FINAL REPORT

ROADS

REUSE ORIENTED APPROACH FOR DOMAIN BASED SOFTWARE

WITH THE FINANCIAL SUPPORT OF THE EUROPEAN COMMISSION – DGIII F4
ESSI #**21649**

Pascal MAHEUT
Jorge VIEIRA DA SILVA (MTA)

THOMSON-CSF Technologies & Méthodes

30 May 1998
Version 3.2

Table of Contents

1. Executive Summary

This document is addressed to the **software engineering community**. It proposes a view on the **benefits and impact on business development** that can be expected through the **implementation of state of the art software reuse within a product-line strategy**. It is based on experiments of the ROADS project conducted in the Thomson-CSF group, and benefits from the support from the European Software Institute. (The work and its dissemination is partially supported by the European Commission)

Using available best practice in reuse, ROADS implemented and developped **software reuse within a business development strategy.** This took place through experiments initiated from corporate management initiative and involving four Thomson-CSF Business units.

Two complementary business objectives, that were translated in the corresponding aspects of the software reuse implementation, drove the project and were conducted in 5 successive and iterative increments:

- at Corporate level **: to increase the average software reuse rate** and **to decrease the average cost of the reused lines of code** through definition and improvement of standard software reuse engineering practices based on a Product Line approach ,
- at Business Units level : to meet more specific business objectives (**cost reduction , time-to-market reduction , improvement of reliability**) through this Product Line approach ,

This was conducted through a product-line-based strategy and the analysis of commonalties and variations between specifications and products, leading to the evaluation of reuse potential benefits and constraints.

ROADS is a total success in reaching these objectives: the consistent approach in software engineering practice is helping Thomson-CSF keeping software development environment at the leading edge of technology and set up a continuous measurable process improvement based on the SEI reference:

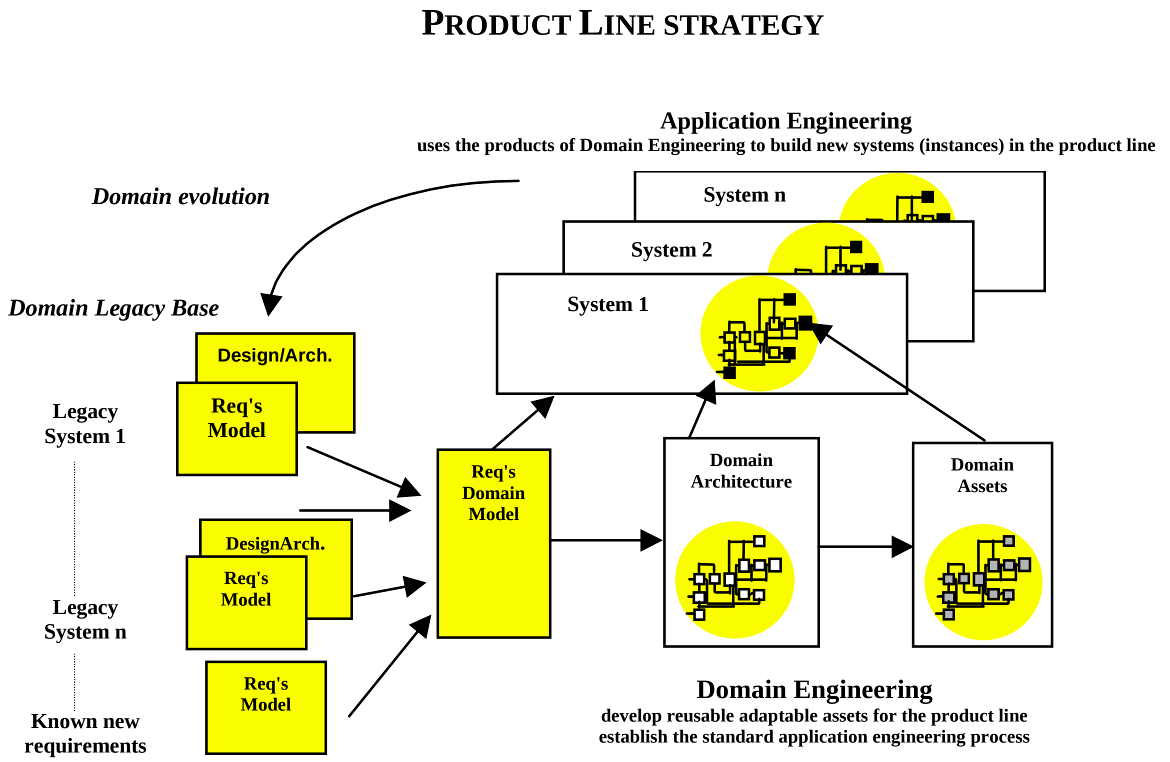
- The total rate of reuse increased from an initial 20% to more than 60%, exceeding the objective of ROADS initially set to 50%
- A global environment assessing costs and economic relevance has been designed and validated. It showed that for a reuse rate of 50%, the cost of reused software vs. new is in the range of 30%. This decreases to ca. 10% for a reuse rate >90%.
- Business objectives specific to each Thomson-CSF division where met : cost reduction by a factor 2,5 , time-to-market reduced by more than 50%
- Spontaneous interest from other divisions grew, based upon the visible success and ROADS assets, and led to an implementation of ROADS approach in a series of other Business Units.

These results have been presented to a series of conferences and workshops related to software reuse, leading to high interest from a variety of global companies.

Access to reuse strategic issues and technology is available, in complement to Thomson-CSF contacts, from the independent reuse experts at the European Software Institute in Bilbao (Spain).

2. Background Information

ROADS is a Process Improvement Experiment. As such, it does not intend to develop new reuse technologies, but rather to implement state of the art models and technologies within a practical environment. The strategy of THOMSON CSF in implementing ROADS was initially to exploit best practice of reuse within its business development, based on a Product Line strategy:



2.1 Objectives

The ROADS Process Improvement Experiment objective is to evaluate a product-line based reuse strategy on four different application domains, in conjunction with the current practices, with a view to follow the experiment with implementation of this reuse strategy within all the other domains of Thomson-CSF.

There are two levels for the experiment:

- **at the Corporate level:** to tailor and apply generic reuse models, methodology, processes and tools within Thomson-CSF to produce a corporate process model for Reuse
- **at Thomson Business Unit (TBU) level:** four mini-experiments to apply this Thomson corporate process model according to local business needs .Each mini-experiments is based upon a relevant baseline project.

Consequently there are two levels for the objectives of ROADS:

-at Corporate level:

- to produce a **validated corporate process model for reuse**
- to **increase the average reuse rate** of all TBU (from 20% at the beginning of ROADS to 50% targeted)
- to **decrease the average cost** of the reused line of code (at least 10%)

-at Thomson-CSF Business Unit level, more specific objectives were set, customised to the unit's requirements

- Airsys/ATM : **time to market reduction** for Air Traffic Control Development (from 6/12 to 4/8 Months)
- Airsys/WSD : **improvement of reliability** of large software (reduction of the nb of defects/KLOC)
- TT&S : **cost reduction** using reusable components (30% targeted)
- SYSECA/SIT : **improvement of flexibility and robustness** of software
- all : **to improve their Reuse Process**.

2.2 Involved companies and their roles

ROADS (Reuse Oriented Approach for Domain based Software) is an EC funded project in the context of its Software Best Practice initiative in which **Thomson-CSF Group** is evaluating the pilot adoption of an existing reuse methodology with the assistance of the **European Software Institute (ESI)** and **Prosperity Heights Software (PHS)**. The project is centred around the controlled realisation of four experiments to introduce reuse processes at an industrial scale.
The partners are:

- **Thomson-CSF** SA , Prime User, through Thomson-CSF Technologies & Méthodes (TTM), practically co-ordinating the ROADS project and its implementation in the 4 Business Units, which are users in the project :
    - **Airsys/Weapon Software Development** (Airsys/WSD)
    - **Airsys/Multi-Domain Control Centers** (Airsys/ATM)
    - **Thomson Training & Simulation** (TT&S)
    - **Syseca Industries et Tertiaire** (Syseca/SIT): Associated Partner , Users
- **Prosperity Height Software** (PHS) , Transtar : Subcontractors , Technology Suppliers
- **European Software Institute** (ESI) : Subcontractor , dissemination. The ESI consultant involved, Sergio Bandinelli, was tightly involved in these experiments since the beginning of the project. The results of these experiments are continuously taken into account in the ESI reuse work products in the context of their ESI ISO-SPICE framework.

2.3 Starting scenario

2.3.1 General business and cultural environment

All the information characterising the current status of software engineering practices is based on a synthesis established at corporate level, on the basis of the Capability Maturity Model (CMM) of the Software Engineering Institute (SEI).
Thomson-CSF corporate strategy introduced, at Business Units (TBU) level, a common and consistent approach in software engineering practices. This is aimed at:

⇒ controlling software development productivity and product quality,
⇒ controlling software development costs by identifying and capitalising on common factors,
⇒ keeping Thomson-CSF software development environment at the leading edge of technology,
⇒ setting up a continuous measurable process improvement based on the SEI reference.

