basic

- Implementation of the IDMP model
- Manipulating the IDMP model
- Manipulating instances of the IDMP model
- Entities of IDMP model graphically represented
- Version mgmnt according to IDMP for docs
- Browse all information through a web interface
- Define and change status of products and docs

Product Definition

- import design base from PRIM
- product structure definition

Dependencies

- Trace customer features, requirements, assignments, IS's, IP's, FF's
- View Functional Anatomy information

Web based Cost Table Editor

- Enter cost of implementation data per design item

MS Project Integration

- Effort estimates to MS Project, time and resources back to CPLTool

Increment Impact Matrix

- Produce different views of the Increment Impact Matrix as a HTML file

Reporting

- List of impacted FS's, blocks and docs per feature increment
- Produce 1317-list for PL-GAS
- Produce AD plan (Dates, test objects and SW units)

Milestone Def and Checking

- Definition of MS's on document type- and doc. level

Currently we see the following roles which will use and benefit from the tool:

- Project Manager
- Project Administrator
- Construction Planer
- Configuration Manager
- IPFF writer
- Developer/Team
- Test Configuration Management
- Methods&Tools Responsible

The name of the roles can slightly differ in projects but the tasks are the same.

Combined Gateway - Pilot project at EPK: Currently, the prototype is at use in the Combined Gateway (CGW) project, which is an assisted project to AMC phase 5. TG1 was May 1997, design started January 1998 and TG2 was at the end of April. TG4 for CGW is planned for end of January 1999. The size of the execution project (start of design to TG4) is 45-50 kmh and there are 8 design teams and 1 test team. The teams are spread over four different locations.

Planned Usage Scenario:

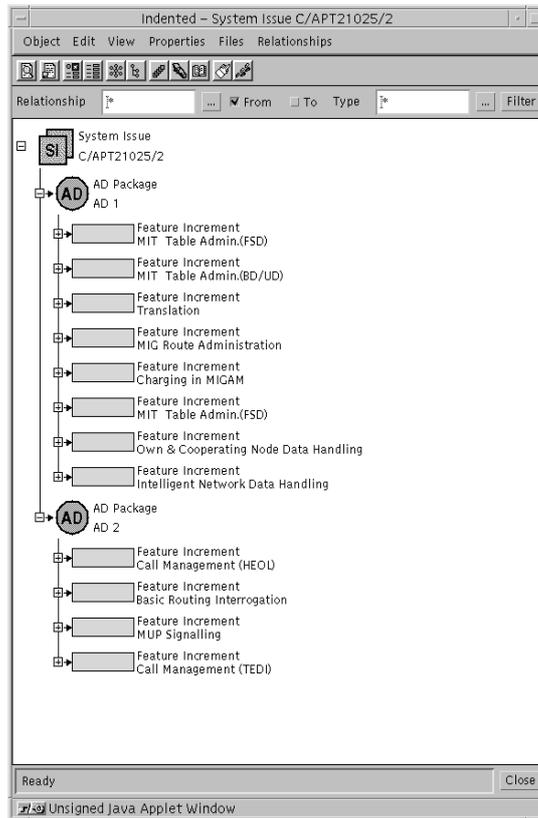
1. All increments and AD-packages are entered. This was updated a few times due to changes in the scope of the project.
2. The design base is imported from PRIM. Minor changes are updated manually, if there are any major changes the design base will be imported from PRIM again.
3. Cost of implementation data from the IPs are entered manually by the project administrator, in this case a consultant from Q-labs. This information is updated by the project if needed.
4. Milestones for all increments are entered and milestone criteria defined for each milestone.
5. Now the teams can export all necessary information regarding their increment to MS project where they can do a time plan for the increment. The updated information is stored in the prototype.
6. As soon as the teams have stored there plans the Project Manager (and everyone else) can check the current status of the project and if necessary take actions to make sure that the project follows the overall time plan.
7. The prototype is used for producing milestone checking reports. The report is a list of all documents that should be checked for this milestone.

The IPs were entered manually by Q-labs since the prototype was not available during the feasibility study. If it had been, and all cost information data had been stored in the prototype in an early stage, it would have been possible to get a very early overview of the project.

