Developments in Business Simulations and Experiential Learning, Volume 32, 2005

COMPUTER BUSINESS SIMULATION DESIGN: THE ROCK POOL METHOD.

Jeremy J. S. B. Hall
Hall Marketing
jeremyhall@simulations.co.uk

ABSTRACT

This paper explores software design methodologies in the context of business simulation design and proposes a methodology - the rock pool method that provides structure while maintaining creative freedom.

Developing computer simulations for management development and business education present particular software design problems. On one hand, for computer software development there is a need for a rigorous, structured approach. But, equally, creating simulation models that deliver learning in an effective, efficient and consistent way is a creative process.

The "rock pool" metaphor was chosen because the process can be likened to exploring the rock pools on a beach after the tide has receded. Each rock pool represents a stage in the design process. Within each rock pool there are several design elements (the rocks) but these are not processed in a predefined order and are revisited several times. Although moving between rock pools is systematic, the order the rocks are explored within a rock pool depends on the simulation and the movement between rocks is based on creative needs.

The methodology is explored in the context of the development of a complex entrepreneurial planning simulation (Strategic Exploration of Entrepreneurial Directions (SEED)) for students of science, technology and medicine and other simulations

INTRODUCTION

General software design methodologies divide into traditional, rigorous methodologies and the more recent "lightweight" or agile methodologies.

Traditional (Highsmith 2002) or "heavyweight" software design methodologies are rigorously structured. They are exemplified by the *waterfall method* (Brooks, 1995). These methods involve working sequentially through a series of stages at the end of each the stage is approved before moving on to the next stage. In the context of computer simulation the *Stairway Methodology* (Allwood, 2001) is a rigorous software method (RSM) where there are five main stages (definition, formulation, evaluation, modification and completion) and within each of these there are several steps that are cycled through.

More recently agile (lightweight) methodologies (Poppendieck, 2003) have appeared such as Extreme Programming (XP) (Beck 2000) and Feature Driven Development (FDD) (DeMarco, 1999) where the design process is not rigorously structured. Rather, it is a feedback based process (Anderson, 2003) where the working software is delivered in stages and where the software *evolves* through frequent refactoring (rework and redesign based on software usage experience to simplify the software). It is argued that this approach is better at dealing with the complex adaptive system of software development (Anderson, 2003). In the context of gaming and simulation design the process is suggested as a spiraling cycle of *goal setting, action through technique* and *interpretation of results* (Bizzocchi, 2003). And, Jones (1998) states "Authorship is not sequential. Teachers love aims, but authors love ideas much more. To stipulate that an author should start then follow it by devising parameters is, to my mind, unrealistic and inefficient". But, as this statement is made in the context of students designing and running their own simulations, it is likely that the feedback process inherent in lightweight methodologies is necessary for these naïve designers to learn about the simulation design process.

Anderson (2003) suggest that the choice of software methodology depends on the maturity of the application and whether the project must be delivered holistically or can be delivered incrementally. Anderson shows these in a two by two grid (Figure 1)

In the Maturity/Delivery Matrix, the upper right-hand corner represents conventional (mature) data processing applications that must be delivered as a whole. And Anderson suggests that rigorous software methodologies (traditional, heavyweight methods) are best for these. But by

	Immature	Mature
Holistic Delivery	Agile Process	RSM Process
Partitioned Delivery	Lightweight Agile Process	Either Agile or RSM

Figure 1: Maturity/Delivery Matrix

their nature, business simulations are immature products where each new simulation may require a considerable innovation. (Where innovation is necessary because of continuing technological change, changes in the use of technology for education, changing business processes and structures and widening application of simulation). But in terms of the delivery dimension, it is not usually possible to deliver a new simulation in stages. Thus computer business simulations fall into the top left-hand corner of the matrix, where Anderson suggests that agile methods are most appropriate as these utilize feedback to handle the immaturity of problem specification.

This paper suggests a middle road where, at a macro level the design process is a rigorous software methodology but at a micro level it is agile. The Rock Pool Method combines both heavy and lightweight methodology using a rigorous structure where the simulation is developed through several major stages (rock pools) but within each rock pool development is unstructured and agile. Experience developing several simulations suggests that the agile, lightweight approach within a rock pool is appropriate to the immature (creative) nature of simulation design. And the rigorous structure imposed by the rock pools ensures that a sound simulation is developed.

The *Rock Pool Design Method* metaphorically proposes that the design process is one that involves moving progressively and sequentially between rock pools where each rock pool represents a major design stage. And where, within each rock pool, there are several design elements, each represented in the metaphor by a rock. At an individual rock pool stage, reflecting the intrinsically creative nature of computer simulation design, design is not a sequential process and does not have a defined starting point. Rather, depending on the simulation, its purpose and the designer, a rock pool development stage can start with any rock. As the development progresses the designer moves between rocks revisiting each several times. When a rock pool's design tasks are complete, the designer moves on to the next rock pool.