At the corporate level, Thomson-CSF Technologies & Méthodes (TTM), a Centre of Expertise, responsible for defining and acquiring or developing the tools, methods and guidelines of the Common Software CASE Environment (Atelier Thomson de Génie Logiciel, ATGL), is in charge to help Thomson-CSF Business Units to reach these objectives. TTM also leads the corporate software process improvement programme, and is supported by the Thomson-CSF training centre (CAMPUS Thomson) which provides courses in software engineering practices and methodologies chosen by Thomson-CSF.
The company main concern in Thomson-CSF in the past five years has focus on the software process in order to improve its maturity and capability. An improvement programme has been initialised in 1992 using the CMM (Capability Maturity Model) model of the SEI (Software Engineering Institute).
Although the SEI CMM-V1 has many good features, it does not explicitly address reuse-centred aspects.
Thus a reuse process definition was identified as a critical issue for Thomson-CSF. In 1994, a working group on reuse was set-up in the frame of the SPICE (Software Process Improvement and Capability Evaluation) programme.
An assessment of the current reuse practices was performed and the first outcome of the initial assessment carried out by the reuse working group was that more than the third of the many millions lines of code delivered to Thomson customers was reused in an opportunist fashion, with an average reuse rate of approximately 20% but that this figure could raise at least a 50% rate if the reuse process was formal, organised, planned and controlled through measures. A pragmatic bottom-up white-box approach was used on the projects.
The development of a new product was traditionally made by picking up and modifying existing modules from older products with comparable functionalities. The disadvantages were important. Using this method, the overall cost was higher after three or four reuses because of problems in configuration management, change control, documentation evolution and extensive re-testing.
It was therefore decided to implement a consolidated methodology coherent with the corporate strategy and applicable in an efficient manner to the various domains and to their specific constraints, in particular be applicable to projects ranging from 3 to 36 months involving resources up to several 100’s of Manxmonths.

2.3.2 Technical environment

Two existing and complementary reuse methodologies were initially planned to be used in the experiments. The main approach is the Software Productivity Consortium (SPC) methodology, called RSP (Reuse-driven Software Processes), which is primarily focused in supporting product-line based reuse. The second one was the REBOOT approach, developed as part of a ESPRIT project. REBOOT takes a component library approach, focusing on re-engineering for reuse of existing legacy code. However, it complements RSP providing metrication instruments for software components and processes based. We have encountered some problems with Reboot because Transtar the company who sell REBOOT has disappeared in 1997. So we have modified our approch of REBOOT , we have received a training on REBOOT to have a better knowledge of the concepts supported by this tool and what we can expect of this kind of tool. But during the experiment we have used an other tool , RDD-100 from Ascent Lofgic Corporation to support the Product Line approach.
The **Reuse Economics Spreadsheet Model tool** from SPC (available on PC) was used during the Definition of the Strategic plan.

2.3.3 Training environment

Each increment start-up session on SPC processes corresponds both to a progress meeting and to a training session, during which contributors dialogue and learn from Thomson's consultants.
A training for Reboot Environment has been performed in 1997. The audience for this training was mainly the Domain developers.

2.4 Workplan

The ROADS project has been conducted in 6 main phases , each phase corresponding to a work package :

1. **WP1 - Coordination and reporting :** to ensure management of the project and proper reporting to EC
2. **WP2 - Definition** : to detail the definition of the actions to be done for the technology transfer , assessment and dissemination and particularly for the experiments of each business unit
3. **WP3 - Technology transfer :** transfer of the reuse technologies from the Software Productivity Consortium and reuse tools to Thomson-CSF and to tailor the reuse processes of the SPC to the in house needs
4. **WP4 - Experiments :** to apply these technologies on the four experiments : Domain and Reuse Capability Assessment, Reuse Economics Spreadsheet Model, 5 increments of the Reuse-driven Software Process , reuse tool.
5. **WP5 - Assessment , measurements and validation of reuse process :** to measure , analyse and validate the experiments
6. **WP6 - Dissemination action and transfer to the ESI :** to transfer the results to the European Software Institute (ESI) which will package , improve and disseminate the results of the experiments and ensure evolution of the reuse process.

2.5 Expected outcomes

Three sets of expected outcomes were defined for the ROADS Process Improvement Experiments:

2.5.1 Corporate level

At corporate level of Thomson-CSF expected outcomes of ROADS are:

- the availability of a validated corporate process model for reuse
- reaching an average reuse rate of all TBU (50% targeted)
- decreasing the average cost of the reused line of code

2.5.2 Thomson Business Units expected outcomes

Each TBU targeted a different level of improvement. This level of improvement was determined, for each TBU, during the experiments using the SPC models (RCM and DAM). The specific TBU expectancies were:

2.5.2.1 AIRSYS Weapon Software Development (WSD)

By experimenting the RSP, WSD has the objective to **increase the rate of reuse components in the delivered software towards increased reliability**. The second objective is the cost and time to market reduction.

One of the main point in AIRSYS WSD quality policy is the client's satisfaction that implies to deliver software in time with a minimum of remaining problems. WSD has developed a policy for reuse since 1990 based on the reuse of components. It appeared that reliability of several time reused components has increased. But the bottom-up reuse has shown its limits. The Short Range Air Defence System domain was chosen because a new development started with the goal to develop a line of products in one of the major domains of AIRSYS activities with hard competition background.

2.5.2.2 AIRSYS (ATC)

The experiment deals with a general goal, which is to **reduce the time-to-market** in delivering software for small Air Traffic Control Centres of the so-called EUROCAT 200 range of products **from the current 6 to 12 months down to 4 to 8 months**.

Success in world-wide business on such ATC systems is, in a large scale, driven by competitiveness and time-to-market reduction, in a kind of " Commercial Off The Shelf " approach. These two main criteria induce constraints on Software methods in our Unit :

- trend to a maximum Reuse of pre-existing software, as a good way to minimise costs and schedules,
- trend to quicken and ease integration phases, during which the end-user variabilities are added to our software baseline and capitalised for further potential use on the following contracts.

2.5.2.3 SYSECA (SIT)

Its main objective is **enhancing the flexibility of software production and robustness of the delivered software**.

The SYSECA Experiment in the ROADS project is focused on the ATS Software (Automatic Train Supervision). The ATS is a facility located in an OCC (Operational Control Centre) of a public transport network, supervising the traffic of the trains.

The experiment consist in :

- Experiment of RAP (Reuse Adoption Process) on ATS
- Experiment of RSP (Reuse driven Software process) on ATS

2.5.2.4 Thomson Training and Simulation (TT&S)

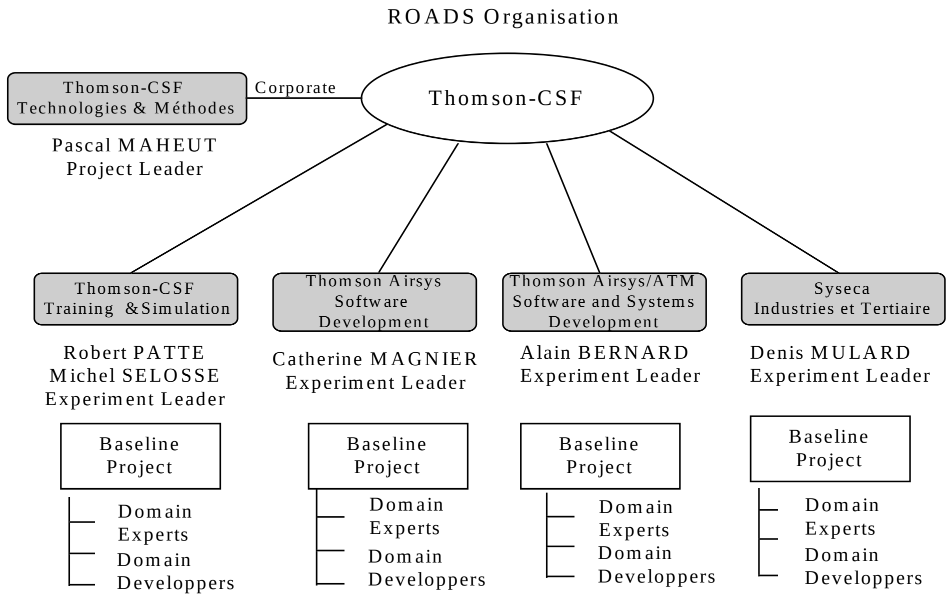
The main objective of TTS is to obtain a **reduction in costs through a better operation of reuse process**.

Reuse experience is not new for TT&S. Reuse is fully implemented in the civil aircraft simulator domain for ten years. However, in tank trainer domain, reuse is poor and very opportunistic, due to the diversity of the products and the very concept of product line. The experiment takes place in a market with an interesting growth, although associated with a decrease of the prices.

3. Work Performed

3.1 Organisation

The internal Thomson-CSF structure involved in ROADS is:



3.2 Technical Environment

We have used some spreadsheet templates and forms used during the Domain Assessment and the Reuse Capability Assessment .These spreadsheets have been tailored from the SPC spreadsheets and forms.

We have used the **Reuse Economics Spreadsheet Model** from SPC (tool available on PC) to compute the Unit cost of new code and unit cost of reused code (hours/KLOC).

We have used the SPC technologies described in three guidebooks and one training support :

1-Reuse Adoption Guidebook

2-Reuse-driven Software Processes Guidebook

3-Reuse Economics Spreadsheet Model User Manual

4-Reuse-driven Processes for Productivity and Quality

The first one describes a typical reuse adoption process (including the Domain Assessment and the Reuse Capability assessment).This guidebook appeared sufficiently mature to be exploited directly without assistance from SPC.

The second one and the fourth one describe Reuse-driven Software Processes (RSP) corresponding to a Domain or Product Line approach for software development. The corresponding guidebook is not sufficiently mature, it is the reason why we need assistance, consulting and mentoring from the RSP expert of PHS (see Paragraph 3.4).

The third one describes a reuse economics model to aid the user in analysing the costs and related economic parameters of interest in connection with the development of new application software systems composed, in part, of reused code.

Further information on these models are vailable in Annex.

We have used RDD-100 , a system engineering tool from Ascent Logic Corporation to manage the Domain requirements, and more specifically the commonalities and variabilities on these requirements .

3.3 Training

Formal training to SPC methods and process took place in 96 : the 6 people constituting the ROADS steering committee attended a pre-training in March 96 , a more extensive session for theoretical training in July for almost 20 people (Roads steering committee + Domain Experts + some domain developers).