The prototype also helped in tracking all conflicts between increments, when the same design item are working with more then one increment. The main adaptation was to create the possibility for the teams to export their increment to MS project. This was done since the teams are responsible for making their own detailed planning, and it was natural to store this information in the prototype to it would be available to everyone in the project.

Before the project agreed to be a pilot there was several meetings with Q-Labs to discuss the usage scenario and needed adaptations. The prototype was installed on a NT-server at the end of January with a few Windows based clients. SUN-based clients were installed later. During the project one person from Q-Labs is regularly visiting EPK to give support and making sure that the necessary adaptations was done. He also assisted in entering the cost of implementation data from the IPs and importing the design base from PRIM.

Tool samples: the following figures show a number of samples from the tool. The first one show the Logical Construction Plan (which features are planned in each AD) at EPK.:



Next, an example of a functional anatomy, which shows the dependencies between the functions in a system (in this case APZ ODEN):

Object	Status	Description
Function Power CPU -	Ready	
Function Clock MAI -	Ready	
Function Working State Control	Ready	controlled by MAU
Function Scanning of all RP's (RPB)	In Progress	
Function Start RP's Start APZ block's (RPB)	Not Started	
Function DMA Single Oden	Not Started	
Function Measurement	Not Started	
Function Program change	Not Started	
Function SAAPZ SW	Not Started	
Function CACLS	Not Started	
Function SRAM Tuning Oden	Not Started	
Function Function change method 1	Not Started	
Function SYATI	Not Started	
Function Handling of software fault	Not Started	
Function Switch EX SB to EX	Not Started	
Function Backup	Not Started	
Function Test System	Not Started	

Finally, an example which shows the tracability from customer through features down to single products and documents:

The screenshot shows the 'Matrix - (Default)' application window. The interface includes a menu bar (Table, Object, Edit, View, Properties, Files, Relationships), a toolbar with various icons, and a 'Relationship' filter section with 'From' and 'To' checkboxes and a 'Type' dropdown. The main area is divided into two panes: a tree view on the left and a table on the right.

Tree View:

- Customer Nippon Telecom Inc
 - Customer Feature MAP -
 - Customer Feature STAT PA0
 - Customer Feature TRAF PA0
 - Set of Requirements 297/1056-FCP 103 280 C
 - Feature Increment TRAF -
 - Implementation Proposal 159 41 297/159 41-11/FCP 1
 - CNT 254 1022 R4
 - CAA 107 9468 R4A
 - Source Program Information 190 55 190 55-C
 - Signal Survey 155 14 155 14-CAA 107 9468
 - Data Change Information 109 26 3/109 26-C
 - Source Parameter List 190 73 190 73-CAA 1
 - Product Revision Information 109 21 109 21-CNT
 - Description 1551 1551-CNT 254 1022 D
 - Application Information 155 18 2/155 18-CNT 254
 - Structure Specification 131 61 131 61-CNT 254 1
 - CNT 254 1041 R6
 - CNT 254 1027 R6

Table View:

Object	Name	Resp	Description
			TRAFFIC REGULATION, PREVENTION OF OVERLO
			Traffic Regulation, Prevention of Blindload in the MSC
	MMMLR	EPL/X/M	
	MMMLRU	EPL/X/M	
			SOURCE PROGRAM INFO.
			SIGNAL SURVEY
			MMMLRCHANGE3
			SOURCE PARAMETERLIST
			PRODUCT REV. INFO
			DESCRIPTION
			MMMLR
			STRUCTURE SPEC.
	MTA	EPL/X/M	
	MNSS	EPL/X/M	

The taskbar at the bottom shows the Start button, the 'matrix' application icon, and the 'Exploring - Hdisk (C:)' window. The system clock shows 2:21 PM.

8. Summary

Incremental development is now becoming a standard way to develop projects within Ericsson, and we have experiences from projects ranging in size from small to very large (e.g. 2 million man-hours spread over 20 sites). Since incremental development is a very flexible concept we still see a lot of variation in how projects choose increments, but the ID method package gives a common reference. We have collected the first set of serious adaptations of the package, which has provided valuable feedback to the second version of the package which is planned for 1998. The tool support for managing incremental development projects are gradually maturing, and we now have several pilots running. The general experience with incremental development at Ericsson is very positive, and most projects have experiences several of the benefits mentioned in section 3.