DEFINED STAGES - THE ROCK POOLS

The rock pool method involves progressively moving from one to the next of the following rock pools:

1. Needs Definition
2. Simulation Specification
3. Simulation Design
4. Simulator Development
5. Simulation Validation
6. Finalize Design

And, these define the structured and sequential elements of the method.

1. THE FIRST ROCK POOL - NEEDS DEFINITION

The "need definition" rock pool consists of four elements:

- **a.** Specifying the Target Audience(s)
- **b.** Defining Learning Objectives
- **c.** Settling Duration
- **d.** Defining Manner of Use

a. Specifying the Target Audience(s) involves defining who will participate in the simulation, who will run it and the type of organization that will use it and authorize its use. Randolph and Posner (1979) emphasize the identification of student skills and motivation, instructor skills and values and the institutional and professional pressures/concerns as factors affecting the effective design on learning situations and this suggests that the specification of these factors is a necessary part of defining needs.

For participants the following are important:

Range of prior knowledge and experience
Diversity of prior knowledge and experience
Maturity and expectations

Knowledge of the range of prior knowledge and experience provides a basis to assess the need for participant and tutoring support. The diversity of knowledge and experience shows the extent to which prior participants can share knowledge and act as a learning resource. Maturity and expectations indicates the extent to which participants can handle the pressures of the activity, ambiguity and uncertainty. Thus range and diversity of prior knowledge and experience define cognitive support needs and maturity and expectations define affective support needs,

For the Strategic Exploration of Entrepreneurial Directions (SEED) simulation, the target audiences were university students, business people who are considering starting their own business and as a role reversal exercise for bankers and tax officers. As the prime student group would be studying science, technology and medicine rather than business the simulation would need to build in comprehensive knowledge support and help them handle the ambiguity and uncertainty associated with entrepreneurship. Further, their lack of world experience may make them uncomfortable with this form of learning (Knowles, 1998)

For the trainers/academics (tutors) running the simulation there are several issues:

Knowledge of the subject being taught
Familiarity with the use of simulations
Familiarity with technology

The level of knowledge define the extent to which knowledge support must be encapsulated in the simulation and the extent to which the trainer must be automated out of the process.

Familiarity with the use of simulation and the technology defines the extent to which the tutor will be comfortable with the activity, the need for training and parallel running (where the tutor runs the simulation in partnered with an experienced used).

b. Defining Learning Objectives is a wider definition than just defining learning (knowledge acquisition or even skill development) needs. Hall (1996) uses a five dimensional model:

Knowledge Exploration
Skill Development & Practice
Motivation/Behavioral Needs
Assessment
Enhancing Learning

For the SEED simulation, the knowledge to be explored was that associated with planning an entrepreneurial start-up company (the entrepreneurial ideal) and developing entrepreneurial skills (analysis, decision-making, etc.). Motivational needs included *"enhancing the entrepreneurial culture"* and providing a learning activity that was engaging and fun. Although, the simulation would not be formally examined, on an informal basis participants should be able to assess their current levels of business knowledge and use this to decide personal development plans.

c. Settling Duration involves deciding the amount of time that can be budgeted for the simulation. After deliberation, the duration for the SEED simulation was to be one day (six hours). As complexity is highly correlated with duration (Hall & Cox, 1994) this would limit the number of decisions and the size of the simulation model.

d. Defining Manner of Use involves defining the way that the simulation would be used and Hall (1996) suggested the following taxonomy of use (Figure 2):

As a course element

1. Course Finale
2. Course Starter (Icebreaker)
3. Course Theme
4. To Reinforce a Topic
5. As a "Break"

Standalone

6. Stand-alone Seminar
7. Distance/Spare Time Learning
8. On an Assessment/Development Center
9. At Business/Company Conference
10. As a Promotional Contest

Figure 2: Manner of Simulation Use

For the SEED simulation, for Imperial students, it would be run as a series of stand-alone seminars. But, for prospective entrepreneurs, reflecting their situation it may be run as a distance/spare time learning activity or as a course finale.

Development sequence

These elements are not progressed in any predefined sequence nor is there any definite starting point. Rather, different simulation designs start from different starting points and progress, recursively, between the other rocks in this rock pool.

The SEED simulation's starting point was specifying the Target Audiences - university students and business people whom are considering starting their own business. But the development of other simulations involved other starting points. The designs of Financial Analysis (1985) and SMART (1987) both had their starting points the Definition of Learning Objectives. Product Launch (1977) and Executive Ladders (1993) both had Duration as the design starting point. The Challenge Series (Management (1986), Retail (1987) and Service (1989)) were developed for use as part of an international promotional contest and so had Manner of Use as their starting points.