The supporting material provided during this training is : Reuse-driven Processes for Productivity and Quality July 8 - 9 , 1996 - Prosperity Heights Software - Grady H. Campbell, Jr. - Rodney Bell

Milestones for training

Short Description of Subject of Training	Dates
Pre-Training on SPC processes (5 trainees , 2 days per trainees)	*MAR 96*
Theoretical training on SPC processes (20 trainees , 2 days per trainee)	*JUL 96*
Training on Reboot (10 trainees , 2 days per trainees)	*DEC 97*
Training on RDD-100 (1 trainee , 4 days per trainee)	*OCT 97*

Besides, each increment start-up session on SPC processes is both a progress meeting and a training session , during which contributors dialogue and learn from Thomson's consultants.

A training for Reboot Environment took place in 1997. The target for this training was mainly the Domain developers .

3.4 Role of Consultants

Two people outside THOMSON employees provided a very significant contribution to the project.

- **Grady CAMPBELL** , from Prosperity Heights Software , was previously a Principal Member of the Technical Staff at the Software Productivity Consortium .He is architect of the **Reuse-driven Software Processes** Guidebook .He designed the Spectrum environment and the TRF2 metaprogramming tool .His experience in software engineering encompasses information systems , embedded mission-critical systems , knowledge-based systems , and software engineering technology. Grady performs most of the training sessions , provides relevant documents , helps us to manage the increments start-up sessions in TBU plants , and check and corrects our progress to keep the team on the right track during the increment session s and between the increment sessions.
- **Sergio BANDINELLI** , from **European Software Institute** , attends some of our meetings, most of our internal assessment using SPC model , and all the increment start-up sessions and all the theoretical training. He gives a more "external" opinion on our progresses, providing advice on the weaknesses he feels in our experiment to help our improvement. He writes this opinion in increment start-up session reports sent to each ROADS business unit experiment leader. These documents have been highly appreciated by the ROADS participants. Sergio is also responsible of ROADS external dissemination actions.

Both are considered very helpful and constructive contributors by the teams of the four Business Units as well as by the co-ordinator TTM. They were recognised as experts and appreciated, on more relational criteria, for their availability and open-mind. However, for some individuals, technical dialogues and specific SPC terminology in English language has been a difficult gap during the first meetings.

3.5 Phase of Experiment

Workpackage WP1 : Co-ordination and reporting

A ROADS Steering Committee was composed by the ROADS Project Leader and the Experiment Leaders of each Business Unit .This steering committee have monthly meetings. Among the objectives of these meetings :
-to do a status of in progress actions both at corporate and mini-experiment level
-to define the schedule of the next short-term milestones : training , increment start-session , dissemination actions
-to collect and package the PPRs , MTRs and the other deliverables of the five business units involved in ROADS
-to inform the steering committee about ROADS events : participation to workshops or any other dissemination actions , internal or external (See § 3.6)
The minutes of these meetings are written by the project leader , approved by the steering committee , and sent to others dedicated people of Thomson-CSF for information. All these meetings minutes (more than 20) are available on the Thomson-CSF intranet « Thom'Web » accessible from all the Thomson-CSF Businees Unit in France and outside.

Workpackage WP 2 : Definition

The corresponding deliverable is the Experiment Document Definition . This document is included in the internal final report of each business unit (5 internal final reports).

The most important part of this workpackage has been to define the implementation of the four mini-experiments based upon a relevant baseline project to apply, validate and update the same corporate reuse process and to tailor it to their local business needs.During the project , there was a contnuous feed-back between the results of the four mini-experiments and the definition of the corporate reuse process.
The basic structure applied to each experiment is described in figure 1.

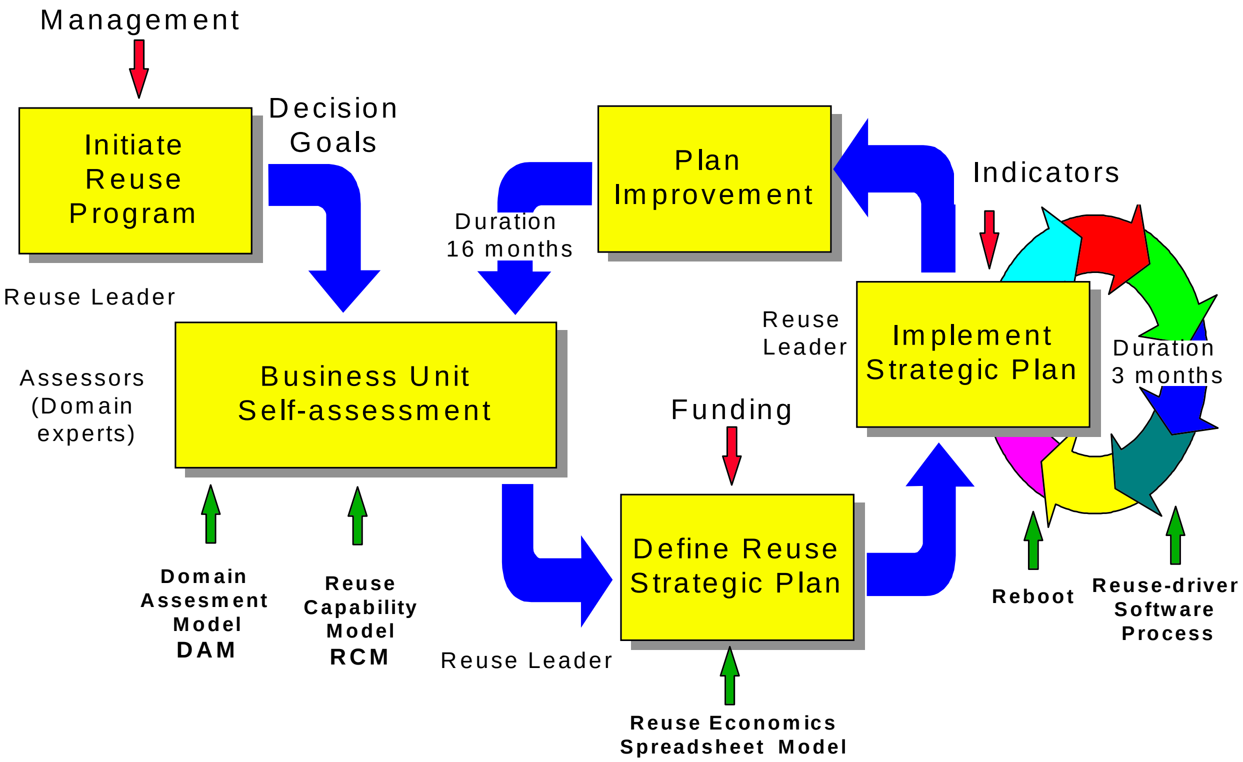


Figure 1 : Structure of each experiment

This process is a spiral or incremental process. Typically each main increment (black arrows) has a duration of 16 months. Globally :

- "Initiate Reuse Programme" box correspond to a part of the workpackage WP2 and WP3
- "Business Unit Self-assessment" box correspond to the activities Domain Assessment and Reuse Capability Assessment of WP41 , WP42 , WP43 and WP44.

- "Define Reuse Strategic Plan" box corresponds to activity Economical Assessment and to few sub-activities of Domain Engineering (Domain Planning for instance) in WP41,42,43,44
- "Implement Strategic Plan" box corresponds mainly to the activities of Domain Engineering in WP41,42,43,44
- "Plan Improvement" box corresponds to a part of WP5.

During this phase we have also defined and/or prepared the actions to be done for the technology transfer , assessment and dissemination :

-Technology transfer
We have choosen the consultant acting as SPC model expert. We have organised training session to « Reuse-driven Software Processess » , Reboot tool and RDD-100 (See § 3.3).
The motivation of the management of each business unit has been done , through information meetings , both by each experiment leader in his business unit and by the Roads project leader in all the business units.
We have defined the assessment packages and the tools used during the experiment , as indicated in §3.2 : Domain Assessment Model , Reuse Capability Model , RDD-100.

-Assessment
We have defined the the metrics, the corresponding evaluation method and specific measurable targets for each baseline project according to their business objectives.
Ex : One mini-experiment deals with a goal, which is to **reduce the time-to-market** in delivering software for small Air Traffic Control Centres of the so-called EUROCAT 200 range of products **from the current 6 to 12 months down to 4 to 8 months** , which was the measurable target.
Another one deals with the cost reduction , the measurable target was the percentage of cost reduction .

-Reuse-driven software processes : **RSP is an incremental process** , each increment has a typical duration of 3 or 4 months .We have planned to do 4 or 5 increments . Increment 0 and 1 are done , increment 2 is in progress increment 3 is scheduled in June , 4 in October and 5 , if needed , in December. ROADS focuses on Domain Engineering .The Domain Engineering process may be summarised as shown in Annex.

-Dissemination : The dissemination plan included both internal and external dissemination actions.

<u>Workpackage WP3 : Technology transfer</u>

The technologies transfer from the Software Productivity Consortium (Reuse Adoption Process, Domain Assessment Model, Reuse Capability Model, Reuse-driven Software Processes) has been made as expected during the ROADS time frame, with the support of external consultants and of the ESI.The technology transfer is done through the SPC and PHS documentation , the training sessions ,the RSP increment sessions , the monthly technical meetings and of course each experiment.

The technology transfer is detailed in the Technology Transfer Report (Internal - French) , included in the internal Mid-term reports of the business units.

-The technologies indicated in §3.2 have been tailored to internal use:

-Reuse Economics Spreadsheet Model **:** we have used only some equations of this model
-Reuse Adoption Process **:** the Reuse Adoption Process of SPC has been tailored as indicated in figure above
-Reuse Capability and Domain Assessments : french translation of the assessment worksheets used during the assessments , definition of the « ideal » team of assessors , definition of a Thomson-CSF template for the findings
-Reuse-driven Software Processes : these processes have been tailored during each domain experiment (Box "Implement Strategic Plan" in Figure 1).

