1.1 Methodology Research

1.1.1 Spiral Model

The Spiral development model is a risk-driven process model generator. It is used to guide multi-stakeholder concurrent engineering of software-intensive systems. It has two main distinguishing features. One is a cyclic approach for incrementally growing a system's degree of definition and implementation. The other is a set of anchor point milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions.

The spiral model, also known as the spiral lifecycle model, is a systems development method (SDM) used in information technology (IT). This model of development combines the features of the prototyping model and the waterfall model. The spiral model is favored for large, expensive, and complicated projects.

The steps in the spiral model can be generalized as follows:

- The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:

    - evaluating the first prototype in terms of its strengths, weaknesses, and risks;
    - defining the requirements of the second prototype;
    - planning and designing the second prototype;
    - constructing and testing the second prototype.
- At the customer's option, the entire project can be aborted if, at this point, the risk appears too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and, if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.

1.1.2 Rapid application development

RAD (rapid application development) is a concept that products can be developed faster and of higher quality through:

- Gathering requirements using workshops or focus groups
- Prototyping and early, reiterative user testing of designs
- The re-use of software components
- A rigidly paced schedule that defers design improvements to the next product version
- Less formality in reviews and other team communication

1.1.3 Waterfall Model

The waterfall model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back without a formal process and plan.

The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a carwash, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping or iterative steps.

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage. Alternatives to the waterfall model include joint application development (JAD), rapid application development (RAD), synch and stabilize, build and fix, and the spiral model.

This model is antiquated and only useful in situations where all of the requirements and implementation details must be and are known prior to any development work, such as a project involving human safety.

1.2 Choosing a Methodology

We researched a number of methodologies that we can incorporate into our Web Based Evidence Management system. After a very intensive and informative literature review, we chose to adopt a Spiral method because:

- Short development cycles allow for quick prototyping and functionality
- Risk-based – ICG may decide risk of our product is too high and cancel after seeing initial prototypes.
- Collaborative – Interfaces between parallel developments must be well defined. Because our project will support a scientific practice and litigation, failure to collaborate with our intended users and sponsors will result in project failure.
- Anchor point milestones – ICG can start using early prototypes for familiarization while later revisions are developed.
- Our requirements are not fully understood at the beginning of the project and the technology that computer forensics supports changes daily.

1.2.1 Qualitative Methodology Selection Table

There follows a qualitative means of selecting a development model. We have compared and contrasted the Waterfall, Spiral, Scrum, Prototyping and RAD development methodologies by assigning numerical values to evaluation criteria for each methodology, and weighting each criteria's importance within our project.

			MODEL				
	Criteria	**Score**	**Waterfall**	**Spiral**	**Scrum**	**Prototyping**	**RAD**
1	Business Value	10	1	3	8	8	8
2	Superior Quality	15	8	8	8	6	11
3	Low Cost	10	3	6	5	7	8
4	Low Risk	10	2	10	6	8	8
5	High Speed	5	1	3	5	5	4
6	Ease of Integration	15	7	5	5	7	9
7	Maximum Reusability	5	2	2	5	3	3
8	High Flexibility	10	1	4	7	3	9
9	Sponsor Feedback	5	1	2	5	5	4
10	Ease of Maintenance	15	10	7	9	9	11
	Total	100	36	50	63	61	75

Table 1.1 - Methodology Table

The EEMS team believes RAD will suit our clients' needs and will provide the best overall value to the development process. Since this project is a real life implementation and RAD suites the best in real life application development, we have chosen to go with RAD. Since we are committed to do JAD session to get a better grasp of our client needs we have chose RAD. Since we are planning to deploy this application by Jan of 2005, frequent prototyping is essential and RAD provides us with that methodology.