

Software Evolution En@bling

CC Reengineering

CC Open Source

CC Outsourcing

CC Consulting

CC Training

CC ANALYZER

CC AUDITOR

CC ARTISAN

CC ASSESS

CC OPAIRA

IT Software Technology
for the Future



"Software Evolution Enabling - this concept is programme. Existing IT solutions can thus be kept on track with the latest advances in technology."

Ajit Nambissan, Direktor CC Indien

CC Software Evolution En@bling, or **SEE** for short, is an integral software engineering approach comprised of methods, tools, architectures and project management techniques. **SEE** ensures the adaptation of existing software systems to modern IT technologies without sacrificing functionalities that have already been implemented. **SEE** not only integrates and incorporates the experience and business logic of your existing software solutions but also allows considerable improvement and expansion of the system's functionalities.

The key to success is to optimize these functionalities

- so they meet the needs of the business and technical requirements,
- they are within budget,
- and to ensure that the risk level is kept to an absolute minimum.

Our clients enjoy the best of today's technology while keeping a foot firmly planted in the IT future. **SEE** opens the door to the IT future - allowing our clients to decrease costs on one hand, while they increase savings on the other.

The squaring of the circle? Maybe - but there is a practical solution:

CC
Software Evolution En@bling®.

"The wheel of time turns faster in information technology than anywhere else - but this does not mean that we have to reinvent the wheel."

Mario Zaleski, President of the CC Group



The Challenge

Modern companies depend more and more on high-tech solutions to ensure optimal performance levels in all their business activities. We protect the investment and on-going development of existing IT systems so that information usage and reuse processes can be beneficially applied towards the achievement of your company and business objectives.

Today's IT technology is developing at an incredibly fast pace, and it is unrealistic for businesses to attempt to compete with this pace. On the other hand, these changes cannot be ignored as this will undoubtedly prove detrimental for business. The picture of an ever-changing moving target and difficult to manage IT landscape is beginning to emerge.

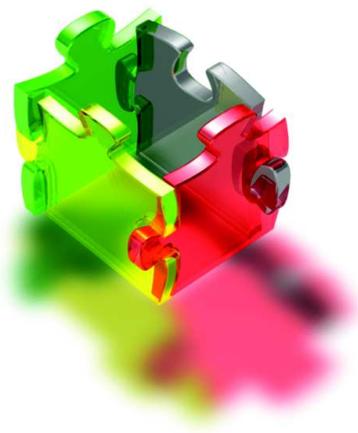
However, we are drawing an alternative picture one where you simply adapt your existing software systems by employing architectures and technologies specifically designed for this purpose. One of the challenges here is to carry out this adaptation process in a timely and cost-effective manner. The appealing feature of our Software Evolution Enabling process is the ability to combine the old with the new. This means retaining tried and trusted functionalities and yet facilitating the unavoidable technical evolution.

The goal of **SEE** is to modernize and optimize your IT operation - taking the best from the past and the present - and to strategically position it for the future. Your operation will evolve as a modern and future-style software environment is introduced while a sense of comfort and familiarity is maintained through your existing systems. The integration of the new and the consolidation of the old go hand in hand to provide you with modern effective tools that guarantee your present and future business competitiveness. Our customers expect us to provide leading-edge solutions in a timely and cost-effective manner, and ...

... we do.

Software Evolution En@bling provides lots of opportunities for cost savings and minimized risks. Current information technology challenges are tackled in a pragmatic, yet focused way. **SEE** opens the way for the IT future the whole technological IT environment is included in the **SEE** strategy, from mainframe back-end with core business processes to the individual client/server applications on the network. Below is a list of activities involved:

- ❑ Definition and implementation of new up-to-date frameworks
- ❑ Breakdown of monolithic IT solutions into platform-independent application components
- ❑ Removal of technical dependencies of the system-specific components
- ❑ Technical and logical rework of the data architecture
- ❑ Ensuring integrated communication between independent applications
- ❑ Creation of cross-system interfaces to existing business processes
- ❑ Integration of specialized applications into the overall architecture
- ❑ Integration of existing IT solutions with standard software products



The CC Process in Practice

Our vision gives existing IT solutions a new lease on life and make them "fit for business" on a technical level. Our **SEE** projects achieve this by addressing three main areas, while at the same time keeping the effort and risk to a minimum:

- Fundamental rework of the software architecture
- Consolidation of the data storages
- Uncoupling of the IT solutions from the technical system software

In order to achieve success in these three areas in a targeted and integrated manner, we define individual tasks and steps in the process. As such we differentiate between "infrastructure" tasks and conceptual "technical" tasks.