Generally, after defining the first *rock* to visit the others are visited immediately. For instance, for the Challenge Series, immediately after defining the Manner of Use the duration was settled. Also, this stage involves moving between and revisiting elements. For the SEED simulation, the duration was initially to be one & a half days but later reduced to one day (6 hours). This required revisiting the learning objectives (but equally might have meant redefining the manner of use). Also, while defining needs, account was taken that the SEED simulation might be used with prospective entrepreneurs, on a distance learning basis, and, perhaps as a management contest.

Stage outcomes

At the end of this stage there is a need to examine how the elements relate to each other and resolve any conflicts. For instance, the target audience coupled with learning objectives for the SEED simulation meant that its scope would stretch the knowledge base of the participants. Because of this, it would be necessary to build in significant participant support (in the form of business advice). Further, the short duration (a day) coupled with the Learning Objectives would cause a major development problem. And, this was amplified by the fact that the simulation would be run in a single, undivided session. (If the simulation had be spread, in separate sessions over several weeks, the participants would have the opportunity to reflect and, possibly, budget extra time.)

2. THE SECOND ROCK POOL - SIMULATION SPECIFICATION

This rock pool consists of these elements:

a. Define issues
b. Decide simulator type
c. Decide Delivery mode
d. Decide version(s)
e. Decide business scenario

a. Defining issues involves translating the learning objectives into business oriented issues that are appropriate to the target audience. For the SEED simulation the issues included market selection, pricing, promotion, working capital and venture funding. The issues define the discussion areas for the team and hence the areas where *deep cognitive processing* occurs.

b. Deciding simulator type involves deciding the structural aspects of the simulation. Biggs (1990) suggested one way of classifying business simulations was total enterprise and functional. Where total enterprise simulations attempt to replicate a whole business. In contrast functional simulations replicate in detail a particular function (such as marketing or operations).

Hall (1996) expanded on this with a taxonomy of model based simulations (Figure 3).

1. Total Enterprise Simulations
2. Functional Simulations
3. Concepts Simulations
4. Planning Simulations
5. Analysis Simulations
6. Computer Enhanced Role-plays

Figure 3: Simulation Taxonomy

Concepts simulations are short simulations that address one or a very limited range of business concepts (such as the Product Life Cycle). These simulations involve progressing through several simulated time-periods but there are other processes. Planning simulations generally involve using the simulation model to investigate several "What-If" scenarios and determining the "best". Analysis simulations involve performing a series of analyses (on sales or inventory data etc.). Based on this analysis participants recommend and defend business policies or forecasts. Computer enhanced role-plays utilize simulation models and databases to support the negotiating parties.

Initially, it was felt that the SEED simulation was a planning simulation where participants would utilize a what-if model to determine the *best* plan. But beside the need to perform What-If analyses, it was necessary for the participants to experience the planning process and especially the time taken to develop a business plan. This meant that participants were only allowed to make a limited number of analyses before a simulated month passed. In turn, this meant that participants had to balance the risk of an incomplete and inadequate plan against being "fast to market" - a situation that was magnified by the seasonal nature of the market and the need to become established before the seasonal peak.

c. Decide Delivery mode involves who uses the simulator. Elgood (1997) and Biggs (1990) divide delivery modes for computer simulation into those where the trainer rather than the participants use the hardware and where it is used directly by the participants rather than the trainer. Total enterprise simulations tend to involve all teams competing in the same marketplace and this means that the same simulation model must process the decisions from all the teams and so it is the trainer who uses the computer hardware Biggs (1990). In contrast, for simulations such a Product Launch (1977), Sales Calls (1983) and Operations (1983) the competitive aspect of the marketplace is simulated by the model and the hardware is used directly by the participants.

As the SEED simulation addresses planning issues it is non-competitive between teams and it is fitting for each team to make direct use of the simulator. Also, as the teams can work asynchronously, this shortens the simulation's duration. However, set against this is the fact that direct use of the simulation can change team behavior (Coote, 1985) and the software must be designed to minimize this risk (Hall, 1995).

d. Decide version(s) involves deciding the versions the simulation will have. For instance, the Management, Service and Retail Challenge simulations were originally designed for use in a managerial contest, But they were developed assuming that they could also be used on business & financial appreciation courses, on assessment & development centers, stand-alone, as a course finale and as a course theme. This range of use helps define the decisions and reports provided by the simulation.

For SEED, the different target audiences and manner of use conjoined and so it was decided to create a single version. (If different uses were desirable then several versions would have to be developed ranging from basic (for the contest) through more complex versions with additional decisions and reports.) However, it may be that at a future time, versions will be created with different tax structures and in different languages.

e. Deciding the business scenario involves deciding the industry to be simulated (Teach, 1990). Often the type of organization that will be using the simulation predefines this and this was the case for CISCO (1988), Casino Challenge (1990) and Profess (1998). (Where CISCO addressed the issues facing construction companies; Casino Challenge the issues facing casino management and Profess the issues facing the sales management of a financial services company.)

