CRUISE Implementation

Gary Sparks, Cray Research, Chippewa Falls, Wisconsin, USA

ABSTRACT: *Many companies are in the process of looking into the purchase and implementation of a "call/problem tracking" system to better manage their operations. Cray Research has recently implemented CRUISE (Cray Research Unified Information System Enterprise), which is based on a third-party call/problem tracking system. This paper will focus on the implementation issues Cray Research faced during the deployment of this information system.*

Cray Research purchased a third-party product from Quintus Corporation called CustomerQ, to be used as a basis for the Cray Research Unified Information System Enterprise (CRUISE). The functionality to be contained in CRUISE included:

Problem Tracking	(Motif and WWW interfaces)
Activity Tracking	(Motif and WWW interfaces)
Dispatch Tracking	(Motif and WWW interfaces)
Escalation Tracking	(Motif interface)
Configuration Tracking	(Motif interface)
Generation of Defects	(Motif interface)

This paper is intended to discuss the evaluation process and implementation of CRUISE.

The Selection

It is important that any evaluation include the following:

- Functionality--what are the key requirements (as detailed as possible) that the software must meet?
- Total cost of the software (including licenses)
- Cost of vendor support/service
- Cost of customization (vendor or local)
- Cost of database licenses if not included in the vendor's software pricing
- Training costs, both user and developer
- Evaluation tool to rate the various vendors against the above criteria

The following steps were used during Cray Research's evaluation process:

1. Assigned an overall project leader.
2. Determined high level management requirements (functional requirements).
3. Formed a technical evaluation team and expanded the functional requirements to include detailed technical requirements. It is advised that end users be included on this technical evaluation team.
4. Cray management and the technical team then prioritized the requirements.
5. Hired an outside consultant[1] to help identify a "short" list of potential vendors that might meet our technical/functional requirements.
6. Each potential vendor was invited to present a half-day technical demo to the technical evaluation team.
7. The technical evaluation team narrowed the vendor list down to two, based on what was seen in the demo.
8. The technical evaluation team then spent two months performing a full technical evaluation based on our requirements. This evaluation included:
    - Installed the vendor's product locally.
    - Spent two full days with each vendor, asking technical questions and learning more about their products.
    - Evaluation team was divided into various technical areas and asked to evaluate the vendors' products based on the technical requirements.

[1] Consultant name and address are available upon request.

- Checked vendors' references.
- Received cost quotes.

9.Once the steps in (8) were completed, the team compared the functionality demonstrated/evaluated against our technical requirements. (See chart 1 for an example of our comparison tool).

10.The group then selected one vendor.

11.We completed a make vs. buy analysis.

12.We made a decision to buy.

The Implementation

The Quintus product was purchased in December 1994. The following teams were formed to help with the deployment:

- Development team of ten (one project manager, seven developers, two writers) was formed in January-February 1995.
- CRUISE management team, made up of approximately eight key senior international managers from our user base. This team was used to make business decisions related to the deployment of CRUISE.
- CRUISE design team, made up of approximately eight key first-line user managers. This team focused on helping the CRUISE development team prioritize user feature requests.
- CRUISE consultants, a team of end-users that provide consulting to other end users.

CRUISE was released worldwide on December 4, 1995, and uses the following. (See chart 2 for the CRUISE distributed architecture.)

- Quintus CustomerQ toolkit (highly modified)
- Quintus batch facility (QWBatch)
- ORAPerl
- C
- UNIX shell scripts
- Pro*C
- Oracle

We learned a lot in this deployment. Following are some key points to remember:

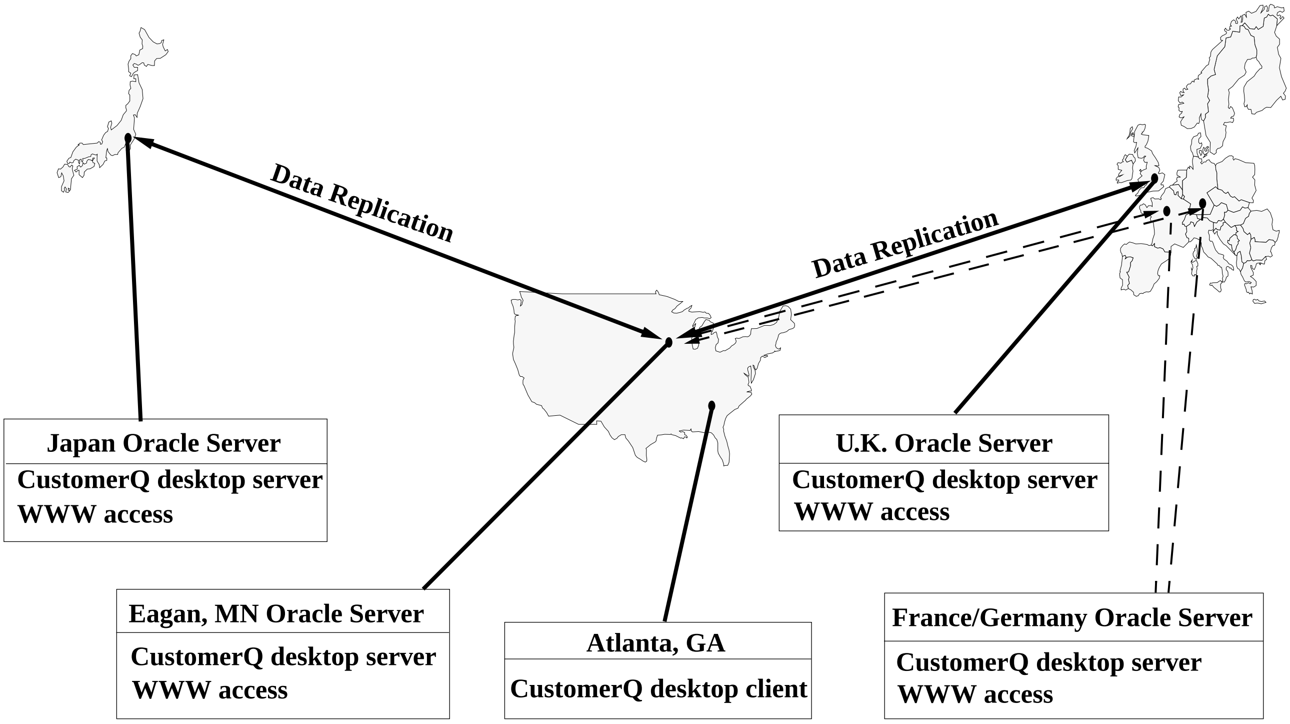
- A complex system such as CRUISE must be based on the processes it is meant to support. The documentation and development based on the processes is a must.
- Users must be involved. As many users as possible should be involved in the process definition, testing, requirement definition, priority setting, etc. Without user involvement, the development team risks being viewed as a group that does things "to" users rather than "for" users.
- Take advantage of usability labs. Do as much prototyping as possible to learn how users are reacting to the software. Be prepared to make drastic changes even if it means compromising schedules.
- If your deployment is based on third party software, as ours was, you must be prepared to find bugs. You must also be prepared to develop your own workarounds, as you may not get immediate resolution from the vendor.
- Be careful during your evaluation and selection to avoid buying "vapor-ware," which is software promised to be delivered. If your core design is based on this non-existent software, be prepared to scramble when the vendor's software is late.
- If your current staff is used to developing applications from scratch, be prepared for a negative reception from programmers toward a purchased product/toolkit. This reaction may be as strong as programmers resigning.
- Don't underestimate the need for user training and documentation. This is often put off until the end. A very well designed product can suffer if the users don't understand what to do or how to use it.
- Keep your users well informed. This will help keep them on your side.

CRUISE continues to be enhanced as we learn more about business issues, problems in our documentation, and receive general feedback from users. My advice to anyone starting up any IS project is to *Keep Your Users Involved*!!

Chart 1 - Vendor Comparison Tool

		Vendor 1		Vendor 2		
Feature	**Priority**	**Rating**	**Scoring**	**Rating**	**Scoring**	**Total Possible**
Report Formatting	5	8	40	9	45	50
Programmable Report Generator	4	8	32	5	20	40
E-mail Output	3	6	18	5	15	30
Vendor Customization	2	5	10	8	16	20
Network Configurator	1	0	0	10	10	10
Total Score			**100**		**106**	**150**

Chart 2 - CRUISE Distributed Architecture


Data Replication
Data Replication
Japan Oracle Server
CustomerQ desktop server
WWW access
U.K. Oracle Server
CustomerQ desktop server
WWW access
Eagan, MN Oracle Server
CustomerQ desktop server
WWW access
Atlanta, GA
CustomerQ desktop client
France/Germany Oracle Server
CustomerQ desktop server
WWW access