Workpackages WP4 and WP5 : Experiments , Assessment , measurements and validation of reuse process
Each mini-experiments has been done according to the incremental process described in § 3.6 defined above.
To obtain the management commitment , the motivation of the management of each business unit has been done , through information meetings , both by each experiment leader in his business unit and by the Roads project leader in all the business units.

Actions done by the business units :

-reuse self-assessments (Reuse Capability Assessment and Domain Assessment) + corresponding findings
-definition of strategic plan
-ROI computation
-5 Reuse-driven software processes increments , according to the Domain Engineering Process

Actions done at Corporate level :

Two guidebooks :

- « Software Reuse through Product Line » adoption guidebook: answers to the questions Why , When , Where , and How launch or improve a Software Reuse Action Plan , based on the ROADS experiment synthesis. Including all the lessons learned during the reuse assessment in the business units and the definition of the reuse strategic plan
- « Software Reuse through Product Line » process guidebook : description of the software development process (including domain engineering and application engineering) used in the product line approach.This guidebook is tailored for Thomson-CSF and include or refer to the work products developped during the 5 RSP increment.

Comments

After performing the **assessments** each unit have prepared a set of slides to present the findings. In addition to specific discussions regarding the capability of each of the units, the presentations of findings lead to some discussions regarding the performance of the assessments.
A reuse capability assessment will be repeated in each organisation unit and the results compared to the baseline set by the initial assessment. This is a simple way of measuring progress in the reuse adoption process.

The reuse capability assessment were carried out using the Reuse Capability Model. Further informations on Reuse capability Model and assessment procedure are available in Annex.The reuse capability assessment results are included in the Experiment Results Report and Results Evaluation Report a part of the Internal Final Report of each Business Units.

The results of the Reuse Capability Assessment are used as a support for reuse adoption planning, since it helps in identifying the priorities for reuse adoption and to define his **Reuse Strategic Plan.** These plans cover product, business model, developing environment, organisational and process and methods concerns. According to the business unit the strategic plan has been applied totally or partially.

A business unit has identified the **ROI** of the reuse strategic plan (See §4 Results and Analysis).

The 5 Reuse-driven software processes increments , according to the Domain Engineering Process included :

-the development of work-products specific to RSP : domain definition , decision model , etc. Several RSP work products have been tailored to be compliant with the template of Thomson-CSF existing work products.
Example : the Product Requirements defined in RSP is based on the template of a System/Segment Specification (SSS) , the Product Architecture is based on the template of a System/Segment Design Description , etc.
-the definition of the **Application Engineering Process** : the process used by the Application Engineering to reuse the adaptable reusable components
-the **Product Family Engineering** : development of reusable components adaptable to the variabilities identified in the decision model , these components may be specification , design , code , documentation , test case, software architecture. Examples of reusable components developed by the business units : generic System/Segment Specification , generic System/Sub-system Design Description , domain Software Requirement Specification , etc.

All these components are now used in the development of the applications in the dedicated domains of each business units

We have used RDD-100 , a system engineering tool from Ascent Logic Corporation to manage the Domain requirements, and more specifically the commonalities and variabilities on these requirements . For the dedicated Domains we have entered in the database of this tool the domain requirements (including commonalities and variabilities) identified during the RSP increments. From these informations and with the help of RDD-100 we can produce **automatically:**

- a Domain Specification (Functional Architecture of the Domain)
- a Domain Design Description (Organic Architecture of the Domain)

with the traceability between domain requirements and the corresponding components.

- any System/Segment Specification (SSS) and System/Segment Design Document (SSDD) specific to an Application whose the rquirements are a subset of the domain requirements (all the commonalities + a part of the variabilities).

Workpackage WP6 : Dissemination actions and transfer to ESI (TTM and ESI)
Sergio BANDINELLI , our reuse consultant from ESI , attends some of our meetings , most of our internal assessment using SPC model , and all the increment start-up sessions and all the theoretical trainings .He received all the complementary supporting material and results about Reuse assessment don in the business unit.
He has written several documents :

-ROADS Increment 1 - ESI- 1997- ROADS02 - January 1997 (Internal Dissemination)
ROADS Increment 3 - ESI- 1997- ROADS02 - (Internal Dissemination)
-ROADS Progress Report - January 1997 (External Dissemination) : review and approval in progress

These documents and a copy of the presentations done by Thomson-CSF at ERW'97 are available on the ESI server , on the ROADS dedicated pages : http://www.esi.es/Projects/reuse/roads.
Other dissemination actions are detailed in §3.6.

Main Deliverables (excluding progress reports)

Deliverable reference	Description of the deliverable (Title)	Availability I - R - P
DWP21	Experiment definition document	R
DWP32	Technology transfer report	I
DWP42	Mid term report	P
DWP43	Experiment results report	I
DWP52	Results evaluation report	I
DWP6	Dissemination report	P
DWP11	F inal report	P

3.6 Internal dissemination

At corporate level , many internal ressources are used to disseminate inside Thomson-CSF group:

- the mission of Thomson-CSF Technologies & Methodes is to be a vector for technology transfer and institutionnalisation of new practices within Thomson-CSF
- the CAMPUS Thomson-CSF , the corporate training centre for managers and engineers
- the software engineering techniques school (ETGL : Ecole des Techniques de Génie Logiciel).The purpose of this school is to train new software engineers for Thomson-CSF needs.This school is opened to other industrial companies in various activities sectors (avionics , car manufacturers , etc)
- the SPIN (Software Process Improvement Network) of Thomson-CSF , the network of all SEPG members
- publications within the Thomson-CSF , TT&S , TTM and Syseca (technical newsletter)

Internal

- presentations and workshops within Thomson-CSF CAMPUS : "Journées Systèmes d'Information et de Commandement" (Juin 96) , "Journée Technique Syseca" (Sep. 96) , "Ecole SIC" (Juin 97) , Forum des CET (October 97 , 2000 people)
- presentation within the SPIN (Software Improvement Network) of Thomson-CSF : (Dec. 96 , April 97:one day dedicated to Software Reuse)
- presentations to Corporate Committee : Software Engineering Steering Committee(2/year) , SPICE Thomson-CSF and Reuse Technical Committee (2/year)
- training : "Cycle de formation Responsables de Lots Logiciels" (2hours/session , 2 or 3 sessions/year), "Journées SPICE - Ingénierie des Produits et Systèmes Logiciels" 2hours/session , 2 or 3 sessions/year
- guidebooks
  - « Software Reuse through Product Line » adoption guidebook: answers to the questions Why , When , Where , and How launch or improve a Software Reuse Action Plan , based on the ROADS experiment synthesis. Including all the lessons learned during the reuse assessment and the definition of the reuse strategic plan
  - « Software Reuse through Product Line » process guidebook : description of the software development process (domain and application engineering) used in the product line approach.This guidebook is tailored for Thomson-CSF and include or refer to the work products developped during the 5 RSP increments.
  - the Product Line Approach is token into account in two others Thomson-CSF guidebooks: "Description des cycles de vie" , "Description des Processus Elementaires"
- intra-web: Thomson-CSF Intranet ("Thom'Web) is accessible from all the T-CSF Business Unit in France and outside.There are several pages dedicated to Th-CSF Reuse Initiative and to ROADS.

During and after ROADS project end, all 4 Business Units involved already expanded usage of ROADS:

- WSD is expanding ROADS methodology towards all activities in software development
- TTS is expanding ROADS methodology from military simulator applications towards civilian simulators, as well as energy application simulators
- SIT is expanding ROADS methodology from ATS to SCADA
- ATM is expanding ROADS methodology from small ATC to large ATCs, including work done under international cooperation with Germany and UK
- Other business units are being informed and trained on ROADS and are implementing the corresponding methods in order to support their product line strategies. ROADS is extensively reported and advertised in the internal website.

External : Channel for benefiting from ROADS

- Dissemination through another ESSI project N°23960, Dissemination Action SURPRISE (SURvey on Possibilities of Reuse In Software Engineering)
- ESSI workshop 12/13 December 1996 Brussels
- European Reuse Workshop ERW'97 26/27 November 1997 (4 Th-CSF presentations)

4. Results and Analysis

The main achievements are detailed below:

Thomson-CSF Business Unit	**TT&S**	**Airsys/WSD**	**Airsys/ATM**	**Syseca/SIT**
Specificities	Driver training simulators	Short Range Air Defence System and SW test bench GUI	Small Air Traffic Control Centres	Automatic Train Supervision and SCADA
Common Objectives	**to produce a validated corporate process model for reuse** **to increase the average reuse rate of all TBU (50% targeted)**			
Specific Objectives	**Cost reduction** **Competitiveness**	**Increase the rate of reuse for increased reliability** **Product line implementation**	**Reduce time to market (from 6/12 Mo to 4/8 Mo)** **Reduce costs**	**Enhance flexibility and robustness** **Reduce costs**
Main quantitative achievements[1]	**49%** of the LOC are produced within Thomson CSF as **reused** code **21%** of the LOC are produced within Thomson CSF through **automatic generation**[2] 30% of the LOC are produced within Thomson CSF from scratch This reveals a strong impulse for reuse implementation from a corporate level to the Business Units level[3](supported by **two Corporate guidebooks** for Reuse Adoption and Reuse Process)			
	ROI on 3rd project Cost reduction for one system by a factor 2,5	Product line implementation. Few applications for quantitative measurements	Time to market reduced by more than 50% Productivity multiplied by 4 in 2 years	Quantitative achievements not yet measurable
CMM level	ISO 9000, Raised to level 3 (level 4 soon)	ISO 9000, Raised to level 3	ISO 9000, Raised to level 2 (level 3 soon)	Raised to level 2
Success and failure factors (common)	1. Technical Top management commitment is required ROADS was a strong incentive for the implementation of reuse Product line approach adequation is proved Maturity of the units (CMM) improved 2. Training Well defined domains helped structure training Volume of activity is an issue to implement experiments 3. Role of consultants Availability of expert domains is key for reuse implementation in a product line strategy			