But, for the SEED simulation, although the target audience did not predefine the business scenario, there were several factors that made the choice important. First, because of the age of the target participants, it was necessary for the simulation to be engaging and the product or service should be "real". But because of its entrepreneurial nature the product could not be an existing product. It must be a product that might exist because of market needs but does not because of technological or economic factors. Taking into account prior knowledge & experience, it needed to be a relatively simple business in marketing, operational and financial terms. That is to say the market structure and marketing mix should be simple and manufacture should consist of a single process with few component materials. Also, it should take into account the scientific and engineering background of the participants and the product should be high tech and thoroughly modern! The product chosen was a soft toy (like a teddy bear) incorporating electronics to link it to a normal home PC and allow it to communicate with a small child (aged between two and five). It would be used as a companion for the child and, via the PC, provide early learning. Reflecting its high-tech aspect the product is the Cuddl-Etoy! The scenario explained that the Cuddle-Etoy was developed after a failed University project (where the toy was developed as a collaborative project involving maths, engineering and medical students as an aid to diagnosis for very young children). A project that failed because of difficulty with sterilization and the screaming fits when the child had to give up the toy after the diagnosis session! At the second pilot we illustrated the product with several real cuddly toys. Immediately on arrival, two female students annexed these, but since they were destined for a member of the academic staff they had to be retrieved.

Development sequence

Again, there is no defined starting "rock" in this rock pool. For the SEED simulation the starting point was defining the scenario. In contrast, the starting point for the development of Financial Analysis (1985) was to define the issues that had to be considered when making a business plan and the scenario was of minor importance. Equally, manner of use may mean that the starting point is deciding the simulator type. For instance, as the Challenge Series (1986, 1987, & 1989) was developed as part of a management contest the simulation had to be competitive with the decisions of one team interacting with the others and where business issues secondary and the business scenarios of no importance.

Stage outcomes

At the end of this stage the key structure of the simulation is defined and specified. This may be used to search for and select a suitable *existing* simulation or as the basis of the development of a new simulation (as was the case with the SEED simulation).

3. THE THIRD ROCK POOL - SIMULATION DESIGN

This rock pool consists of these elements:

- **a.** Decide decisions
- **b.** Decide results
- **c.** Create models linking decisions & results
- **d.** Create validation & quality assurance support
- **e.** Develop preliminary documentation

Besides these, there is the need to design or have access to software routines to manage the user interface - decision entry, report preparation and display, on-line help system etc. However, for SEED a *software shell* was used to provide this functionality.

a. Deciding decisions is a creative process starting from the issues list (2a) and learning objectives (1b). For each issue the designer must ponder the cognitive processing required to ensure that the participants fully think through the issues and adequately explore the knowledge needs. For example for the SEED simulation and reflecting its entrepreneurial background, decisions separated into three key areas:

I. Marketing
II. Finance
III. Resourcing

The Marketing Plan decisions are shown in Table 1. When considering whether to sell via retail outlets, the web or both, participants must think through issues such as pricing, promotion, sales growth, profitability (margins and inventory investment) and cash flow.

Besides considering the decision areas, it is necessary to think of the *form* of the decision. And this is illustrated in Table 1 where for some decisions are numbers (e.g. the prices) and others involve choices from one or more options (e.g. sell to retailers or form of website). The decisions that only allowed a (limited) range of options were to limit cognitive complexity and allow for the short duration (1c).

Marketing Plan				
	Plan 1	Plan 2	Plan 3	Plan 3
Market Served	Learning	Learning	Companion	Autistic
Launch Month	July	July	July	July
Number in Range	4	4	4	4
Sell to Retailers	Yes	No	Yes	Yes
Web Site	Full	Full	Full	Full
Web Price	64.99	64.99	64.99	64.99
Recommended Retail Price	64.99	0.00	64.99	64.99
Price to Outlets	44.99	0.00	44.99	44.99
Advertising Spend	5000	5000	5000	5000
Public Relations	Both	Launch	Both	Both
Enhanced Web Site	Yes	Yes	Yes	Yes
Point of Sale Display	Yes	No	Yes	Yes
Packaging	Fancy	Plain	Fancy	Fancy

Table 1: Sample Marketing Decisions

b. Decide Results involves determining what results are produced by the simulation model and these can be divided into the categories:

I. Results for the participants
II. Results to reconcile results
III. Results that *comment* on team performance
IV. Results that *explain* the model
V. Results that validate models

Participant Results are those that will be provided to the participants as the simulation progresses. Reconciliation Results are those that can be used if necessary to help answer participant questions about accounting and operational models. The *comments* provide feedback that is not quantitative and exact but is qualitative and *fuzzy*. The model explanation reveals to the designer and the tutor how the model is responding to decisions such as price and promotion. The validation results are provided (during the design stage) to help the simulation designer explore model behavior and these are expanded on later (3d).

b. Create models linking decisions & results involves developing the logic and algorithms that link the decisions and results and defining the data associated with these.