We initially focus on preparatory stages as a general development process for the actual on-site implementation (Table1).

Interrelated activities in for the general development process:

- Definition of goals based on technical environments
- Assessment of the existing software from a technical and economic perspective
- Identifying interlinks and dependencies between existing systems and their logical structures
- Creation of technical components for a new target architecture, based on industry standards
- Rework of existing systems

Table 1: General preparatory activities



The specifications for on-site infrastructure tasks depend mainly on our clients goals and requirements and thus they vary accordingly. We ensure the efficient and practical application of the defined steps by developing a project-specific methodology (Table 2). This methodology also allows us to be flexible in supporting and meeting our partners needs and requirements. While individual tasks are closely correlated, they can also be achieved in parallel and repetitive tasks automated.

In addition to the activities already mentioned, the underlying project methodology includes:

- Creation sets of tasks for all software components and determining the processing order
- Securing the co-existence of existing and future environments during the migration process
- Organizing and synchronizing the necessary parallel maintenance processes in a **SEE** project

Table 2: Project methodology

Depending on current implementation, a selection of the following technical tasks may be required:

- | | |
|---|--|
| <ul style="list-style-type: none"> □ Change programming languages that are not future-oriented □ Transfer into one or two standard programming languages □ Break online programs into interface- and business-logic modules □ Remove all system specific constructs from application modules □ Establish the data usage of all application programs □ Build reusable object-service modules with defined methods and uniform interfaces □ Create "views" which provide the necessary partial view of the data to the application programs and remove all system-specific dependencies □ Develop an optimized physical data model □ Eliminate unnecessary redundancies □ Migrate data from data access systems that are no longer actively supported □ Provision utilities and methods for data migration into the consolidated physical data-model | <ul style="list-style-type: none"> □ Integration of additional developments <ul style="list-style-type: none"> ❖ Integration with standard software systems ❖ Use of particular software components ❖ Rework of interfaces to external systems based on newly established standards □ Provision of a global development and test environment <ul style="list-style-type: none"> ❖ Facilitate the co-existence of standard and individual components ❖ Ensure decentralized configuration management ❖ Provision cross-platform test management □ Derivation of the logical data model based on the physical model and the actual data usage □ Develop an optimized logical data model as a basis for further development steps □ Adapt the existing physical model to the new logical model |
|---|--|

Table 3: Overview of possible "technical" tasks

This not only means that restrictions imposed by manual and mostly schematic activities are removed, but also that productivity and quality as a whole process is optimized. The underlying technical tasks typically comprise:

1. Consolidation of the deployed languages
2. Opening of online-systems and isolation at the presentation level
3. Encapsulation of the data access logic
4. Technical consolidation of data models
5. Integration of further developments and standard software
6. Consolidation of the logical data model

Detailed descriptions of possible technical tasks are listed in Table 3.

Through the use of automated testing processes we can guarantee the technical functionality. As a result of a **SEE** project, our clients software system is significantly upgraded from a qualitative, technical and functional perspective.

"Evolution Enabling basically boils down to exploring the present in order to master the future."

Ajit Nambissan

The Results at a Glance

A typical **SEE** project will lead to the following technical improvements:

- Significant reduction of the technical heterogeneity
- Minimization of the technical mix
- Introduction of future-orientated application architecture
- Integration with standard software systems (COTS, enterprise, application, etc.)

In addition to these technical improvements, our clients also have considerable economic advantages, including:

- Increased production stability and system efficiency
- More transparent standardization of interfaces to external inputs/outputs
- Higher flexibility
- A more structured development and maintenance environment, thus ensuring a low risk project.

Technical improvements in greater detail:

- Reduction of number of programming languages used
- Reduction of different data systems
- Strict division into presentation, application, and data components
- Encapsulation of all data and communication interfaces
- Implementation of standardized environments
- Elimination of proprietary development platforms and opening up for standardized environments
- Expanded systems for new communication channels (Internet, WAP, XML, SOAP, ...)