[1] As per an internal Thomson CSF Benchmark report from December 1997

[2] An important part of the automatic generation corresponds to an automation of the reuse process

[3] An estimation of the level of reuse above 60% at the end of 97 is realistic, vs. some 20% at the start of ROADS

4.1 Technical

Generally speaking, the technical impact of ROADS implementation led to:

- globally the average level of each reuse capability profile decreases when the level of the model increases. The business units assess themselves better in the lower level of the reuse capability model.
- reuse capability profile differs between the business unit. There is no "standard" profile.
- the business unit with a high level of maturity in software development (CMM) are often more severe than the other about their own software reuse capability level
- the main axis of improvement identified during the self-assessment are always strongly related to the business unit. The more frequent improvement axis identified are "Commonalties and Variabilities Identification", "Costing and Pricing Strategy", "Training", "Organisation commitment" , "Reuse Strategy"
- the level of reuse capability is globally consistent with the level of software maturity (CMM). From a macroscopic analysis we have found that Opportunistic level of RCM corresponds to "Repeatable" level of CMM , "Integrated" to "Defined" and so on.

A summary of the methodological issues for each Business Unit is detailed in the following table:

Methodological issues				
Thomson-CSF Business Unit	**TT&S**	**Airsys/WSD**	**Airsys/ATM**	**Syseca/SIT**
RCM assessment results	Target integrated process. Priorities defined. Concentrate on lower level goals first. Progressed rapidly to CMM level 3	Highlighted the limits of components reuse. Clarified priorities. Checked management support for integrated process	Potentiality to target leveraged process.	Showed problems of integrating the usage of reusable components in the process. High risks in resource commitment.
Domain definition	Get synopsis written/reviewed by domain experts. Extend glossary and, if necessary, provide mappings to terms used by customers. Provide higher-level description of comm. and var.	Good start for the synopsis. The glossary is very brief. Commonalties need not be too specific, but only a means to elicit variabilities.	Synopsis should describe purpose of functions rather than how they are performed. Also the glossary should take an external view of the system. Var. at too low level.	The document that describes the domain should be shaped, to be a synopsis avoiding superfluous information. Glossary contains mainly acronyms and is to be extended.
Decision model	Done by looking at former projects. Decisions are too specific and may need to be reformulated in user level terms.	Marketing perspective needed to define the set of values that decisions may take. Complexity may be reduced by binding some of the decisions to given values.	Current version is a good starting point. The DM could behave as a reference to establish the baseline for a new application dev. It is difficult to anticipate user needs (no direct relationship with clients).	The decision model should be linked to variabilities and not be developed independently. Some correction needs to be made and the dependencies among decision must be considered.
Product family engineering	Some experiments done with requirements document. The cost-effectiveness of taking a formal approach is doubtful, thus it is preferred to take a light approach (e.g., provide some guidelines in prose).	Examples done with requirements and design documents. Interest declared also for tests and project plans. A change in the structure of the documents may be necessary to favour reuse. The approach using instructions (instead of conf. commands) is preferred.	Application of product family engineering to all workproducts looks very promising. Examples done with requirements document. Different degrees of automation identified for the derivation of a document from its family.	Some examples were explained, but not developed yet for domain specific documents.
Process engineering	The unit was assessed at CMM level 3. The CMM processes and culture impose some constraints on how appl. dev. processes can be changed.	For the moment, domain eng. will be carried out as part of appl. eng. A coming project will be used as a starting for domain activities, conceived as an extension to the project.	No radical changes are needed in appl. eng., since the idea of domain is already existent. The idea is to make the appl. dev. more disciplined by formalising and making explicit the domain concept.	Not specifically addressed.

4.2 Business

ROADS has been implementing a Reuse Economics Model, allowing a quantitative and qualitative assessment of the impact.

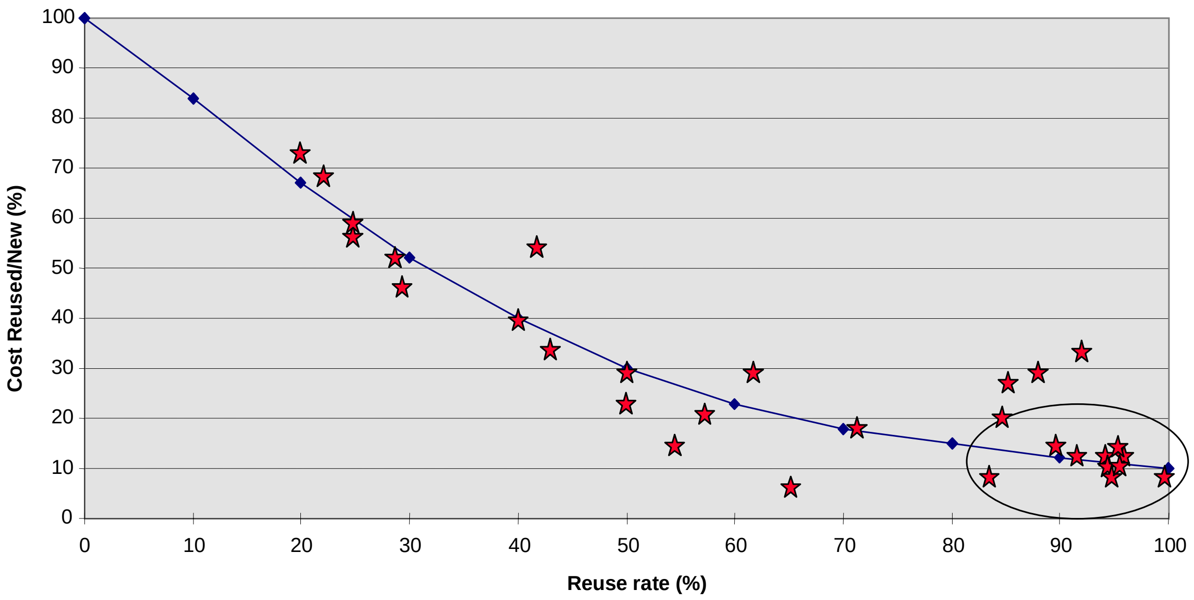
The main input data of the economic model are :

- Proportion of Code Reuse R — $0 \le R \le 1$
- Unit cost of New Code (man-hours) — Cvn — $0 < Cvn$
- Unit Cost of Reused Code (man-hours) — Cvr — $0 < Cvr$
- Average Size of Application System (KLOC) SS — $0 < SS$
- Application Engineering Cost per System (man-hours) — $Ca=(Cvn-(Cvn-Cvr).R).SS$

- Relative Reuse Cost Cvrr = Cvr/Cvn

The two main indicators of the software reuse are R and Cvr . A reuse action plan may be considered successful if there are a R increasing and a Cvr decreasing concurrently. So the organisation must establish a data collection and metrics programs including at least R and Cvr to verify and to prove the success of the reuse action plan.
R , Ca , SS are generally known and available in the business units , but it is not the case for Cvn , Cvr and Cvrr . On the assumption that Cvr and Cvn be constant for the applications of the same Domain and that Ca , R and SS be known for , at least , two applications of the same Domain , we have try to calculate Cvn and Cvr . Often we cannot obtain consistent results because Cvn and Cvr are not constant or because the data coming from different applications are not consistent , but sometimes we have obtained consistent value of Cvr and Cvn .
The first level findings are : 0.1 < Cvrr < 0.8 and Cvrr decrease when R increase
The results of some 30 cases analysed within the Thomson-CSF group (both inside and outside the ROADS work) lead to the following relationship between Cvrr and R:

Reuse relative cost evolution



This curve seems rather stable, and can be taken as a general basis of reference. The circled examples, which show a good reuse rate, relate to businesses that have a long history in product-line orientations, sometimes since several years.
For the cases situated above the curve, the efficiency of the reuse process might be questioned. An improvement of the processes is to be implemented, in order to reach the curve level, or even go below.
The quantification of the improvement can be evaluated, together with the potential gain derived from an improvement plan.

4.3 Organisation

It is a main goal of ROADS to enable the replication of these experiments in other organisations, such as other Thomson-CSF business units, other ESI members and the European software industry in general. The four experiments defined in ROADS project provide a very valuable field-test for the validation of the approach, since they are not “invented” projects with the only scope of demonstrating a technology but they are real projects with the benefits and constraints that real activities impose. In addition, the four experiments provide the opportunity to try the same approach in four different contexts, making it possible to make meaningful comparisons and learn from the commonalties and differences among the experiment performances.
The approach being followed at Thomson-CSF, has been used in a limited number of industrial experiments so far (including Rockwell, Boeing and Lockheed-Martin). ESI contributes to this maturation process by capitalising on these experiences and making them available as an asset for the planning of future improvement actions.
The main observations based on ROADS results are:

4.3.1 An incremental iterative approach is essential to foster understanding of the reuse approach.

This incremental approach will serve also to reduce risks associated with reuse, since it will allow to have partial benefits (support to on-going projects) from the work done so far.

4.3.2 One of the firsts tangible results from reuse introduction is improvement of communication.

The domain engineering process defined in RSP suggests to start with a domain definition. This involves writing a brief domain synopsis and a domain glossary. The domain synopsis should briefly describe the domain in non technical words, but significant effort is required by domain experts to avoid references to the specific solution provided by the systems they build. It is also surprising that people that work side by side everyday on a specific domain have substantial disagreements on the meaning of basic definitions to be included in the domain glossary. At the end of this process, the common shared understanding of the domain is significantly increased.