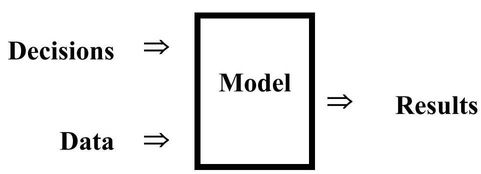

Figure 4: Linking decisions and results

At this stage the model algorithms are researched and described and if necessary *prototyped* on a spreadsheet. As the SEED simulation was complex several models were prototyped (Figure 5)

Innovation Diffusion Model
Outlet Penetration Model
Price/Margin Sensitivity
Production Cost/Investment
Production Volume/Cost Analysis
Office Investment
Selling Method Analysis
Bank Financing
Inventory/Sales Tax
Product depth factors
Outlet Penetration Factors

Figure 5: Prototyped Models

c. Create validation & quality assurance support involves creating routines and reports that reveal explicitly how the models work. With SEED, translating marketplace decisions into sales demand was very complex with multiple interacting factors that affected demand. Thus, as part of the validation and quality assurance process the individual market responses were captured and were made available in a report. A second example, was the reconciliation of creditor changes. Some of the reports used during the design process to *drill down* into the model are shown in Figure 6.

Outlet Sales Audit
Materials Audit
Price & Cost Trends Audit
Operating Expense Audits
Office Staff Needs Audit
Staffing Levels Audit

Figure 6: Reports to check and explain results

Table 2 details the Outlet Sales Audit showing how sales, number of outlets, etc. developed over time (taking into account seasonality and penetration).

d. Developing preliminary documentation involves recording and describing the decisions, results and models. This may take place before, concurrently or after particular decisions, results and models have been designed. The documentation separates into three parts - the participants' brief, the trainer's manual and online help. If a predefined *shell* program had not been used a fourth document describing the technical aspects of the software would need to be developed. Also, if appropriate, for inexperienced trainers and academics it might be necessary to produce a document discussing the classroom aspects of running the simulation. Again, for SEED this document existed.

Outlet Sales Audit								
	July	Aug	Sept	Oct	Nov	Apr	May	June
Unit Sales (outlets)	20	80	316	388	324	272	280	364
Number of Outlets	4	8	13	17	21	95	99	104
Sales per Outlet	5	10	24	23	15	3	3	4
Sales Revenue	899	3597	14208	17446	14568	12237	12597	16376
New Outlets	4	4	5	4	4	4	4	5
Initial Sales	8	8	35	56	88	8	8	10
Repeat Sales	14	69	278	331	235	260	271	353

Table 2: Example of design audit report

I. Participants' Brief
II. Trainer's Manual
III. Online Help
IV. (Using the Software)
V. (Using Simulations)

Figure 7: Documentation Needs

In parallel to the development of the online help system the *participants' brief* and the *trainer's manual* were drafted. The participant's brief is the document provided to the participants to read and become familiar with before starting to use the simulation. The trainers' manual is for the trainers to help them understand the simulation and the learning derived from it. Additionally, at this stage, the trainer's manual serves to document the design and both are drafts and focus mainly on the decisions, results and the model.

Development sequence

In this rock pool, reflecting the level of creativity, movement between the rocks (elements) was very complex as the decisions; results and models were refined and refactored. For instance, the first design work involved the marketplace models. But the complexity and dynamics of these meant that these were also the last to be completed. Effectively, this pool does not comprise a few large rocks but consists of several piles of weed encrusted stones with a large number of agile and elusive creatures darting between them! For the SEED simulation the starting point for a cycle tended to be defining a decision or group of decisions, deciding on how they interacted and produced results and then defining the result set. However, in contrast, for the Financial Analysis planning simulation the starting point was to define what results were to be produced and only then the decisions that impacted these. And, for the DISTRAIN (2004) simulation the issues (2a) were used to design the models and then the results and decisions were determined.

Stage outcomes

At the end of this stage an *alpha test* version of the simulation exists ready to refined by the next stage into the *beta test* version. At this stage, the documentation's purpose is to support the designer rather than for use by the participants and the tutors. Thus, later, it must be modified and refined to be of use to the participants and the tutors.