Table 4: Technical improvements

Experience has proven that benefits are measurable, as illustrated below:

- General improvements
 - ❖ Elimination of approximately 25 % of the total software code
 - ❖ 20 - 25 % of batch programs are no longer required
 - ❖ The technical quality of the IT solution is improved often considerably
- Production
 - ❖ Unnecessary data-transformations are removed
 - ❖ Transaction-orientated processing is possible
 - ❖ System administration becomes significantly easier
- Development
 - ❖ Reduction on critical system resources dependency
 - ❖ Lower maintenance upkeep, cost and fees
 - ❖ Fewer dependencies on "special developers". For example for languages like Assembler, non-mainstream database systems, special generators, etc.

Technical benefits in greater detail:

- Open systems
- IT operations and solutions possible on various platforms
- Component formation
- Integration of standard and individual components
- Future oriented software architecture
- Quality assurance and test management
- Divided configuration management
- Develop ongoing test and quality mindsets

Table 5: Technical benefits

CC Software Evolution En@bling

- IT Software Technology for the Future -

Conclusions

Software Evolution En@bling supports the visions of today's IT businesses. Visions which are put into practice on a step-by-step and controlled basis. These visions are performed by a highly motivated and capable team that produces fast and economical results.

SEE will overcome many technical deficiencies and in fact removes potential barriers and hurdles at an early stage in the process, leaving minimal and calculable risks.

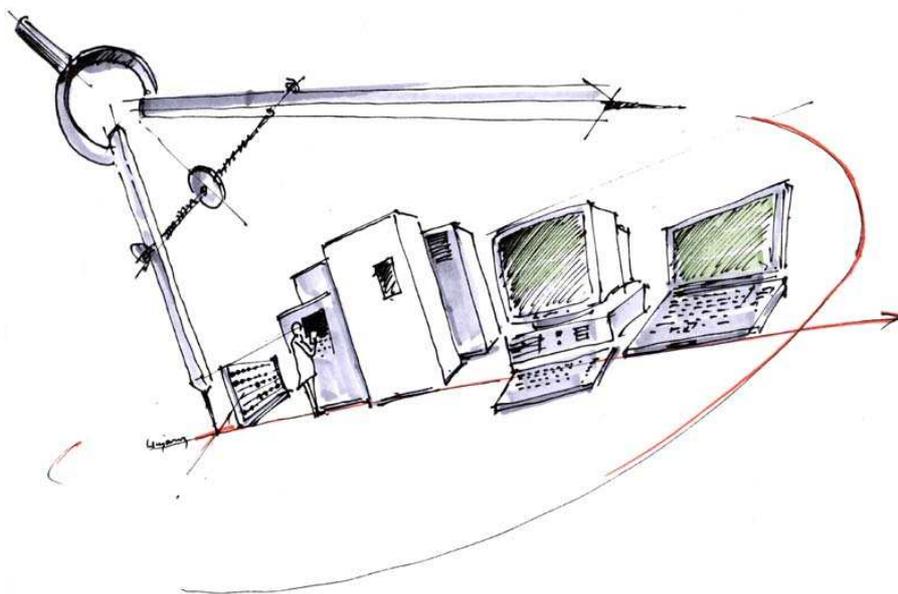
SEE enables your systems to evolve and your business to compete. Cost savings can range from 30 % to 70 %.

The process involves all developers and users, and has positive effects on present and future development methodologies in the areas of:

- Quality Assurance
- Test Automation
- Configuration Management

With **CC** you get performance that is on time, on target, and on budget. In fact, talk to us about fixed price projects.

That's what we do best.



"Our clients challenges are also our challenges only when our solutions fully satisfy the requirements of our clients are we really satisfied. This underlines our vision of customer relationships as partnerships and is an important prerequisite of software evolution."

Mario Zaleski



CC Germany

CC GmbH
Kreuzberger Ring 36
65205 Wiesbaden
Phone +49-611-942040
info-europe@caseconsult.com

CC India

Case Consult (India) Pvt. Ltd.
D1 Periyar, Technopark Campus
Trivandrum 695 581
Phone +91-471-2700176
info-india@caseconsult.com

CC USA

Case Consult Corporation
18 Lyman Street, Suite O
Westborough, MA 01581
Phone +1-508-651-9898
info-usa@caseconsult.com

www.cc-gmbh.de