4.3.3 Reuse involves changes in the way requirement engineering is performed.

The engineering process by which systems are developed can be seen as a decision taking process. These decisions are either derived from customer requirements or from an engineering trade-off. RSP encourages making all these decisions explicit in a decision model. After some reflection, some of the decisions that initially seemed independent, turned out to impose constraints on each other or even be one derived from the other. This “simplification” of the decisions smoothes the way for requirements engineering, which needs only to deal with a reduced set of customer-dependent decisions (engineering trade-offs should actually be irrelevant for the customer).

The results from ROADS contribute to the dissemination of the experimented good practice. It is much easier to convince other units of the group with tangible results stemming out from experience, rather than with written guidelines and procedures. Eight evaluations of reuse potential have been performed in 1997, on domains not covered by ROADS and/or with units not involved in ROADS. A majority of these evaluations is already followed by implementation of concrete actions. Other evaluations are already programmed all through 1998.

This reuse model is implemented in the "Reuse Guideline" document, that tackles the adoption of reuse with domain evaluation and capacity of reuse based on the SPC model, but also the more general software engineering practice within a product-line approach. Other software engineering guides are currently being produced, focusing of various aspects of the engineering practice, but integrating the reuse dimension from the ROADS experience. This particularly the case for the "Description of Life Cycles" and "Fine-tuning elementary processes". All these guides are part of the Software Development Reference of Thomson-CSF and widely used in the group. All these documents are referenced in the "Catalogue of the SPICE/reuse assets" of Thomson-CSF, and available on the internal web site.

4.4 Culture

It is demonstrated in ROADS that the rewards of reuse implementation can be detected within the first process increment time frame. However, full results are obtained within a product life cycle, and the strategy must have a long term view, over a period correspond to several times the average time-to-market. (although one of the objectives of the reuse implementation is to specifically reduce this time to market)

The reuse software process has shown an important contribution to an improved domain assessment capability, leading to more efficient business approaches beyond the improvement of the software process.

Furthermore, the work done in domain assessment has a catalytic effect on bringing together the various experts at domain, software, user environment and business levels, bringing a better view on users requirements and avoiding excessive technical drive.

At this stage, ROADS is not building a comparative evaluation of the various businesses and activities of the different units. However, it proves efficient to assess the efficiency of each unit, and its evolution.

Generally the Domain experts are really interested by the Reuse Product Line Approach (even if they have not always sufficient availability to implement it), because this approach is based on their domain expertise , at least during the Domain Analysis , it is a kind of recognition of their status of expert. It is not the same thing for other software process improvement (CMM for instance).

Of course the software reuse process improvement is easier to accept (but not necessarily easier to implement) by organisation having any previous experience of software improvement process (like CMM for instance).

4.5 Skills

Today the acquired skills essentially concern the reuse assessment methods. Now we have a reuse assessment guideline with examples, findings annotated outline, useful presentation of results, assessment team ideal composition. Many people having participated to an assessment , as assessor, are able to be facilitator for new assessment. So people can see now how to use the results of the assessment for their reuse improvement plan. The most important asset is probably the taking into account of the commonalties/variabilities concept which is the base of the product line approach.

ROADS is evidently replicable within the following environment:

- Large organisation, with a multiplicity of operations in disconnected domains, although smaller companies with high software activity are also within the target
- Having a level of maturity already CMM 2 or higher
- With availability at corporate level to introduce a cross-domain common reuse strategy as part of a business development strategy

5. Key Lessons

We quote below an analysis of lessons learnt from ROADS in *The complementary aspects of process capability and reuse capability*, by Sergio Bandinelli and Álvaro Sanz (ESI, 1997):

"The incremental nature of the adoption process makes it possible to start obtaining benefits very early in the reuse adoption process. Even if these benefits have not been measured (largely because they are very difficult to be measured), it is fundamental report them to keep the process moving and on the right track.
The first benefits that have been reported include:

- New opportunities for improvement have been identified, including possibilities for automating the generation of documents, to decrease the number of necessary validation tests, etc.
- The organisation is better aware of the range of applications it is capable of building by capitalising of past project experience.
- Some of the domain engineering workproducts are started to be used as an additional support for negotiating and setting new contracts or to decide whether to bid for a contract or not…

…It is possible to derive some lessons learned from the experience.

- Reuse adoption requires some level of process maturity. There are a few basic processes, such as project management and configuration management that need to be in place before the organisation embarks into a reuse programme.
- Established processes are much difficult to change. To some extent, reuse adoption is slowed down by previous achievements in defining and deploying standard processes. There is significant resistance to change an already defined way of working, unless you have very convincing reasons to change.
- Difficulties and resistance are also encountered when the reuse adoption programme follows other quality improvement actions (such as obtaining ISO 9000, achieving a certain CMM level [SEI95], etc.). Practitioners often perceive this as yet another perturbation of their work.

These observations lead us to conclude that reuse adoption must be conceived as an integrated part of a comprehensive process improvement programme. Within such a programme, a synchronisation between growing process capability and in adopting reuse should exist.
The RCM model does not distinguish between process capability goals and reuse specific ones. In our view, a clearer model is achieved if process capability issues are separated from reuse specific ones."

5.1 Technological point of view

Reuse of previously developed software has for long been an essential part of software engineering practice. Although systematic implementation of reuse methods and technologies is still limited to a reduced number of software organisations, the obtained results are often spectacular in terms of engineering efficiency and effectiveness.
The traditional approach to reuse is based on “component engineering”. The basic assumption here is that reusable components, which can be of different nature (e.g., code, design components, etc.), may be identified in the systems under development. These components are then classified and stored in a reusable component repository, making them available to developers for being reused.
A larger definition of reuse relates to commercial repositories of components, available to a variety of companies. This document and the analysis of ROADS impact and replicability is restricted to internal reuse of components that are the property of the company, which is at this stage, a major preoccupation of software intensive organisations.
Beyond the technical issues, the integration of a software engineering reuse strategy within a business management approach appears to be appropriately exploiting the technical benefits of reuse in a global business perspective. Obviously, this requires a strong management drive and the proactive involvement of all sectors of the company: technical, commercial, financial, etc..
One of the most interesting of such management approaches consists in a "product-line engineering" approach. A product-line is a collection of (existing and potential) products that addresses a coherent business area or domain. Product-line engineering is concerned with the efficient development of a product-line that delivers high quality products tailored to the specific needs of each customer within a variable range of commonalties and variabilities from one instance to the other.
Reuse in a product-line approach takes full benefit of the component as well as the architecture reuse.

5.2 Business point of view

Obviously, the concept of software reuse is associated to software intensive activities. However, it is important to stress that reuse can be affective in a variety of company and product environments, in particular in company size, line of code count, software engineering teams size, domain of applications, etc.

5.2.1 Reuse and company size

It is now a general experience that **reuse applicability does not depend on the size of the company**, nor on the size of the software engineering teams. Teams with as few as 3 people, but also larger teams of several tens of people, have been identified among the most efficient in reuse.

The interest for reuse comes essentially from the analysis of the commonalties and the variabilities between products to be proposed to the customers. This requires a good knowledge and experience both at product and at domain levels.It is generally observed that the implementation of reuse relates more to a management drive and to a flexibility to adapt engineering processes than to the size of the organisation.

5.2.2 Reuse, product size and product life

Reuse is applicable within software intensive companies where applications of a significant size (typically 50 KLOC and up) are expected to be implemented in several instantiations within a product-line strategy.

It is important to stress that starting with a progressive approach, within iterative increments as suggested in the ROADS experiments, allows the control of the applicability of reuse and of its potential return on investment. It also allows to raise the maturity of the company (CMM level).

The product life time (and the number of applications), together with the product size, define the critical mass allowing the efficient implementation of reuse.

5.2.3 Reuse and application domain

Reuse is applicable to software systems serving virtually all domains of application. However, the most interesting applications for component reuse are related to domain where stable user environments (and corresponding system architectures) exist.

The high level of maturity (as per the Reuse Capability Model of SPC) necessary to the implementation of a complete reuse strategy induces a strong capacity to address significant variabilities in the applications.

5.3 Strengths and weaknesses of the experiment

It is clear from the ROADS work as well as from input from other reuse experiments that software reuse is not to be compared to other alternative solutions: **reuse is an irreversible trend in software engineering practice within a maturing software industry**.

As such software reuse is to be implemented within software intensive organisations, whenever sufficient commonalties can be found in a line of products, in complement to other engineering methods and according to the business objectives set at corporate level.The **management involvement** in the implementation of reuse within the business practice of the company is of paramount importance.

6. Conclusions and Future Actions

6.1 General conclusions

ROADS is a total success in reaching the objectives initially set: the consistent approach in software engineering practice is helping Thomson-CSF keeping software development environment at the leading edge of technology and set up a continuous measurable process improvement based on the SEI reference:

- The total rate of reuse increased from an initial 20% to more than 60%, exceeding the objective of ROADS initially set to 50%
- A global environment assessing costs and economic relevance has been designed and validated. It showed that for a reuse rate of 50%, the cost of reused software vs. new is in the range of 30%. This decreases to ca. 10% for a reuse rate >90%.
- Technical objectives specific to each Thomson-CSF division where met
- General CMM maturity level of each division raised significantly
- Spontaneous interest from other divisions grew, based upon the visible success

These results have been presented to a series of conferences and workshops related to software reuse, leading to high interest from a variety of global companies.

Furthermore, a high number of THOMSON CSF Business Units, both those having participating to ROADS and those having not, started or reinforced Reuse Actions based on the Product Line approach.

Those plans have often been incorporated in more general Process Improvement actions. Three months after the end of ROADS, more than 10 of those actions have been initialised.