4. THE FOURTH ROCK POOL - SIMULATOR DEVELOPMENT

This rock pool consists of these elements:

a. Test models
b. Calibrate models
c. Ramp workload
d. Create learning & tutoring support
e. Refine documentation

This rock pool brings the simulation to the beta test stage

a. Testing models involves ensuring that algorithmic logic and program code are correct. If logic (the algorithms) are incorrect the simulation will lack face validity (Woolfe, 1989) and this will affect learning (Teach, 1990). And if the simulation is incorrectly coded results will be calculated incorrectly and, even worst, the software may *crash.* Biggs and Halpen (2004) arguing a counter view about the *utility* of BUGS (defined by them as Basic Unplanned Glitch Situations) provide a taxonomy of problem areas.

The quality and validation support (3c) help with this process but for a complex simulation it is often necessary to develop a series of spreadsheets to reconcile results. And, for financial validity, the Balance Sheet must balance and this tests the accounting models.

b. Calibrating models involves ensuring the realism of outcomes and balancing results. In particular the balance between the ability to generate profits (profitability) and cash flow (survival) is crucial. Also, for SEED it was important that no particular plan was obviously the best. This was both to ensure adequate discussion and to ensure that the teams had to face the problem of deciding when they should choose which plan they were to go with.

c. Ramping workload involves increasing the number of decision areas or reports produced as the simulation progresses thus reducing the risk of role overload (French, 1972) in the early stages of the simulation. Teach (1990) discusses the same problem in terms of "analysis paralysis". And Hall (2004) suggests that by ramping the complexity as the simulation progresses not only is the risk of role overload diminished but learning is enhanced as the additional reports and decisions "stimulate discussion and provide the opportunities for additional cognitive development". For the original pilot of the SEED simulation workload was not ramped and, as described later, this caused role overload.

d. Creating learning & tutoring support involves defining reports, help texts and decision screens that provide information to reconcile the results, advice & explanations, knowledge support (for tasks) and identification of strengths, weaknesses and possible problems.

The standard *shell* used for SEED provided a full *online help system* with using the software, the current task, definitions of terminology and an online manual. Software help already existed but help with the current task, definitions and the online manual had to be developed as a context sensitive, hypertext database. Much of the data in the database could be copied from the draft participants' brief and background notes. But, reflecting the difference between print media and screen display, the data was edited and abstracted.

e. Refining documentation involves taking the draft documentation created in stage 3d and editing it to improve clarity, remove omissions and incorporate the parameters determined during calibration (4b). And, now the major development work is complete, it is appropriate to develop a Power Point brief.

Development sequence

This rock pool is closely associated with the simulation design rock pool and generally involves some movement between the two. For instance as the models are calibrated they may need to be modified. Also, as happened with the previous stages there was movement between the *rocks* in this rock pool. In particular, creation of learning and tutoring support was closely linked with refining the documentation. Also, as the models were tested it becomes apparent where both the online help system and the printed documentation needs to be clarified and expanded. Finally, testing and calibration are linked.

Stage outcomes

At the end of this stage the simulation has reached the *beta test* stage and is ready for piloting (testing with *real* people) and the authentication of learning.

5. THE FIFTH ROCK POOL - SIMULATION VALIDATION

This rock pool consists of these elements:

- **a.** Pilot simulation
- **b.** Refine and modify the simulator
- **c.** Refine and modify documentation
- **d.** Authentication

a. Pilot simulation involving testing the simulation with a group of *forgiving* participants or trainers. The SEED simulation had two main pilots with several groups of students. Piloting not only tested the robustness of the models and code and how the simulation delivered learning but also tested *process aspects* - workload, behavior and usability.

Although it was realistic to allow participants access to the full set of decisions from the start, on the first pilot this caused role overload (French, 1972). As this risk was considered when settling duration (1c) and again when ramping complexity (4c) we were ready to introduce the decisions in stages and this was done for second pilot.

Checking the *behavioral* aspects of the direct use of the simulation by the participants (Coote, 1985) was identified as a possible problem earlier when deciding the delivery mode (2c), but it did not cause problems.

b. Refine and modify the simulator involves taking feedback from the pilot and correcting logic and code errors. For SEED the major change was to phase the introduction of decisions and results over the first four simulated months (Figure 8).

c. Refine and modify documentation involves checking the completeness of *all* documentation (participants' brief, trainer's manual and the online help). For the participants' brief this involves improving clarity and this may mean adding to the manual or subtracting from it. Experience suggests that the design of the participants' brief is a balance between completeness and length. For, if the brief is too long it will not be read! Because there is an online help system there is the choice between adding to the paper document or to the online help or both.

d. Authentication involves obtaining the views of the participants, subject matter experts (SMEs), academics and trainers and teachers. For SEED feedback from the participants was through a questionnaire.

Development sequence

Unlike the previous rock pools where there are no defined sequences, this stage involves a repeating cycle of piloting and modification until the simulation authentically meets the needs and issues defined in the first and second "rock pools".

Stage outcomes

At the end of this stage a fully operating simulation exists and all that remains is *packaging* and *dissemination*.

7. THE SIXTH ROCK POOL - FINALIZATION

This rock pool consists of these elements:

- **a.** Finalize documentation (participant & tutoring)
- **b.** Finalize tutoring support
- **c.** Dissemination

a. Finalize documentation (participant & tutoring) involves editing the documentation to improve accessibility (readability, spelling, layout etc.) and incorporate the changes made during the validation stage.

b. Finalize tutoring support involves the final completion of the software. For instance, after the pilot of the DISTRAIN simulation, the reports for the trainers and the participants were modified to match the requirements of the people who were to run the simulation.

Month	**Planning Options**
January	Business Research & Policy Advice
February	As January plus Marketing decisions
March	As February plus Resourcing & Working Capital decisions
April onwards	All decisions

Figure 8: Phased introduction of decisions and results

d. Dissemination to organizations, to trainers, to students (participants) and via the academic community involves writing papers (such as this), press releases, printed and electronic materials (such as a website) etc.

Development sequence

Again there is no defined sequence between these tasks and very often the finalized documentation and tutoring support use the same textual and graphic data in a different form. Further, for SEED some of the graphics from the Power Point briefing and the online help system were used on the website.

Stage outcomes

A complete simulation package.

CONCLUSIONS

Anderson (2003) uses the Lapre and Van Wassenhove Project Matrix (2002) to suggests that the choice of appropriate software methodology depends on the degree of *Operational Learning* (experience "know-how") and degree of *Conceptual Learning* (cause and effect understanding). Where each of these rank from low to high (Figure 10).

		Operational Learning	
		Low	High
Conceptual Learning	Low	fire fighting	artisan skills
	High	unvalidated theories	operationally valid theories

Figure 10: Lapre and Van Wassenhove Project Matrix

In the context of computer simulation design, the Operational Learning dimension relates to design experience. Here low Operational Learning is where the designer or designers have no or little experience of simulation design and high Operational Learning is where the designer has developed several simulations and has researched simulation design. And, the Conceptual Learning dimension relates to the *difficulty* of the current design where Design Difficulty is a combination of simulation complexity and novelty. The measurement of simulation complexity has been explored in several papers (Wolfe 1978; Pray and Gold 1982; Gold and Pray 1984) that link it to the number of decisions and software size (model size). Simulation *novelty* depends on the degree of innovation in simulation structure and the situation (industry) modeled. Two simulations illustrate this. In model terms and industry modeled, Financial Analysis (1985) is simple but the structure of the simulation was an innovation for the designer. In contrast, Foundation Challenge (2002) has intermediate complexity and a standard structure, but as it models a not-for-profit organization it was an innovation for the designer. These two examples also illustrate that the novelty level depends on the experience and knowledge of the designer and hence the amount of conceptual learning required. For the SEED simulation, the model was complex and the structure necessary to enable a complex simulation to have a short duration had high novelty.

The choice of the appropriate simulation software methodology depends on the complexity & novelty of the simulation and the experience of the developers as shown in Figure 11.

		Design Experience	
		Low	High
Design Difficulty (novelty/complexity)	Low		Stairway Method
	High	Lightweight Method	Rock Pool Method

Figure 11: The Project Matrix (modified for simulation design)

One suggests that for simple simulations involving a few decisions and a simple simulation model a rigorous method such as the Stairway method is appropriate and desirable. Because one aspect of design difficulty is due to the *novelty* of the design to the designer, where the designers have no design experience the simulation defaults to highly novel. For this situation, a lightweight method is the most appropriate as it allows the designers to learn experientially as the simulation is created. For the situation where the simulation is novel or complex but the designer has considerable design experience the Rock Pool Method balances the rigor of structure with the creativity and flexibility necessary to learn to deal with the novelty or complexity of the simulation.

REFERENCES

Allwood, Julian M., Cox, Benita M., Latif, Syed S. The Structured Development of Simulation-Based Learning Tools With an Example for the Taguchi Method *IEEE Transactions in Education, Vol 44, No 4 November 2001.*

Anderson, David J. (2003) *Agile Management for Software Engineering* Prentice Hall.

Biggs, William D. & Halpin, Annette L. (2004) "On the Value of Bugs in Simulation Environments" *Developments Business Simulation & Experiential Exercises*, Volume 31, Las Vegas, Nevada: ABSEL.

Bizzocchi, Jim & Woodbury, Robert F. "A case study in the design of interactive narrative: The subversion of the interface" *Simulation & Gaming* (2004) Volume 34 550-568.

Beck, Kent, *Extreme Programming Explained - Embrace Change,* Addison Wesley 2000.