On another hand, the Corporate level of THOMSON CSF, using the central role of Thomson Technologies et Méthodes (TTM), continues to explore and to enlarge the Product Line approach. This is implemented through:

- **Software Engineering**: Management and tracability of common requirements and variabilities of the Domain, definition of functional and organic Domain architecture.

  In the frame of the **ESSI PIE#23983 POSE**, THOMSON CSF evaluates the opportunity of including a Process Area « Product Line Adoption » to a System Engineering « staged model » based on the SEI SE-CMM. This Process Area is based on ROADS achievements, extended to the System level.

- **Software Workshop**: Management of configuration and of technical features « Product Line », metrics tools specific to reuse (e.g. reuse rate of a software component code, documentation, specification, etc.), archiving, classification and extraction tools for reusable components, tools for the automatic generation of components and applications.
- **Object Oriented Methods**: dedicated to the Product Line approach.
- **Software methodologies**: definition of the « Software Reuse by Product Line » Life Cycle.
- **Continuous improvement of Software Engineering Process in general**: impact of reuse on the elementary processes (specification, design, coding, test, etc.) and their tailoring.

6.2 Internal implementation of ROADS at THOMSON CSF

Internal implementation of ROADS at THOMSON CSF started already during the ROADS project final period (1997) and is expanding. Early 1998, all units involved in ROADS were already using the methodology for other activities than those initially experimented. New units are being informed and trained to the ROADS methods in a continuous way.

This consolidates the methodological role of THOMSON Technologies et Méthodes (TTM) within the group.

6.3 ESI and software reuse

Access to reuse strategic issues and technology is available, in complement to Thomson-CSF contacts, from the independent reuse experts at the European Software Institute in Bilbao (Spain).

The European Software Institute (ESI) action has been of paramount importance in the implementation of ROADS and in the exploitation of its achievements.

The Institute is currently considering reuse a main strategic axis of strategic importance. This leads to a comprehensive surveying activities of the global environment in reuse, and to direct support to European projects and companies wishing to get involved in this area of software engineering.

As a first introduction to the topic, ESI proposes on its web site relevant information for newcomers to reuse. For advanced and reuse knowledgeable individuals, a policy of partnership can be implemented.

6.4 How to get involved with ROADS methodology

As indicated before, the initial steps shall definitely be decided at general management level, and supported by a general business approach.

External support is normally of great help in these preliminary steps, and two levels of expertise are necessary:

Reuse expertise: Thomson-CSF is fully satisfied of having used the services of Grady Campbell in these approaches.

Domain expertise: this is complementary to the technical support, and links with the commercial and business orientations of the reuse product-line approach.

Experts at both levels might be identified with the help of the ESI.

6.5 Relevant contacts

Direct contacts with **Thomson-CSF** ROADS project manager can be established with **Pascal Maheut**:

Thomson-CSF TTM

pascal.maheut@ttm.thomson.fr

Tel: +33 1 69 33 01 38

Fax: +33 1 69 33 94 76

Sergio Bandinelli is the contact at the **ESI**

Sergio.Bandinelli@esi.es

Tel: +34 4 420 95 19

Fax: +34 4 420 94 20

7. Glossary

ATGL	Atelier Thomson-CSF de Génie Logiciel
CMM	Capability Maturity Model
CSF	Critical Success Factor
DAM	Domain Assessment Model
ESI	European Software Institute
ESSI	European Systems and Software Initiative
ISO-SPICE	International Standard Organisation -Software Process Improvement and Capability dEtermination
KPA	Key Process Area
MCC	Multi-Domain Control Centers
MTR	Mid-Term Report
PHS	Prosperity Heights Software
PPR	Periodic Progress Report
RAP	Reuse Adoption Process
RCM	Reuse Capability Model
ROADS	Reuse Oriented Approach for Domain-based Software
RSP	Reuse-driven Software Processes
SEPG	Software Engineering Process Group
SIT	Syseca Industries et Tertiaire
SPICE	Software Process Improvement and Capability Evaluation (Thomson-CSF specific)
SPIN	Software Process Improvement Network
SPC	Software Productivity Consortium
TBU	Thomson-CSF Business Unit
Th-CSF	THOMSON-CSF
TTM/DLS	THOMSON-CSF Technologies & Méthodes/Direction Logiciel & Systèmes
TT&S	Thomson Training & Simulation
WSD	Weapon Software Development

8. References

A complete list of references is available on the ESI web site: http://www.esi.es.

Main documents of relevance for the reuse implementation are:

1) Reuse Adoption Guidebook SPC-92051-CMC Version 02.00.05 November 1993

2) Reuse-driven Software Processes Guidebook SPC-92019-CMC Version 02.00.03 November 1993

3) Reuse Economics Spreadsheet Model User Manual SPC-91158-CMC Version 03.00.08 July 1993

4) Grady H. Campbell , Jr. - Rodney Bell , Reuse-driven Processes for Productivity and Quality - July 8 - 9 , 1996

5) James O'Connor , Software Productivity Consortium , Catherine Mansour and Jerri Turner-Harris , Rockwell International , Grady H. Campbell , Jr , Software Productivity Consortium , Reuse in Command-and-Control Systems, IEEE Software September 1994 , page 70 to 79

6) Software reuse introduction requires a process perspective ESI-1996-REUSE01 November 1996 by Sergio Bandinelli and Santiago Rementeria

7) Reference Architectures in a product line process context ESI-1996-REUSE02 November 1996 by Sergio Bandinelli

8) A Unifying Framework for Reuse Economic Models ESI-1996-REUSE03 November 1996 by Sergio Bandinelli and Goiuria Sagarduy.

9) The complementary aspects of process capability and reuse capability by Sergio Bandinelli and Álvaro Sanz

9. Annexes

Annex I - Roads detailed methodology

The business unit self-assessment was based on two models :
-**Domain Assessment Model** : the set of business environment and products factors that determine an organisation's opportunities for reusing technology
-**Reuse Capability Model** : the set of organisational and process factors that influence how effectively an organisation practices reuse.

Domain Assessment Model

Five main concerns are assessed during the Domain Assessment : Market potential, Existing domain assets, Commonalties and variabilities, Domain stability ,Standardisation in the domain
The results may be presented in a Kiviat diagram .

Concern	What to ask
Market Potential	In the given market , is there a healthy demand for products that could benefit from reuse?
Existing domain assets	Are assets and expertise available? Are assets of high quality? Can they adapt to market needs?
Commonalities and variabilities	Is there much similarity between needs and between products? Can differences be clearly identified and managed?
Domain stability	Are changes in technology and customer needs predictable and manageable?
Standardisation in the Domain	How far do standards in the domain go in controlling variation and in making the context for use of an asset predictable?

Reuse Capability Model

The RCM consists of a set of critical success factors (CSF) that correspond to issues most critical to improve reuse capability. These CSFs are organised in four groups: application development factors, asset development factors, management factors and process and technology factors. Each CSF is defined in terms of one or more goals that state what needs to be achieved in order to fulfil it.The Reuse Capability Model has four stages of implementing : Opportunistic , Integrated , Leveraged , Anticipating.

Stage	Key characteristics
Opportunistic	Projects individually develop a reuse strategy ; reuse activities are defined in the project plan Existing assets (small and large) are reused. Throughout the project life cycle , assets are identified for potential reuse on the basis of similarity between current developer needs. Reusable assets are under configuration control. Where applicable , existing tools and methods are applied to reuse activities.
Integrated	Reuse activities are defined and integrated into the organisation's standard processes. Assets are designed for reuse on the basis of current and anticipated developer needs.. Common architectures are developed for product families. Feedback on reusable assets is used to maintain or enhance them. Tools are tailored to support reuse.
Leveraged	A product-line reuse strategy is developed to maximise the benefits of reuse over sets of related products in terms of current customer needs. Assets are developed to allow the reuse of early life-cycle assets with automatic reuse of derived life-cycle assets. Process performance is measured and analysed. Tools supporting reuse are integrated with the organisation's software engineering environment.

Anticipating	New business opportunities that build on the organisation's reuse capability and reusable assets are created on the basis of anticipated customer needs and technology innovation. The effectiveness of reuse technologies is measured and used to determine which is or are the most effective. The organisation's reuse-driven process is flexible enough to adapt new process/product environments.

Reuse Capability Assessment Procedure

One reuse capability assessment was performed in each of the organisation units participating in ROADS. The assessment teams included persons belonging to the unit (self-assessment). The smallest team was composed by three individuals and the largest one by eight individuals .The probably most important issue is to select the good people in the assessment team , they should be known as domain experts in their respective job (software development , project management , business development , etc.) .Each assessor must representative in his job and the whole assessor team must be representative of the different job involved in the domain. All the assessments but one included the participation of a facilitator that managed the meeting.

Each assessment had a duration of one day. The agenda for each of the assessments included the following issues:

1. The facilitator provides a short introduction to precise the context and scope of the assessment (the processes to be assessed) and the assessment organisation (approximately 1/2 hour).
2. Each assessor independently evaluates the extent to which the organisation unit meets each of the goals identified in the RCM in a scale of 1 to 5 (1=not satisfied, 5=fully satisfied) (approximately 1 hour).
3. Each assessor independently evaluates the expected impact of each goal of RCM on the organisation unit reuse capability in a scale of 1 to 5 (1=no positive impact, 5=high positive impact) (approximately 1 hour).
4. The scores provided by the assessors are introduced in a spreadsheet-based tool to derive a graphical representation of the results (approximately 1/2 hour)
5. The scores are discussed, giving special attention to those goals in which there is discrepancy among the scores given by each of the assessors, with the objective of obtaining a consensus among the assessors (approximately 2 to 3 hours).