Brooks, Frederick P. (1995) *The Mythical Man Month,* Addison Wesley

Coote, A., Crookall, D. and Saunders, D. (1985) "Some Human and Machine Aspects of Computerised Simulations" *Simulation/Games for Learning* 11

DeMarco, Tom, and Lister, Timothy (1999) *Peopleware - Productive Projects and Teams,* 2nd Edition, Dorset House

Elgood, Chris (1997), *Handbook of Management Games and Simulations,* Gower Publishing Limited

French, Jr., John R.P. and Caplan, Robert D. (1972) "Organizational Stress and Individual Strain" *The Failure of Success* AMACOM American Management Association, New York

Gold, Steven C., and Pray, Thomas F. (1984) "Modelling Non-Price Factors in the Demand Function of Computerized Business Simulations" *Developments Business Simulation & Experiential Exercises*, Volume 11, Oklahoma State University, ABSEL.

Hall, Jeremy J. S. B., and Cox, Benita M. (1995) "Complexity: Is it really that simple" *Developments Business Simulation & Experiential Exercises*, Volume 2, San Diego, California: ABSEL.

Hall, Jeremy J. S. B. (1996) "Computerised Business Simulations: The Need for Unfriendly Interfaces" *Journal of Intelligent Systems*, Volume 5 N0s 2-4.

Hall, Jeremy J. S. B. (1995) *Simulation Virtual Business Experience* Hall Marketing

Hall, Jeremy J. S. B. (2004) "Computer Simulation: A Design Architectonic" *Developments Business Simulation & Experiential Exercises*, Volume 31, Las Vegas, Nevada: ABSEL.

Highsmith, James A. *Agile Software Development Ecosystems,* Addison Wesley 2002.

Jones, K. "What happens when students design and run their own simulations?" *Simulation & Gaming* (2004) Volume 29 342-347

Knowles, M. S., Holton III, Elwood F. and Swanson, Richard A. (1998), *The Adult Learner* Woburn, MA: Butterworth-Heinemann

Lapre, Michael A, and Van Wassenhove, Luk N., "Learning across lines. The secret to more efficient factories" *Harvard Business Review, October 2002*

Poppendieck, Mary and Poppendieck, Tom (2003) *Lean Software Development: An Agile Toolkit* Boston, MA: Addison Wesley

Pray, Thomas F. and Gold, Steven C. (1982) Inside the Black Box - "An Analysis of the Underlying Demand Functions in Contemporary Business Simulations" *Developments Business Simulation & Experiential Exercises*, Volume 9, Illinois State University, ABSEL.

Randolph, W. Allan and Posner, Barry Z. (1979) "Consideration in the Design of Learning Situations" *Journal of Experiential Learning and Simulation 1*

Teach, Richard D. (1990), "Designing Business Simulations" *Guide to Business Gaming and Experiential Learning,* ed James W. Gentry, Nichols/GP Publishing

Thavikulwat, Precha (2004) "The architecture of computerized business gaming simulations" *Simulation & Gaming* Volume 35 Number 2 June 2004

Wolfe, J. (1978) "The effect of game complexity on the acquisition of business policy knowledge" *Decision Sciences* 9(1)

Wolfe, Joseph and Jackson, Ralph (1989) An Investigation of the Need for Valid Business Game Algorithms, in *Developments in Business Simulation & Experiential Exercises*, Volume 16, Tulsa, Oklahoma, ABSEL

SIMULATIONS

Year	Simulation	Designer	Complexity	Novelty
2001	SEED	Hall Marketing	Complex	High
1990	Casino Challenge	Hall Marketing	Intermediate	High
1988	CISCO	Hall Marketing	Complex	Intermediate
2004	DISTRAIN	Hall Marketing	Complex	Intermediate
1993	Executive Ladders	Hall Marketing	Very Simple	High
1985	Financial Analysis	Hall Marketing	Simple	High
2002	Foundation Challenge	Hall Marketing	Intermediate	High
1986	Management Challenge	Hall Marketing	Intermediate	Low
1983	Operations	Hall Marketing	Simple	Low
1977	Product Launch	Hall Marketing	Simple	Low
1998	Profess	Hall Marketing	Intermediate	Intermediate
1987	Retail Challenge	Hall Marketing	Intermediate	Low
1983	Sales Calls	Hall Marketing	Simple	Low
1989	Service Challenge	Hall Marketing	Intermediate	Low
1987	SMART	Hall Marketing	Complex	Intermediate

The table referencing the simulations shows both complexity and novelty. Where complexity is the size of the simulation model and novelty indicates the degree to which the simulation incorporates novel features and structures. So although Management Challenge is of intermediate complexity and has a very standard generic industry model and so has low novelty. In contrast, Foundation Challenge also has a similarly intermediately complex model but replicates a novel industry. The Executive Ladders model is very simple but the simulation is very novel as it only lasts two minutes! The SEED - Entrepreneurial Planning simulation not only has a very complex simulation model but is novel because of the structure to ensure that the six hour duration is realizable.