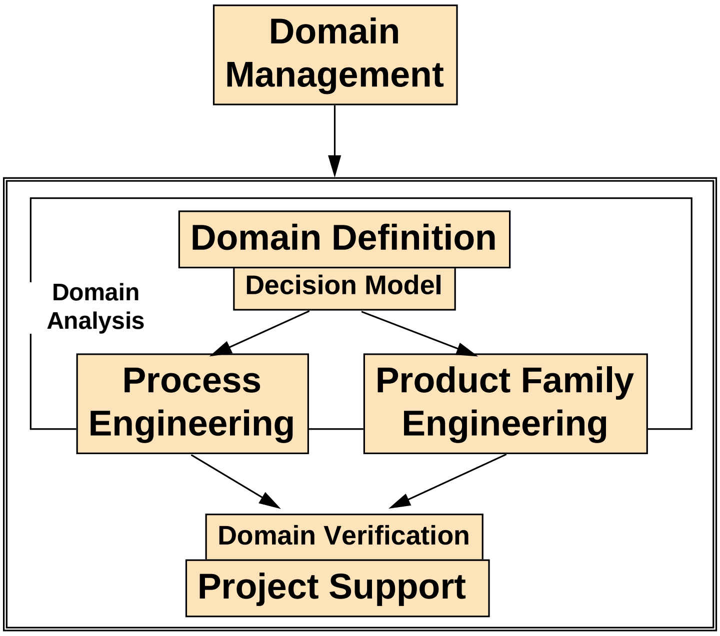
The final scores are updated in the tool to obtain the final graphical representation of the assessment results (profile).

Reuse-driven Software Processes(RSP)

The scope of RSP is a product line business enterprise , it tries to address the needs not only of individuals projects within that enterprise but also of the enterprise as a whole.It consists of 2 activities : Domain Engineering and Application Engineering .

Application engineering is a standardised approach to creating a product for a customer .Its objective is for projects to accomplish the same results that they do today but more rapidly and at lower cost with more consistent quality.

Domain Engineering makes this possible for all the projects of a business .As such , the exact character of application engineering may differ depending on the degree to which domain engineering is able to standardise and streamline the work of projects.The Domain Engineering Processes may be summarised as below:



The main work products of Domain Engineering are :

- Domain Management : **Domain Plan** schedules , budgets , assignements and progress evaluations (box "Define Reuse Strategic Plan")
- Domain Definition :
    - **Domain Synopsis** : informal description of the Domain
    - **Domain Glossary** : definition of terminology of the Domain
    - **Commonalties and Variabilities** : list of characteristic of the domain that corresponds to a similarity (commonalties) or to features that distinguish (variabilities) among members of the associated family of systems
    - -**Decision Model** : abstract form (concepts , decisions , and structure) , based on variabilities , of all the engineering decision that should be solved by the Application Engineer
- Product Family Engineering :
    - **Product Requirements** : definition of the requirement of systems in a domain relative to a Decision Model
    - **Product design** :
        - **Product Architecture** : specification of the (adaptable) architecture of the products for a system (possibly as a set of components)
        - **Component Design** : definition of the design of a (adaptable) components identified in the Product Architecture
        - **Generation Design** : specification of the Generation Procedure (i.e. , the lmapping from a Decision Model and Product Architecture to work products for an application)
- Product Implementation :
    - **Component Implementation** : creation of the Adaptable Components as specified by a component design
    - **Generation Implementation** : creation of Generation Procedure
- Process Engineering :
    - **Process Requirements** : definition of an application engineering process and concrete form (syntax) of an associated decision model
    - **Process Support Development** : standards and procedures, in the form of documents and supporting automation , that institute a standard Application Engineering process , as specified by the Process Requirements.
    - -**Domain Verification** : evaluation of compliance of Product Implementation and Process Support Development with Domain Definition , Decision Model and Product Requirements and Designs.
- Project Support :
    - **Domain Validation :** evaluation of the quality and effectiveness of Process Support from the perspective of Application Engineering Needs
    - **Domain Delivery :** assistance of Application Engineering projects in the effective use of Process Support

Annex II - Summary of the ROADS individual experiments

Thomson-CSF - Airsys/ATM

Project goal is to apply guided software reuse methods in order to decrease costs and time to delivery for small Air Traffic Control systems belonging to a line of products called EUROCAT 200 and 1000.
The work done aims towards increasing the ratio of reused software (versus newly developed one), and easing, thus shortening, the system integration phase.
It mainly started with a Domain analysis, through identification of commonalties and variabilities coming from a sum-up of end-users requirements.
Pre-existing data help us in refining our internal process, to evolve towards the more formal SPC method and tools.
Main visible results in this beginning of 98 are the building-up of several documents such as :

- ⇒ a Decision Model and an Application Modelling Notation, which are already applied on new incoming projects, to check requirements consistency as compared to our product software baseline.
- ⇒ the Domain Assessment and Reuse Capability Evaluation, which give a more detailed view of the Domain, thus suggesting potential impact on our processes, organisation, internal communication paths, skills, and tools used for software configuration management.
- ⇒ the additional instruction to our ISO procedure PDL109 on Software Development, to have teams reuse the software components of the product library in a more industrial way
- ⇒ generic documents such as the SSS (system segment specification) and SDP (software development plan) to be easily instantiated on new incoming bids and projects

This work is performed on a background which is already defined, both at product line level and at process level (including our CMM level 2 evaluation, our ISO 9001 certification, and our goal to reach CMM level 3 in 98).
It is in tight correlation and complements the on-going works performed by the team dedicated to benchmarking and competitiveness measurements.
A new evolution in product line has been induced in October 97 by the set up of AIRSYS ATM, a Joint-Venture of THOMSON and SIEMENS on ATC activities.
Both companies having their own line of products for Small ATC Centres, the challenge for 98 is to merge them into a new product line called now ‘WATCHKEEPER’, without impacting on our customers requirements satisfaction, on our costs and time to delivery, and taking full benefit of the pre-existing assets in both product lines for a better cost effectiveness.
The competencies and support documents raised through ROADS experiment will be major assets for driving this new domain extension analysis.

Thomson-CSF - Airsys/WSD

The ROADS experiment was initially concerned with the short range air defense weapon systems, the sector of activity represented by the generation of systems NG-PH1 (New Generation Phase 1).

The initial objective of ROADS was to increase the amount of reuse in this domain, enabling the reduction of anomalies. The secondary objectives were to reduce the overall cost and the time-to-market.

Experimentation in this domain having suffered from a lack of availability of teams and from the market weakness, has been reoriented towards IHM for simulations as of September 1997. Within this framework, development in this new sector is under way and so enabling examples of software products within a product-line strategy : domain definition, definition of the decision models, SRS implementing the decisions on the application domain.

The auto-evaluations Domain Assessment Model (DAM) and the Reuse Capability Model (RCM) had been carried out in the short range weapon system domain.

The DAM allows a good overall view of the domain and was judged to be reliable for choosing a sector and to define the practical level of reuse to set up, according to the business and the business' objectives.

The RCM allows for a good insight into the practical technical parameters and allows the definition of an action plan corresponding to the practical levels needed.

The principal facts established from this experiment were :

⇒ The process of reuse oriented towards the product line is in the right direction, which pose little technical difficulties and which can be adapted to different business objectives and desired reuse methods.

⇒ The domain choice for business and for organisation and the definition of the objectives are very important in order to succeed the implementation of the product line.

The experiment led also to a CMM level raise to level 3, and benefited from an ISO 9000 certification.

The contributions of the experiment allow the dissemination at the centre of Reuse Leader's service and the management of information actions in the software division of WSD.

Thomson-CSF - TT&S

The objective of Thomson Training & Simulation (TT&S) is to reduce the cost of software development using reusable components.

The following actions have been undertaken during the experiment:

- Transfer of reuse technologies from SPC,
- Precise definition of the domain of the experiment,
- Domain and Reuse Capability assessments,
- Progress of 5 increments of the experiment, with :
    - For the Domain : Domain definition, Synopsis, Glossary and Decision model,
    - For the components: Decision model, specification, design, test documentation, applicative engineering process definition,
- Impact of object technologies in the product-line approach,
- Computation of the ROI (Return Of Investment) of the SPC process,
- Proposal for future actions

More particularly, the ROI computation was based on 3 alternative strategies for reuse within the terrestrial weapon sector:

⇒ The current process, where a new business is derived from the closest previous business

⇒ A process based on the development of domain related generic components

⇒ A product-line process, based on the reuse driven development process (RSP)

After a number of instances within the product line approach, the total cost increase is limited. The ROI is reached after the 3rd instance of the RSP process.

Thomson-CSF - SYSECA/SIT

The Syseca Experiment in the ROADS project is focused on the ATS Software (Automatic Train Supervision). The ATS is a facility located in an OCC (Operational Control Center) of a public transport network, supervising the traffic of the trains.

The set objectives were:
- Decrease the cost through:
  - covering the overall life cycle of a project
  - allowing better flexibility
  - increasing quality & robustness of reusable software components
  - Managing the diversity("variabilities")
- Standardising the basic architecture to ease Research and Development on new functionalities and/or requirements.
- Experiment RAP (Reuse Adoption Process) and of RSP (Reuse driven Software process) methodology on a first domain

After 24 months of intensive process reengineering, the main results achieved within ROADS include:
- Comprehensive approach of RAP & RSP : product, process, organisation, business, ...
- RAP & RSP are practical to organise and give consistency to scattered initiatives
- Assessments & incremental philosophy allow to focus primary on more profitable aspects
- Extend the RAP & RSP approach to a larger domain in the organization (i.e. an horizontal domain common to several market)

Other positive impact includes:

Working on Domain Glossary
- ⇒ work on database modelisation
- ⇒ TRANSMODEL (European prestandard)

Working on big variabilities
- ⇒ work on new programming techniques
- ⇒ Programic Logic Constraint